

**“Universidad Tecnológica De Torreón”
“T.S.U Tecnologías de la Información y desarrollo de Software
Multiplataforma”
“Abel Sosa Escobedo”
“08/10/2020”**

**-Luis Manuel Ramirez Roque-19170169-4B-
-Miguel Ángel Esparza Villanueva-19170052-4B-
-Dana Stephanie Lopez Lopez-19170116-4B-**

Temas a tratar

- Pruebas de aplicaciones WEB
- Pruebas de regresión humo y usabilidad
- Pruebas de regresión
- Pruebas de humo
- Pruebas de usabilidad
- Herramientas de Defectos
- Objetivos de la herramienta
- Como utilizar la herramienta en los diferentes niveles organizaciones, ----- -
- Características y ejemplos

Introducción

Es una breve introducción sobre las aplicaciones web, los tipos de prueba los cuales son pruebas de Regresión, pruebas de Humo y pruebas de Usabilidad, también sobre las diferentes herramientas de organizaciones y sus características.

Aplicaciones Web

- Pruebas de software en aplicaciones web

En el caso de las aplicaciones web, además de las pruebas tradicionales, es necesario comprobar el funcionamiento del servidor web, SGBD, red y el conjunto del sistema. Tanto de manera funcional como realizando un test de carga para comprobar la capacidad del mismo.

En este caso, algunas herramientas básicas para el desarrollador son:

Herramientas de validación de código html/css

No conviene olvidar que en las aplicaciones web la comunicación entre el servidor y el cliente se realiza mediante [HTML](#)/[CSS](#). Una cuestión fundamental para garantizar que el cliente va a poder representar la interfaz de la aplicación de la manera correcta es que los datos que el servidor envía sean válidos. Por ello, en cualquier aplicación web se deberá validar todo el HTML/CSS transmitido al cliente.

Para ayudar en dicha tarea se pueden utilizar diferentes validadores como el del

W3C <http://validator.w3.org/>

Aplicaciones Web

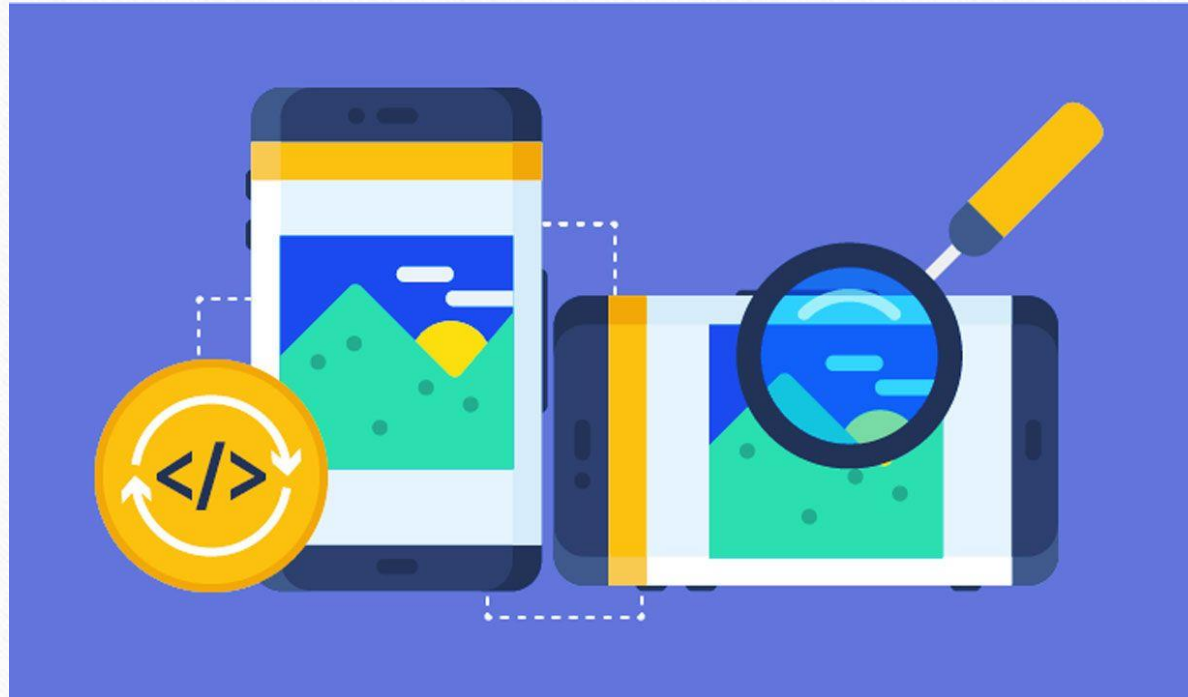
- **Herramientas para poner a prueba el servicio**
- En una aplicación web es vital conocer la capacidad del sistema en cuanto a número de clientes y concurrencia de los mismos. Por ello resulta necesario simular un enjambre de clientes que ponga a prueba la aplicación para poder valorar su rendimiento, o que realice en ella ciertas operaciones que se quieren poner a prueba.
- ApacheBench (comando **ab**) es un pequeño programa que acompaña a la distribución de Apache y permite realizar peticiones de forma repetida y concurrente para valorar el rendimiento del mismo.

Pruebas de regresión

- Cualquier tipo de pruebas de software con el objeto de descubrir errores (bugs), carencias de funcionalidad, o divergencias funcionales con respecto al comportamiento esperado del software, causados por la realización de un cambio en el programa. Se evalúa el correcto funcionamiento del software desarrollado frente a evoluciones o cambios funcionales.

<https://www.youtube.com/watch?v=OEcIXnU6EQw>

Pruebas de regresión



Pruebas de Humo

Las **pruebas de humo** (smoke testing) son una revisión rápida de un producto de software para comprobar que funciona y no tiene defectos evidentes que interrumpen la operación básica del mismo.

Son pruebas que pretenden hacer una evaluación inicial de la calidad de un producto de software.

<https://www.youtube.com/watch?v=W3tYXzJiyBE>

Pruebas de Humo

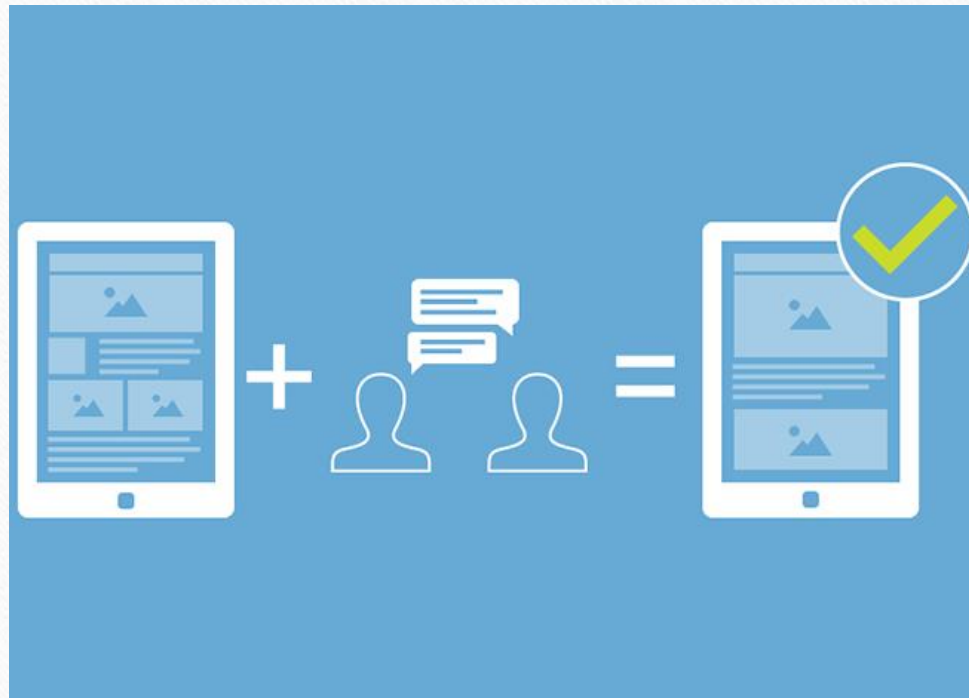


Pruebas de Usabilidad

- La **prueba de usabilidad por parte del usuario** es una técnica usada en el diseño de interacciones centrado en el usuario para evaluar un producto mediante pruebas con los usuarios mismos.
- Se enfocan en medir la capacidad de un producto de fabricación humana en cumplir el propósito para el cual fue diseñado.

<https://www.youtube.com/watch?v=uh4sZmQiRr4>

Pruebas de Usabilidad



Herramientas de defectos

- La implementación de la idea comienza con una planificación clara de todo el proceso. *Dueño del producto* o *Gerente de proyecto* tiene que seleccionar el mejor equipo, otorgar las responsabilidades y asignar tareas, y elegir las herramientas de gestión de pruebas que se utilizarán durante todo el ciclo de vida del desarrollo de software.

Herramientas de defectos



Objetivos de las herramientas

- Sistema de gestión de pruebas – Crear y gestionar proyectos, miembros.
- Oportunidad de planificación de pruebas – Creación de planes de prueba, casos de prueba, escenarios de prueba o historias de usuario.
- Disponibilidad de ejecución de prueba – Creación de ejecuciones de prueba, gestión de los miembros del proceso de prueba, facilidad de uso y eficiencia.
- Fabricación informes de las pruebas En calidad y cantidad de trabajo.
- Herramienta de gestión de tareas – La disponibilidad del rastreador de errores y su velocidad.
- Utilizando características adicionales – Importación / exportación, integración con otros sistemas, análisis de fallas, etc.

¿Como utilizar la herramienta en los diferentes niveles organizacionales?

- Describir diferentes tipos de Casos de prueba
 - Crear un equipo de prueba, asignarlo a un probador y nombrar fechas exactas de pruebas
 - Reportar un defecto, un mejora o un tarea
 - Cubrir Los requisitos del cliente (o los suyos propios) con casos de prueba.
- ejecutar pruebas usando ejecuciones de prueba

Características

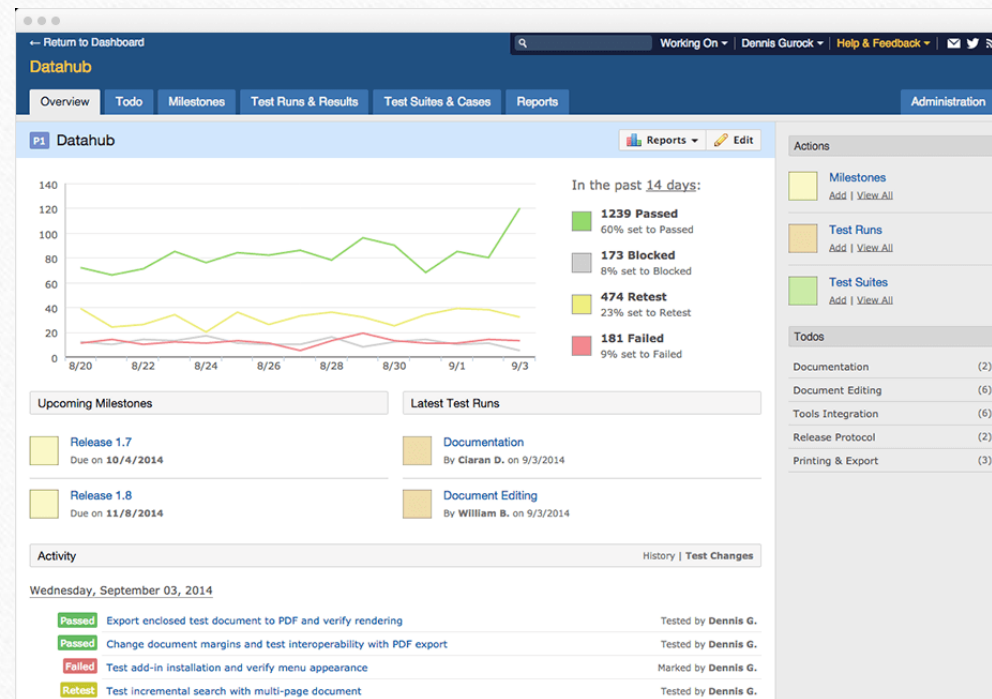
- Proporciona una interfaz simple y común para compartir archivos y comunicar errores.
- Proporciona notificaciones y registros para ayudar al equipo a seguir, controlar y estimar el trabajo relacionado con errores.
- Proporciona una base de datos de errores que el equipo de desarrollo pueda usar como referencia en el futuro
- Automatiza las tareas manuales asociadas con la captura de errores y la actualización de problemas

Ejemplos

TestRail: La herramienta tiene una interfaz fácil de usar, una disposición de botones intuitivamente comprensible y un sistema de cambio entre características contiguas.

- Ejecutar pruebas basado en escenarios de prueba creados
- Crear un error y subirlo a rastreadores de errores integrados como *JIRA*, *GitHub*, *YouTrack* y otros
- Personalizar El sistema a su medida utilizando el abierto *API de TestRail*

Ejemplo

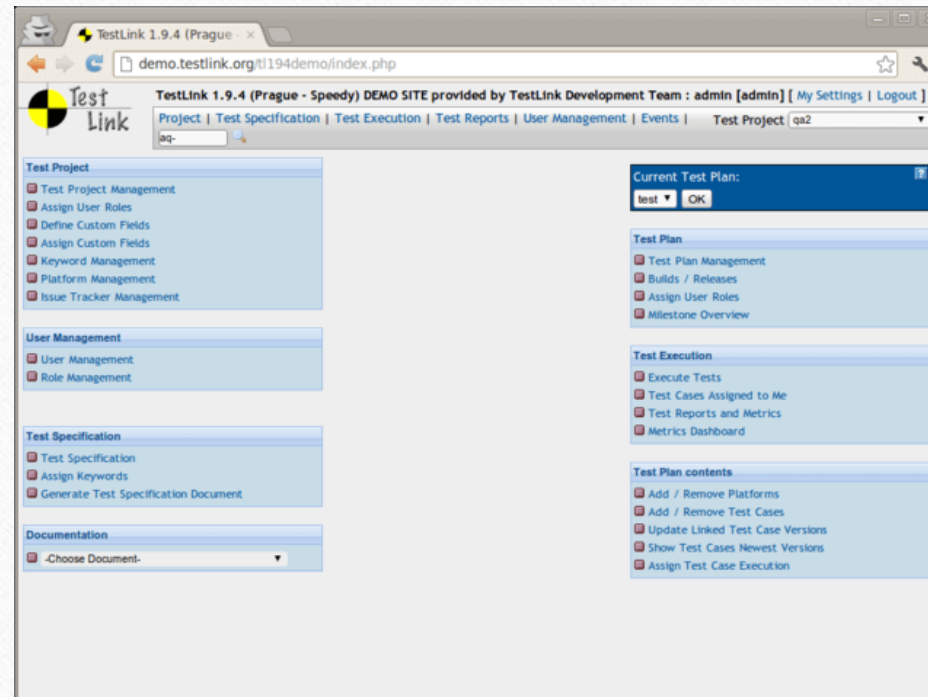


Ejemplos

TestLink: Es un proyecto de código abierto en la lista de herramientas de gestión de pruebas conocidas. Tiene una interfaz simple y un diseño «técnico» sin lujos.

- Crear Casos de prueba Sobre la base de estos requisitos.
- Cubrir los requisitos del cliente (o los suyos propios) con casos de prueba.
- Recibir el informe una vez finalizada la prueba

Ejemplo





Ejemplos

- **JIRA + Zephyr:** *Jira* tiene algunas opciones para los servicios de escritura de casos de prueba, sin embargo, es el mejor y perfectamente integrado rastreador de errores para *Céfiro*.
- Capaz de crear un Plan de prueba
- Describir Casos de prueba
- Ejecutar pruebas
- Generar informes
- Si un producto de prueba se comporta mal, puede crear un error

Ejemplo

The screenshot displays the JIRA IronClad interface. At the top, a dark blue navigation bar contains the JIRA logo, followed by links for Dashboards, Projects, Issues, Tests, and a 'Create issue' button. Below this, the IronClad project header shows the project key 'IC' and the lead 'Zephyr Admin'. The main content area is divided into two sections: 'Overview' and 'Administration'. The 'Overview' section on the left lists links for Summary, Issues, Road Map, Change Log, Test Summary (which is highlighted), and Test Cycles. The 'Administration' section on the right is titled 'Test Summary' and lists 'Total Tests', 'Total Executed', and 'Total Remaining'. A dropdown menu is open from the 'Tests' link in the top navigation bar, showing options such as 'Search Tests', 'Create a Test', 'Plan Test Cycle', 'Search Test Executions', 'Manage Execution Filters', 'Execute Tests', 'Test Summary', 'Test Metrics', 'Welcome', 'Zephyr Help', and 'About Zephyr'.

JIRA Dashboards ▾ Projects ▾ Issues ▾ Tests ▾ Create issue

 **IronClad**
Key: IC · Lead:  Zephyr Admin

Overview Administration

Summary
Issues
Road Map
Change Log
Test Summary
Test Cycles

Test Summary

Test Summary
Total Tests
Total Executed
Total Remaining

- Search Tests
- Create a Test
- Plan Test Cycle
- Search Test Executions
- Manage Execution Filters
- Execute Tests
- Test Summary
- Test Metrics
- Welcome
- Zephyr Help
- About Zephyr

Conclusión

(Luis Manuel Ramirez Roque)

- Yo siempre he pensado que antes de entregar algo ya sea una actividad, un programa, o algo, debemos de probarlo antes para nosotros estar seguros de que es trabajo va funcionando al 100%, pero no sabía que estas pruebas tenían nombre, o se les identifica de una forma, diferente a cada una, ya sea por ejemplo con la caja negra y la caja blanca donde los dos son pruebas pero una es al código interior y la otra es a lo que se le muestra al cliente, y así como estas existen muchas pruebas mas que esas, para todo tipo.

Conclusión

(Dana Stephanie López López)

Existen diferentes herramientas como lo son las pruebas de regresión, las pruebas de humo estas se encargan de ver que el software no tenga fallos, que sea funcional y que cumpla con el propósito para el cual fue diseñado.

Por otra parte existen las herramientas de defectos que nos facilitan las pruebas de nuestro software y otra que también es muy importante son las pruebas de usabilidad, ya que el mismo usuario es el encargado de comprobar que realmente sea lo que se pidió.

Conclusión

(Miguel Ángel Esparza Villanueva)

- En este nuevo tema donde investigamos más tipos de casos de prueba pudimos observar que cada caso tiene un objetivo y distintas características que nos ayudan para que dentro de un proyecto y que se debe de escoger el adecuado para poder entregar un proyecto bien hecho. Dentro de estos métodos podremos realizar proyectos seguros, confiables y que sean funcionales para que el cliente tenga un software de acuerdo a las especificaciones que nos den.