

## HERRAMIENTAS DE TESTING DE SOFTWARE DEL 2024

### ¿Por qué es importante elegir las herramientas correctas?

Da una mirada a la siguiente lista

- **Reducción de costos:** invertir dinero en las herramientas de automatización adecuadas a la larga puede reducir los costos. Al utilizar estas herramientas automatizadas de QA testing, los costos pueden reducirse, ya que ayudan a disminuir la cantidad de pruebas manuales a ejecutar. Si bien, invertir dinero y emplear este tipo de herramientas implica un costo inicial, con el tiempo, la inversión puede generar una recompensa significativa.
- **Testing integral:** para reducir significativamente la presencia de errores o defectos en un software, es esencial probar la totalidad de sus componentes. Esto se puede lograr utilizando herramientas de QA testing que proporcionen una cobertura de pruebas completa.
- **Escalabilidad:** en la mayoría de los negocios, el tamaño del software aumenta a medida que se agregan más funcionalidades a su entorno. Por lo tanto, las herramientas de testing de software deben poder gestionar las cargas de trabajo y adecuarse a las necesidades del proyecto. También es crucial indagar sobre la pila tecnológica de la herramienta de testing, para verificar que pueda integrarse correctamente con la de nuestro software. Puede incluir lenguajes de programación, servicios en la nube o arquitectura de aplicaciones.
- **Soporte para varios tipos de testing:** muchos softwares necesitan una variedad de técnicas para ser probados. Estas pueden incluir pruebas de UI, para testear la interacción con el usuario final; pruebas de seguridad, para identificar vulnerabilidades subyacentes y pruebas de performance para testear la confiabilidad del software. Por lo tanto, es importante seleccionar y utilizar herramientas capaces de ejecutar distintos tipos de pruebas.

## HERRAMIENTAS DE TESTING

Estas herramientas garantizan la eficacia y calidad del desarrollo de software ofreciendo diversas funcionalidades. Incorporando estas herramientas a tu SDLC, puedes realizar pruebas de control de calidad en aplicaciones web, pruebas de API, evaluación de performance y pruebas de carga. Esto puede aumentar la productividad y mejorar los estándares de calidad en el sector de desarrollo de software.

### 1.Selenium (Gratis)

Selenium es una reconocida plataforma de testing de automatización web, de código abierto, compatible con varios lenguajes de programación y diferentes navegadores, lo que la hace accesible para un amplio espectro de programadores. Sin embargo, Selenium por sí sola resulta insuficiente para testear aplicaciones móviles o de escritorio, es necesario integrarla con otros frameworks para potenciar sus funcionalidades.

Características

- Selenium IDE: complemento de navegador que permite grabar y reproducir casos de prueba.
- Selenium Grid: permite la ejecución de pruebas en paralelo en múltiples dispositivos y navegadores.

- Proporciona flexibilidad en la creación de scripts de pruebas y puede mejorarse con numerosos complementos.
- Cuenta con una versión actualizada de WebDriver, así como con Selenium Grid y funciones adicionales de gestión de ventanas.
- Extensibilidad: para trabajar con otros complementos y frameworks como TestNG, JUnit y Cucumber, proporcionando así soporte a muchos otros módulos.

Conoce más sobre [Selenium](#).

## **2. Playwright**

Playwright es una biblioteca Node.js de código abierto, diseñada para asistir a los desarrolladores en la automatización de las interacciones de navegadores web. Ofrece una gran variedad de funcionalidades que ayudan a cumplir con las distintas necesidades de las pruebas.

Proporciona un amplio conjunto de herramientas para la automatización de navegadores y facilita las pruebas de aplicaciones web en diferentes navegadores y plataformas. A su vez, admite modernas funcionalidades web, garantizando así que las pruebas ejecutadas sean correctas y reflejen la experiencia real de usuario.

### **Características**

- Contextos: puede crear sesiones de navegador aisladas, posibilitando la realización de pruebas en paralelo.
- Compatible con varios navegadores: funciona en Chromium, Firefox y WebKit.
- Espera automática: espera automáticamente a que los elementos estén listos para ejecutar las acciones.
- APIs sólidas: APIs con multifuncionalidades en interacciones, navegación y validación de elementos que interactúan con las páginas.
- Compatible con distintos lenguajes: JavaScript, Python, .NET, Java.

Descubre el [Template de Playwright](#).

## **3. JMeter (Gratis – Código abierto)**

JMeter es la herramienta más utilizada para pruebas de carga y performance de aplicaciones web. Se trata de un software de código abierto diseñado para medir la performance y testear el comportamiento de las funcionalidades. JMeter ayuda a simular cargas pesadas en servidores, grupos de servidores, redes u objetos. Puede testear la robustez o analizar la performance.

### **Características**

- Protocolos: HTTP, HTTPS, FTP, JDBC, LDAP, SOAP, JMS, entre otros.
- Grupos de subprocesos: simulan el comportamiento de conexiones simultáneas de múltiples usuarios.
- Plan de pruebas: ayuda a crear un plan de pruebas flexible utilizando muestreadores, controladores lógicos, oyentes y temporizadores.

- Testing de carga: este tipo de pruebas de performance define cómo se comportará el sistema bajo la carga ejercida por la propia aplicación.

Descubra cómo realizar [pruebas de rendimiento distribuido con JMeter](#)

#### **4. SoapUI**

SoapUI permite a los usuarios realizar pruebas automatizadas de distintos tipos, funcionales, de regresión, de cumplimiento y de carga, contra diversas APIs Web. Proporciona soporte para todo tipo de estándares, protocolos y tecnologías; y contribuye a garantizar la posibilidad de realizar pruebas integrales contra cualquier tipo de API. Los usuarios pueden utilizar fácilmente sus funcionalidades ya que cuenta con una interfaz de usuario intuitiva, tanto para usuarios técnicos como no técnicos.

##### Características

- Soporte para pruebas SOAP y REST: soporte completo para servicios web SOAP y RESTful.
- Simulación de servicios: permite modelar servicios simulados para testarlos en caso en que los servicios reales no estuvieran disponibles.
- Testing basado en datos: utiliza fuentes de datos externas como Excel, XML y bases de datos para ejecutar data testing.
- Testing de seguridad: detección de vulnerabilidades mediante escaneos de seguridad contra XSS, SQL Injection y más.
- Testing automatizado: ejecución de pruebas automatizadas y lógica personalizada programadas en Groovy.
- Integración: esta herramienta tiene gran compatibilidad con herramientas CI/CD como Jenkins y Maven.

#### **5. Jira**

Jira es una herramienta de gestión de proyectos con algunas funcionalidades potentes de testing, como el seguimiento de incidentes, elaboración de informes y tableros de tareas. Puede integrarse fácilmente con la mayoría de las herramientas de testing automatizado, lo que acelera el proceso de optimización del testing y potencia el aumento de la productividad.

Esto le permite a Jira integrarse con herramientas de testing automatizado como Selenium, JUnit y TestNG. Potencia el testing, automatizando y gestionando las pruebas relativamente más rápido; y permite la normalización y el seguimiento de los cambios, garantizando que las pruebas acompañen el ciclo de vida del desarrollo.

##### Características

- Flujos de trabajo personalizados: crea y personaliza flujos de trabajo para adaptarlos a los requisitos del proyecto.
- Soporte para tableros ágiles: la técnica de gestión de proyectos ágiles admite tableros Scrum y Kanban.
- Modo autónomo: puede ejecutarse incluso sin el servidor Selenium, directamente con el navegador.

- Soporte de framework: el desarrollo orientado al comportamiento (BDD) se realiza utilizando Mocha, Jasmine y C.
- Servicios: servicios preconstruidos para Selenium Grid, Appium y Docker.
- Regresión visual: funciona con las herramientas actuales de testing de regresión visual.

## **6. Docker**

Docker es una plataforma de código abierto utilizada para desplegar automáticamente aplicaciones en contenedores portátiles livianos. Proporciona entornos estructurados y aislados en los que las pruebas se ejecutan de forma similar en todas las fases de desarrollo.

Los contenedores Docker encapsulan todo en su interior, por lo tanto, nunca habrá discrepancias entre la configuración del entorno de desarrollo y la configuración del entorno de producción. Es más, la capacidad de Docker garantiza habilidades de CI/CD realmente fluidas para el testing y el despliegue continuo.

Características

- Contenedores: almacenaje de aplicaciones y sus dependencias en contenedores.
- Docker Compose: definición y ejecución de aplicaciones de varios contenedores con archivos compose.
- Gestión de imágenes: capacidad para adjuntar capas de imágenes, almacenarlas y distribuirlas.
- Interconexión: funciones de red avanzadas mediante redes personalizadas y descubrimiento de servicios.

## **7. GitHub**

GitHub es una de las herramientas más reconocidas para el control de versiones y el trabajo colaborativo. Además, la herramienta en sí misma aporta una completa funcionalidad de pruebas de automatización mediante GitHub Actions. Con ella, los desarrolladores y testers pueden automatizar miles de tareas repetitivas relacionadas con las pruebas, la creación y el despliegue de sus aplicaciones.

Al utilizar GitHub se puede definir, en sintaxis YAML, la secuencia de pasos y condiciones en las que se desea que se ejecuten las pruebas. Es decir, se puede llevar a cabo el proceso de automatización en base a las necesidades del proyecto.

Características

- GitHub Actions: plataforma de integración (CI/CD) para testing y despliegue.
- Repositorios: cantidad ilimitada de colaboradores en repositorios públicos y privados.
- Seguridad: ofrece avanzadas funciones integradas de seguridad, como Dependabot, escaneo de códigos y gestión de confidencialidad.
- Integraciones: sólida integración con servicios y herramientas externas.

[perfil de GitHub](#)

## **8. Postman**

Postman es una interesante herramienta automática para testing de API. Presenta una interfaz de usuario amigable para crear, verificar, parametrizar y analizar las solicitudes de pruebas. Permite automatizar pruebas recurrentes y confiables y utilizarlas en diversos escenarios.

### Características

- Tiene una gran variedad de bibliotecas integradas en los scripts de prueba y de pre-solicitudes.
- Proporciona un sólido soporte para GraphQL.
- Almacena datos para múltiples pruebas.

[cómo usar Postman para las prueba de API.](#)

## **9. Jenkins**

Jenkins es un servidor de código abierto, que les permite a los equipos de ingeniería automatizar el proceso completo de desarrollo de software. Abarca desde la construcción y el testing de aplicaciones, hasta su despliegue. Es altamente extensible, con cientos de plugins que dan soporte al ecosistema de herramientas en torno a integración y entrega continuas (CI/CD).

### Características

- Builds distribuidos: los builds se ejecutan en diferentes nodos para equilibrar la carga.
- Altamente flexible: mantiene la flexibilidad con plugins personalizados e integraciones.
- Notificaciones: perfectamente integrado con correo electrónico y Slack, junto con cualquier otro sistema de notificación.
- Integración de control de fuentes: compatible con Git, SVN, Mercurial, etc.
- Crea disparadores: Poll SCM, webhook y time-based builds.

## **10. WebDriverIO**

WebDriverIO (WDIO) es un framework de pruebas de automatización gratuito y de código abierto, muy popular, desarrollado para Node.js. Permite a desarrolladores y testers interactuar fácilmente con los navegadores de forma automatizada y ejecutar pruebas automáticas mediante múltiples navegadores y dispositivos.

WDIO facilita la creación y ejecución de pruebas con Selenium haciendo posible el desarrollo de suites de pruebas estables y sostenibles, de forma sencilla y bastante rápida.

### Características

- Pruebas entre navegadores: funciona en Chrome, Firefox, Safari y Edge.
- Integración de framework: perfectamente integrado con Mocha, Jasmine y Cucumber para soportar una escritura de pruebas más flexible.

- Sincronización automática: espera automáticamente a que un elemento esté disponible para ejecutar finalmente la acción.
- Ejecución en paralelo: permite ejecutar casos de prueba en paralelo.
- Testing de regresión visual: se integra con cualquier herramienta de regresión visual para garantizar la consistencia en la interfaz de usuario.
- Integración CI/CD: se integra muy fácilmente con la mayoría de los pipelines CI/CD, a través de los cuales realiza pruebas automáticamente.
- Comandos personalizados: escribe y controla tus comandos personalizados para ampliar las funcionalidades de WebDriverIO.
- Pruebas de API: las pruebas de API, independientemente y de la mano de las pruebas de UI, amplían la cobertura según los testers.
- **11. ZAP** (Gratis)

### Lo mejor para realizar pruebas de seguridad

Mientras llevaba a cabo mi evaluación, descubrí que ZAP no es sólo una herramienta de prueba ordinaria sino una parte importante del OWASP Iniciativa para mejorar la seguridad web. Su detección automatizada de vulnerabilidades y su soporte para ataques éticos me impresionaron mucho.

ZAP puede actuar como proxy para el navegador y el servidor de aplicaciones. También se puede utilizar como un proceso de solicitud independiente sin interfaz de usuario.

#### Características:

- **Análisis y Modificación de Tráfico:** El proxy ZAP ayuda a analizar, cambiar e inyectar tráfico en el contenido del mensaje entre el navegador y el servidor.
- **Integración de seguridad CI/CD:** El complemento Jenkins permite integrar pruebas de seguridad dentro del proceso de CI/CD.
- **Accesibilidad del usuario:** ZAP es una de las mejores herramientas de prueba para pruebas manuales que es apropiada tanto para especialistas en seguridad experimentados como para desarrolladores web.
- **Pruebas de seguridad ética:** Fuzzer, centrado en la seguridad, prueba aplicaciones en estados autoritarios con cargas útiles éticas.
- **Automatización de tareas:** Admite la automatización de diversas tareas para una mayor confiabilidad y ahorro de tiempo. Explorar el [principales herramientas de gestión de pruebas](#) para obtener ayuda.
- **Usado para:** Pruebas de seguridad.
- **Plataformas compatibles:** Windows, Linux y macOS.