

AUTOMATIZACIÓN DE PRUEBAS WEB
SELENIUM WEBDRIVER

Selenium WebDriver:

El Driver como Puente

Automatización de Pruebas Web con Java

Presentación Técnica



Agenda

01 ¿Qué es el Driver?

02 ¿Para qué se usa?

03 Cómo funciona paso a paso

04 Funciones principales

05 Tipos de Drivers

06 Ventajas para tu equipo

¿Qué es el Driver en Selenium WebDriver?

- **Definición:** El driver es el "puente" o mensajero entre tu código Java y el navegador web (Chrome, Firefox, Edge). Es como un control remoto que le dice al navegador qué hacer: abrir páginas, hacer clic, escribir texto, etc.
- **Analogía simple:** Imagina que el driver es un traductor que convierte tus instrucciones de código en acciones reales del navegador, igual que un usuario humano.
- **Función principal:** Actúa como un control remoto para Chrome, Firefox, Edge y otros navegadores, permitiendo la automatización completa de interacciones web.

Aplicaciones del Driver

Automatizar navegadores

Simula clics, escritura, navegación como lo haría una persona. Permite interactuar con páginas web de forma programática y reproducible.

Ejecutar pruebas

Verificar que formularios, botones, links funcionen correctamente. Automatiza el proceso de testing para garantizar la calidad del software.

Trabajar con datos reales

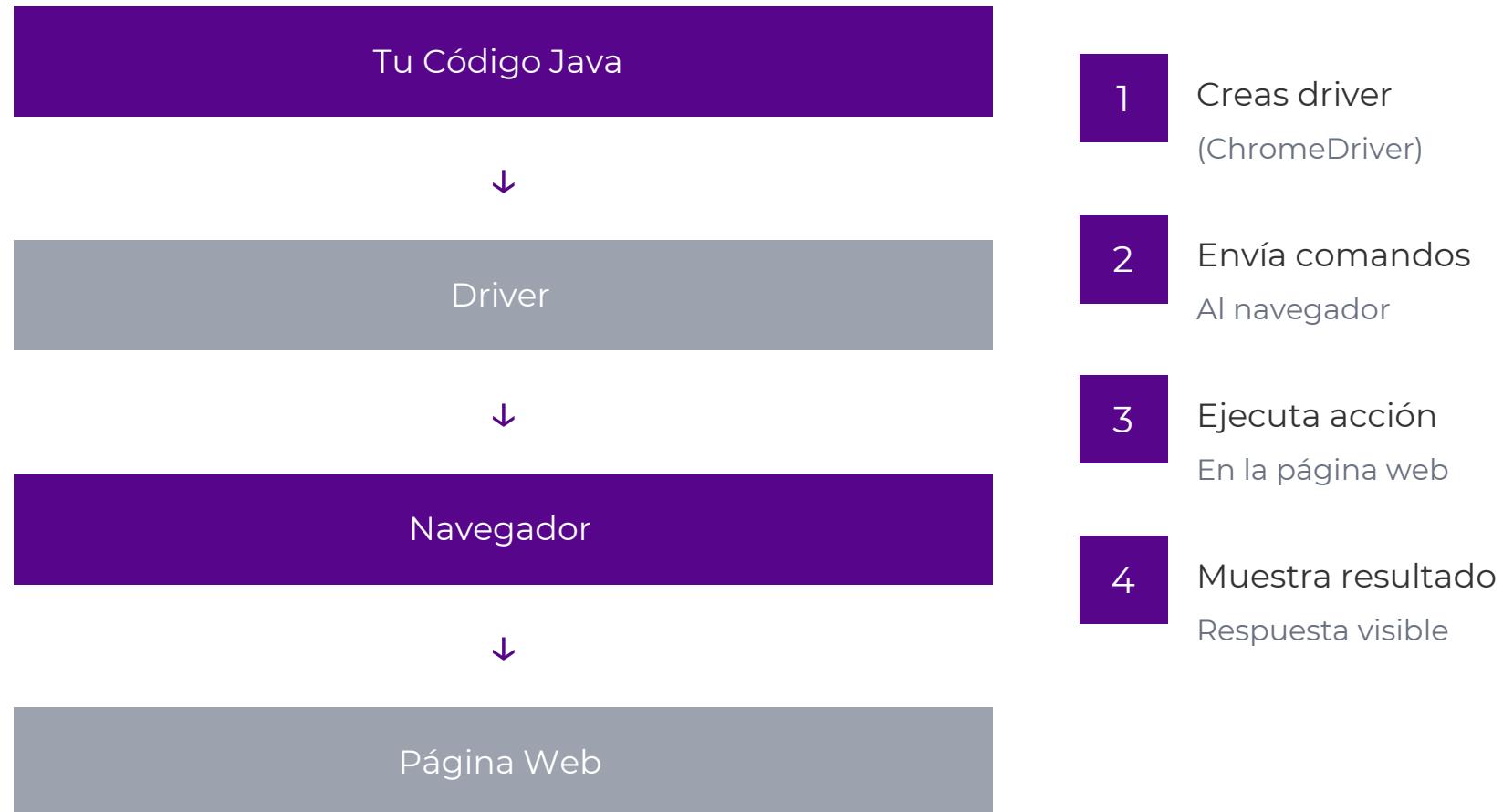
Leer Excel, bases de datos y usarlos en formularios web. Integra fuentes de datos externas para pruebas más realistas y completas.

Ejecutar en paralelo

Probar en varios navegadores/máquinas simultáneamente. Acelera el proceso de testing y mejora la cobertura de pruebas cross-browser.

Flujo de Trabajo del Driver

Diagrama de Flujo



Ejemplo Práctico

```
WebDriver driver = new ChromeDriver();  
  
driver.get("https://parabank.com");  
  
driver.findElement(By.id("username")).sendKeys("miusuario");  
  
driver.quit();
```

← Creas el "control remoto"
← Le dices: "Abre esta página"
← "Escribe esto"
← "Cierra todo"

Cada línea ejecuta una acción específica en el navegador

Funciones Principales del Driver

Acción	Código ejemplo	Qué hace
Abrir página	driver.get("url")	Navega a una URL
Encontrar elementos	driver.findElement(By.id("btn"))	Localiza botones, campos, links
Hacer clic	elemento.click()	Simula clic del mouse
Escribir texto	elemento.sendKeys("texto")	Escribe en campos
Obtener texto	elemento.getText()	Lee lo que dice un elemento
Maximizar ventana	driver.manage().window().maximize()	Abre navegador en pantalla completa
Esperar	WebDriverWait	Pausa hasta que cargue algo
Cerrar	driver.quit()	Cierra navegador completamente

Tipos de Drivers por Navegador

ChromeDriver

→ Controla Chrome

FirefoxDriver

→ Controla Firefox

EdgeDriver

→ Controla Edge

SafariDriver

→ Controla Safari

Nota: Cada navegador necesita su driver específico desde selenium.dev

Ventajas Clave

- **Gratis y open source:** Sin costos de licencia, con una comunidad activa que contribuye constantemente con mejoras y actualizaciones
- **Funciona en cualquier OS:** Compatible con [Windows](#), [Mac](#) y [Linux](#), permitiendo flexibilidad en el entorno de desarrollo
- **Soporta todos los navegadores:** Chrome, Firefox, Edge, Safari y más, garantizando cobertura completa de pruebas cross-browser
- **Integra con Excel, BD, APIs:** Conecta fácilmente con fuentes de datos externas para pruebas basadas en datos reales
- **Ejecuta pruebas 24/7 sin intervención humana:** Automatización completa que ahorra tiempo y aumenta la confiabilidad de las pruebas

En Resumen

"El driver es tu piloto automático que maneja navegadores reales desde código Java"

Perfecto para automatizar pruebas repetitivas de aplicaciones web

Key Insight:

Ahorra tiempo y aumenta la confiabilidad de las pruebas

¡Gracias!

¿Preguntas?

Selenium WebDriver

Automatización de Pruebas Web con Java

