

Herramienta de Testing

Manuel Ezequiel guerrero granados GG241501

Enrique Alexander Solano López SL223188

Norman Enmanuel Quijano Amaya QA240640



Introducción

- En el desarrollo de aplicaciones web, las herramientas de Testing juegan un papel crucial para asegurar la calidad y funcionalidad de los sistemas. Especialmente en el contexto de aplicaciones web con JavaScript, estas herramientas permiten realizar pruebas automáticas para detectar errores, verificar el comportamiento esperando y asegurar la interacción correcta unitarias, de integración, de aceptación, funcionales y rendimiento, que pueden realizarse en las aplicaciones web.



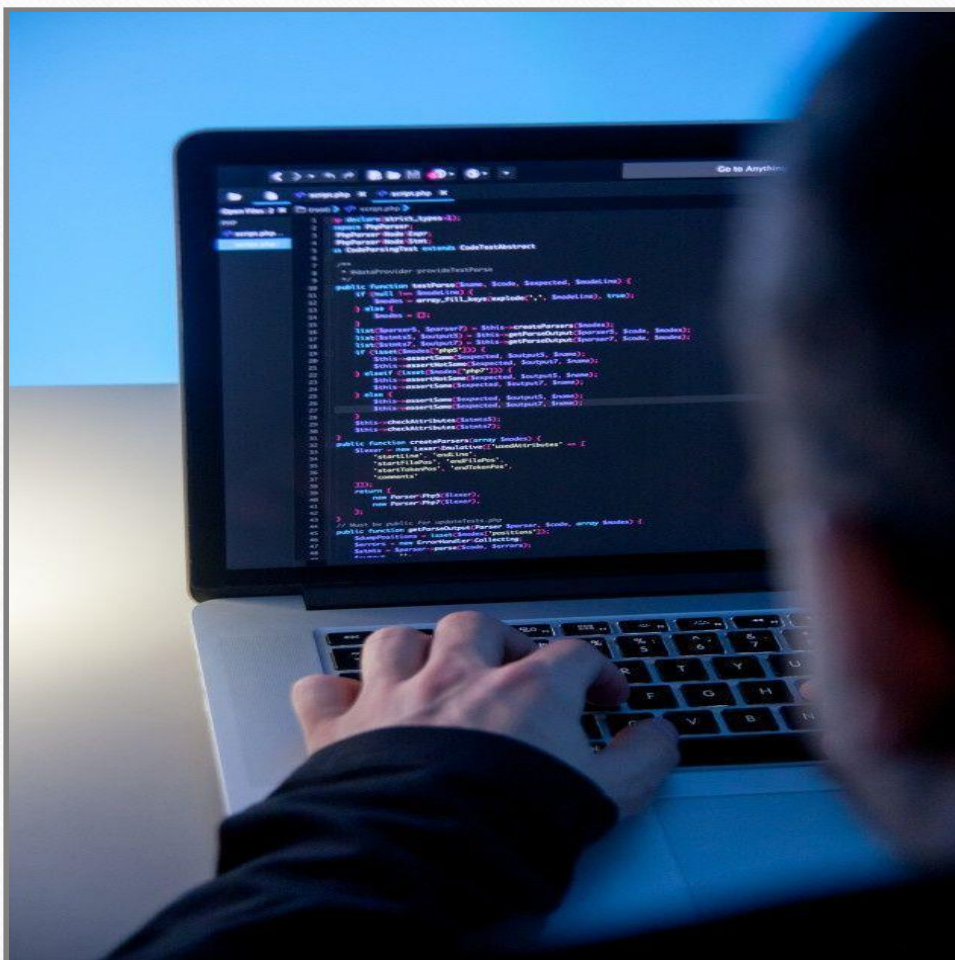
Descripción detallada

Selenium es una herramienta de software que se utiliza para automatizar navegadores web. Ya que permite realizar pruebas automáticas a aplicaciones web, pero también se puede usar para tareas repetitivas de interacción con un navegador, como la extracción de datos, la automatización de formularios y la simulación de interacciones de usuario.



Usos más comunes de Selenium

- **1.Pruebas automatizadas:** se usa Selenium para probar aplicaciones web, asegurando que funcionen correctamente después de realizar cambios en el código.
- **2.Web Scraping:** Selenium se puede usar para extraer información de sitios web, en especial en sitios web en donde el contenido se carga dinámicamente mediante JavaScript.



Selenium cuenta con diferentes componentes principales que ayudan en la automatización.

- **1.Selenium Web Driver:** herramienta principal que permite controlar el navegador de manera programática. Puedes escribir scripts en lenguajes como java, Python, c#, Ruby, etc.
- **2.Selenium IDE:** es un entorno de desarrollo integrado para Selenium, que permite grabar, editar y depurar pruebas en el navegador.



Selenium cuenta con diferentes tipos de pruebas que puede soportar tales como

- **Pruebas funcionales:** verifican que las funcionalidades de la aplicación se comporten correctamente, simulando la interacción del usuario con la interfaz (clics, entrada de texto, navegación etc.).
- **Pruebas de regresión:** aseguran que nuevas actualizaciones o cambios en el código no afecten funcionalidades ya existentes.
- **Pruebas de integración:** verifican la correcta integración entre componentes de la aplicación web.



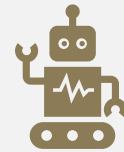
Cuenta con las siguientes integraciones disponibles como

- **Jenkins:** Integración continua para ejecutar pruebas automáticas después de cada despliegue.
- **Junit/TestNG:** Frameworks de testing en Java para estructurar y ejecutar casos de prueba.
- **BrowserStack y Sauce Labs:** Plataformas de testing en la nube que permiten ejecutar pruebas en una variedad de dispositivos y navegadores.

Comparativa



1.Cypress: Es una herramienta moderna diseñada específicamente para aplicaciones JavaScript.



2.Puppeteer: Una biblioteca de automatización de navegador para Google Chrome y Chromium.



3.Playwright: Una herramienta de Microsoft que permite la automatización de navegadores como Chromium, Firefox y WebKit.



-
- Aunque herramientas como Cypress ofrecen una mayor simplicidad en la configuración y pruebas más rápidas debido a su arquitectura más moderna, Selenium sigue siendo la opción preferida para proyectos grandes que requieren pruebas en múltiples entornos.

Mejores prácticas y consideraciones

1. Compatibilidad con múltiples navegadores: si la aplicación necesita ser probada en diversos navegadores, Selenium es una opción adecuada debido a su amplia compatibilidad.

2. Complejidad del proyecto: esto es para proyectos simples o para aplicaciones JavaScript modernas, herramientas como Cypress o Playwright pueden ofrecer una experiencia más sencilla y rápida.



3.Facilidad de integración en CI/CD: Si es crucial para tu equipo tener una integración sólida en un pipeline de integración continua (como Jenkins), Selenium es ampliamente soportado y probado en este ámbito.



4.Curva de aprendizaje: Herramientas como Selenium requieren más configuración y experiencia técnica, mientras que Cypress o Selenium IDE son más accesibles para principiantes.



5.Entornos distribuidos: Si necesitas ejecutar pruebas en paralelo en diferentes máquinas o navegadores, Selenium Grid ofrece soluciones avanzadas para pruebas distribuidas.

Conclusiones

- Selenium es una herramienta poderosa y versátil, adecuada para proyectos grandes que requieren compatibilidad multiplataforma y pruebas distribuidas, aunque puede no ser la opción más eficiente en términos de velocidad para proyectos más pequeños.

¡MUCHAS GRACIAS POR SU ATENCIÓN PRESTADA!



Link al repositorio GitHub:

- <https://github.com/Manuelguerrero234/api-auth.git>

¡Si deseas conocer mas del tema te recomendamos las siguientes páginas donde puedes conocer mas del tema y otros temas relacionados alas herramientas testing !

- www.youtube.com/@QualityStream
- www.youtube.com/@GeekQA
- www.youtube.com/@JoanMediaES
- www.youtube.com/@joanmedia