



Material RAP - 3

Manejo de pruebas de software

Gestión de cambios y ajustes en la documentación de pruebas

Tabla de Contenido

1. Introducción	3
2. Estructura de contenido	4
3. Gestión y control de la ejecución de pruebas	5
3.1. Alistamiento	5
3.2. Documentación.....	6
4. Herramientas de soporte de pruebas de software	8
4.1. Tipos	8
4.2. Ejemplos.....	10
5. Material de apoyo	22
6. Glosario	23
7. Referencias bibliográficas	23
Créditos	24
Creative Commons	24

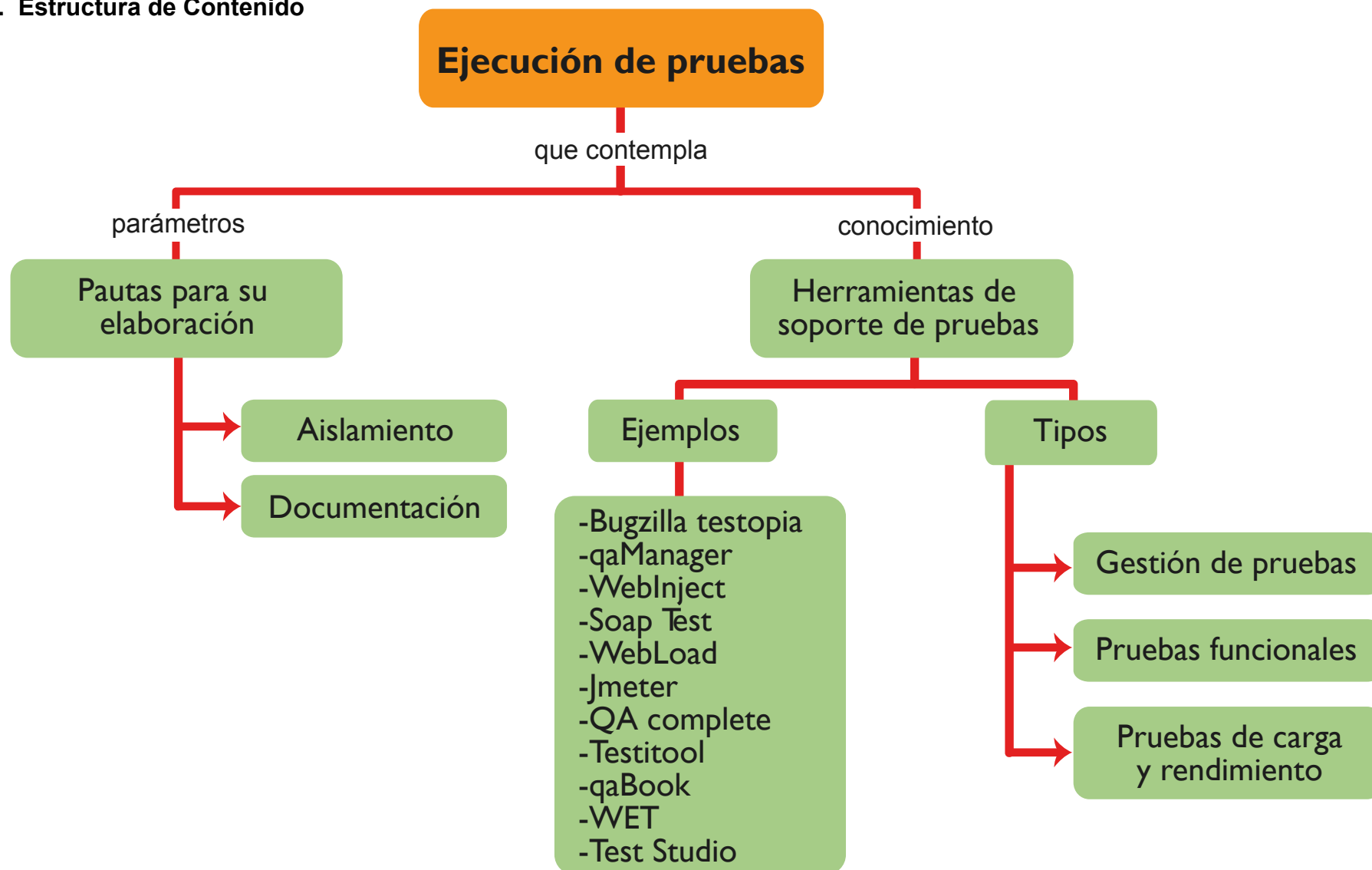
1. Introducción

La aplicación de pruebas de software, requiere tener como precedente la planeación del alcance de estas, respecto al equipo necesario que las ejecutará y el tiempo asignado para esto, a partir de los casos de prueba que han sido diseñados. De esta manera, los casos de prueba serán los documentos que se diligenciarán durante este proceso. Adicionalmente, es necesario generar un informe con el resumen de las pruebas que permita identificar el estado del proceso realizado para tomar las acciones del caso en la siguiente fase de ajustes y de cambios.

Teniendo en cuenta lo anterior, en este resultado de aprendizaje se presentarán algunos aspectos a tener en cuenta al momento de ejecutar las pruebas y las técnicas de soporte de Pruebas de software que facilitarán esta actividad.



2. Estructura de Contenido



3.1. Gestión y control de la ejecución de pruebas

El proceso de gestión de la ejecución de pruebas, contempla el alistamiento para la ejecución de pruebas, su seguimiento, toma de decisiones y cierre de ciclo de pruebas. Para esto a continuación se presentarán algunas acciones que se deben realizar.

3.1. Alistamiento

Es necesario tener listo lo que se requiere por parte del equipo asignado para la ejecución de pruebas con relación a:

- **Datos**

Los datos del software pueden ser datos generados, datos previamente guardados, datos aleatorios (usados en pruebas de carga) y en general todos los datos que sean necesarios ingresar o visualizar en el software, por eso es necesario tenerlos listos y configurados en el sistema.

- **Software**

A nivel del software es necesario realizar la configuración de los servidores, el alistamiento de la información de pruebas en la base de datos (instancia exclusiva para las pruebas), si se requiere, el alistamiento de máquinas virtuales, servidor web y simuladores.

- **Herramientas de soporte**

De igual forma, se requiere realizar la configuración de las herramientas de software que se desean utilizar.

- **Hardware**

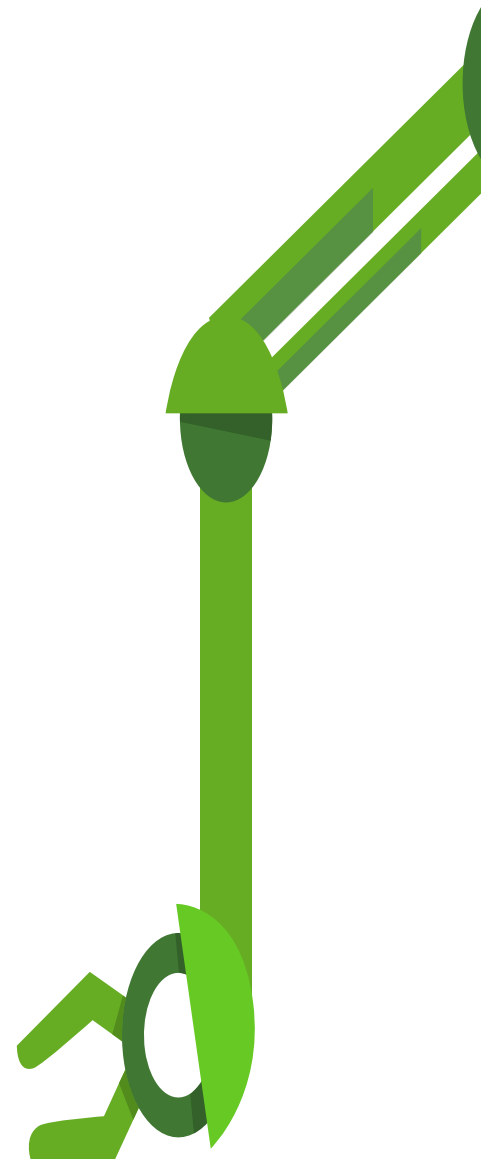
Respecto al hardware es necesario tener listo lo que se requiera a nivel de disco duro, procesador, memoria, dispositivos multimedia, entre otros.

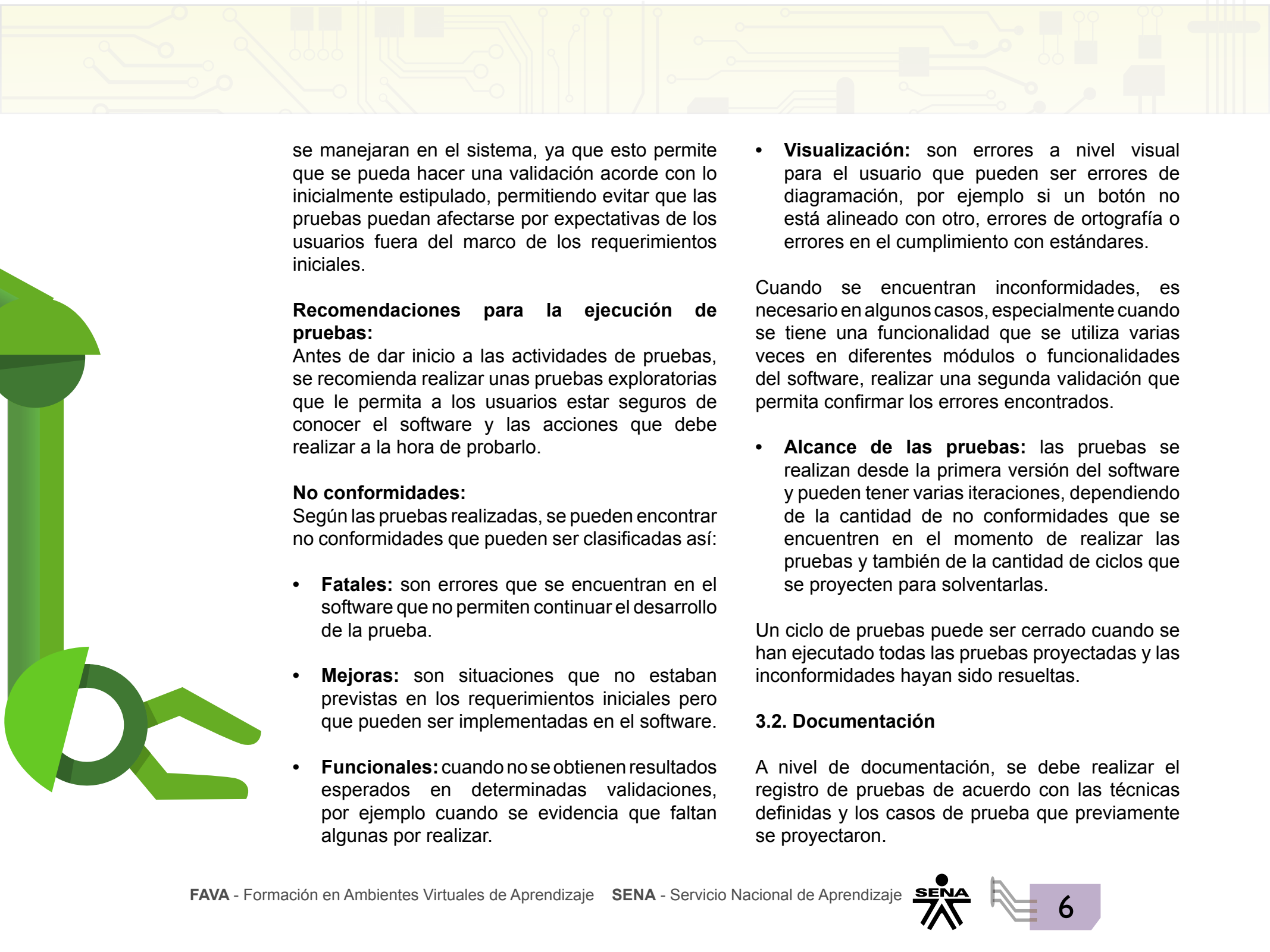
- **Cronograma de ejecución de pruebas**

De acuerdo con lo que se estipuló en el plan de pruebas, se hace necesario tener el detalle de las pruebas que se van a realizar según los casos de prueba identificados y las técnicas definidas que se implementarán.

Por otra parte, también es necesario contar con el equipo de personas que realizará las pruebas. Este equipo, debe tener dominio del software y conocer los requerimientos iniciales del proyecto a nivel técnico y funcional ya que esto puede exigir diferentes áreas de conocimiento.

Por ejemplo, cuando se hicieron las pruebas del software SOFIA Plus se requirió que el equipo que realiza las pruebas funcionales tuviera el dominio y experticia en el funcionamiento de los procesos del SENA y la manera como fueron proyectados que





se manejaran en el sistema, ya que esto permite que se pueda hacer una validación acorde con lo inicialmente estipulado, permitiendo evitar que las pruebas puedan afectarse por expectativas de los usuarios fuera del marco de los requerimientos iniciales.

Recomendaciones para la ejecución de pruebas:

Antes de dar inicio a las actividades de pruebas, se recomienda realizar unas pruebas exploratorias que le permita a los usuarios estar seguros de conocer el software y las acciones que debe realizar a la hora de probarlo.

No conformidades:

Según las pruebas realizadas, se pueden encontrar no conformidades que pueden ser clasificadas así:

- **Fatales:** son errores que se encuentran en el software que no permiten continuar el desarrollo de la prueba.
- **Mejoras:** son situaciones que no estaban previstas en los requerimientos iniciales pero que pueden ser implementadas en el software.
- **Funcionales:** cuando no se obtienen resultados esperados en determinadas validaciones, por ejemplo cuando se evidencia que faltan algunas por realizar.

- **Visualización:** son errores a nivel visual para el usuario que pueden ser errores de diagramación, por ejemplo si un botón no está alineado con otro, errores de ortografía o errores en el cumplimiento con estándares.

Cuando se encuentran inconformidades, es necesario en algunos casos, especialmente cuando se tiene una funcionalidad que se utiliza varias veces en diferentes módulos o funcionalidades del software, realizar una segunda validación que permita confirmar los errores encontrados.

- **Alcance de las pruebas:** las pruebas se realizan desde la primera versión del software y pueden tener varias iteraciones, dependiendo de la cantidad de no conformidades que se encuentren en el momento de realizar las pruebas y también de la cantidad de ciclos que se proyecten para solventarlas.

Un ciclo de pruebas puede ser cerrado cuando se han ejecutado todas las pruebas proyectadas y las inconformidades hayan sido resueltas.

3.2. Documentación

A nivel de documentación, se debe realizar el registro de pruebas de acuerdo con las técnicas definidas y los casos de prueba que previamente se proyectaron.

De esta manera, es necesario diligenciar la información de resultados obtenidos en cada caso de prueba que fue diseñado en la primera etapa del proceso de gestión de pruebas.

Informe de incidente de pruebas: en este informe es fundamental la recopilación adecuada y bien redactada de lo que evidencia el equipo de pruebas.

De esta manera se recomienda tener un informe con la siguiente información:

- Fecha de la prueba.
- Persona que realiza la prueba.
- Módulo/ funcionalidad.
- Código del caso de prueba.
- Nombre del caso de prueba.
- Resultado obtenido.

- Estado de ejecución (pasó, falló, no se pudo ejecutar).
- Estos datos son adicionales a los que ya se tengan definidos.
- En dónde se encontró el problema: unidad, módulo, API o funcionalidad del software.

Cómo: pasos con los cuales se evidenció el error.

Condiciones: qué datos se utilizaron, qué equipo o dispositivo (computador, celular) y en general con cuáles condiciones se generó el incidente.

Hallazgo: descripción de lo que se encontró, relacionar imagen el problema encontrado.

Qué se esperaba: qué estaba esperando como respuesta de la prueba en el software.

Fecha de la prueba	Persona que realiza la prueba	Módulo/ funcionalidad	Código del Caso de Prueba	Nombre del caso de prueba	Resultado Obtenido	Estado	Descripción de pasos ejecutados	Imágenes de soporte	Qué se esperaba
						Aprobado			
						No aprobado			
						No se pudo probar			
						No aprobado			

Figura 1. Ejemplo informe incidente de pruebas

Fuente: SENA

Es necesario tener en cuenta que existen herramientas de software para la gestión de pruebas que traen reportes definidos para la consolidación de incidentes, así como gráficas que permiten evidenciar esta trazabilidad del estado de las pruebas.

Figura 2. Ejemplo informe resumen de pruebas
Fuente: SENA

No Iteración	Fecha Inicio y Fin	# Pruebas realizadas	# No conformidades/ casos de prueba no aprobados	# de eventos o casos de prueba ejecutados	# de casos de prueba diseñados	# casos de prueba por ejecutar	# de eventos o casos de prueba aprobados

De esta manera, los anteriores formatos son sugerencias de la información que debe ser registrada y que permitirá la toma de decisiones en un proyecto de software, sin embargo es necesario evaluar las herramientas que pueden ser útiles en este proceso y que dependiendo de lo que se quiera analizar puede favorecer el seguimiento del proyecto.

Resumen de pruebas: de igual manera, es necesario tener un resumen de las pruebas realizadas que permita identificar desde la gestión de pruebas, el avance en porcentaje de las pruebas realizadas, el porcentaje de no conformidades encontradas, el porcentaje de pruebas aceptadas, entre otros.

Por lo tanto se sugiere tener la siguiente información:

4. Herramientas de soporte de pruebas de software

4.1. Tipos

Existen diversos tipos de herramientas de soporte de pruebas que apoyan las distintas labores, tales como:

Gestión de pruebas: estas herramientas permiten la organización de actividades de modo tal que pueda realizarse un seguimiento al proceso. Por lo tanto, en estos instrumentos se puede hacer el registro de incidentes que se vayan encontrando, permite asignar el equipo encargado de realizar

pruebas y solventar fallas, permiten usualmente generar reportes y también gráficos que facilitan la toma de decisiones de acuerdo con los resultados y avances que se van obteniendo.

Pruebas funcionales: este tipo de herramientas permiten el registro de casos de prueba y la generación de eventos que ayudan a validar las funcionalidades del software. De esta manera, dependiendo de la herramienta seleccionada, se hace necesario configurar la información, detallar los objetivos del software y aplicar las pruebas diseñadas. Usualmente contemplan la generación

de reportes que permiten ver la trazabilidad de las pruebas realizadas y los resultados obtenidos.

Pruebas de carga y rendimiento: las herramientas de carga y rendimiento permiten simular las condiciones con las cuales se puede evaluar el software en condiciones máximas de acceso y disponibilidad. Por ejemplo, se puede simular la ejecución de acciones por parte de múltiples usuarios al mismo tiempo para evaluar los niveles de respuesta, la capacidad y los tiempos que genera el software.

Algunas herramientas permiten realizar todas las pruebas anteriormente nombradas. Sin embargo de acuerdo con las necesidades del proyecto y las expectativas planteadas, se deben seleccionar aquellos instrumentos que se adapten a los requerimientos puesto que también es necesario contemplar el costo que estas puedan representar. De acuerdo con lo publicado en varios recursos de internet (Software, 2013), a continuación se listan algunas herramientas de uso libre y comercial:

Herramientas de uso libre

Gestión de Pruebas	Pruebas Funcionales	Pruebas de carga y rendimiento
<ul style="list-style-type: none"> • Bugzilla Testopia • FInese • qaManager • qaBook • RTH (open source) • Salome-tmf • Squash TM • Test environment Toolkit • TestLink • Testitool • XQual Studio • Radi-testdir • Data Generator 	<ul style="list-style-type: none"> • Selenium • Soapui • Watir • WatiN • Capedit • Canoo Webtest • Solex • imprimatur • SAMIE • ITP • WET • Webinject 	<ul style="list-style-type: none"> • Funkload • FWPTT load testing • loadUI • jmeter

Figura 3. Herramientas de uso libre
Fuente: SENA

Herramientas comerciales

Gestión de Pruebas	Pruebas Funcionales	Pruebas de carga y rendimiento
<ul style="list-style-type: none"> • HP Quality Center/ALM • QA Complete • qaBook • T-Plan Professional • SMARTS • QAS.Test Case Studio • PractiTest • SpiraTest • TestLog • ApTest Manager • Zephyr 	<ul style="list-style-type: none"> • QuickTest Pro • Rational Robot • Sahi • SoapTest • Test Complete • QA Wizard • Squish • vTest • Internet Macros 	<ul style="list-style-type: none"> • HP LoadRunner • LoadStorm • Neo Load • WedLOAD Professional • Forecast • ANTS - Advanced.NET Testing System • Webserver Stress Tool • Load Impact

Figura 4. Herramientas comerciales
Fuente: SENA

4.2. Ejemplos:

Existen diversas herramientas que facilitan las actividades que se realizan al momento de realizar pruebas al software. Esto permite automatizar actividades que agilizan algunas actividades al interior de este proceso. De esta manera se encuentran herramientas gratuitas y pagas. Cada una tiene beneficios, a continuación se describen algunas de estas herramientas:

Bugzilla

Es una herramienta de gestión de pruebas para el reporte de errores. Es un software gratuito que

facilita el registro de errores por parte del equipo de pruebas y la visibilidad y acceso por parte del equipo desarrollador y administrativo del proyecto que permitan dar solución a lo reportado de manera eficiente.

Esta herramienta permite indicar la prioridad de cada error y solicita indicar el detalle del software, así como los responsables de la ejecución de pruebas, el tiempo estimado para la resolución de cada problema, el detalle de cada bug, adjuntar un archivo si es necesario, asignar a una persona encargada de solucionarlo, entre otras cosas.

Croma Campus - Bugzilla Tutorial.pdf - Adobe Reader

File Edit View Window Help

Open [Icons] 6 / 21 89% [Icons] Tools Fill & Sign Comment

Reporter: nurusama@soberit.hut.fi

Version: other

Product: TestProduct

Component: TestComponent

Platform: PC

OS: Windows 2000

Priority: P5

Severity: trivial

Assigned To: juhitko@cc.hut.fi (Leave blank to assign to default component owner)

Cc:

URL: http://

Summary: Splash screen has a typo "Copyrigt"

Description: Splash screen displays a text "Copyrigt ...", should be "Copyright"

Only users in all of the selected groups can view this bug:
(Leave all boxes unchecked to make this a public bug)

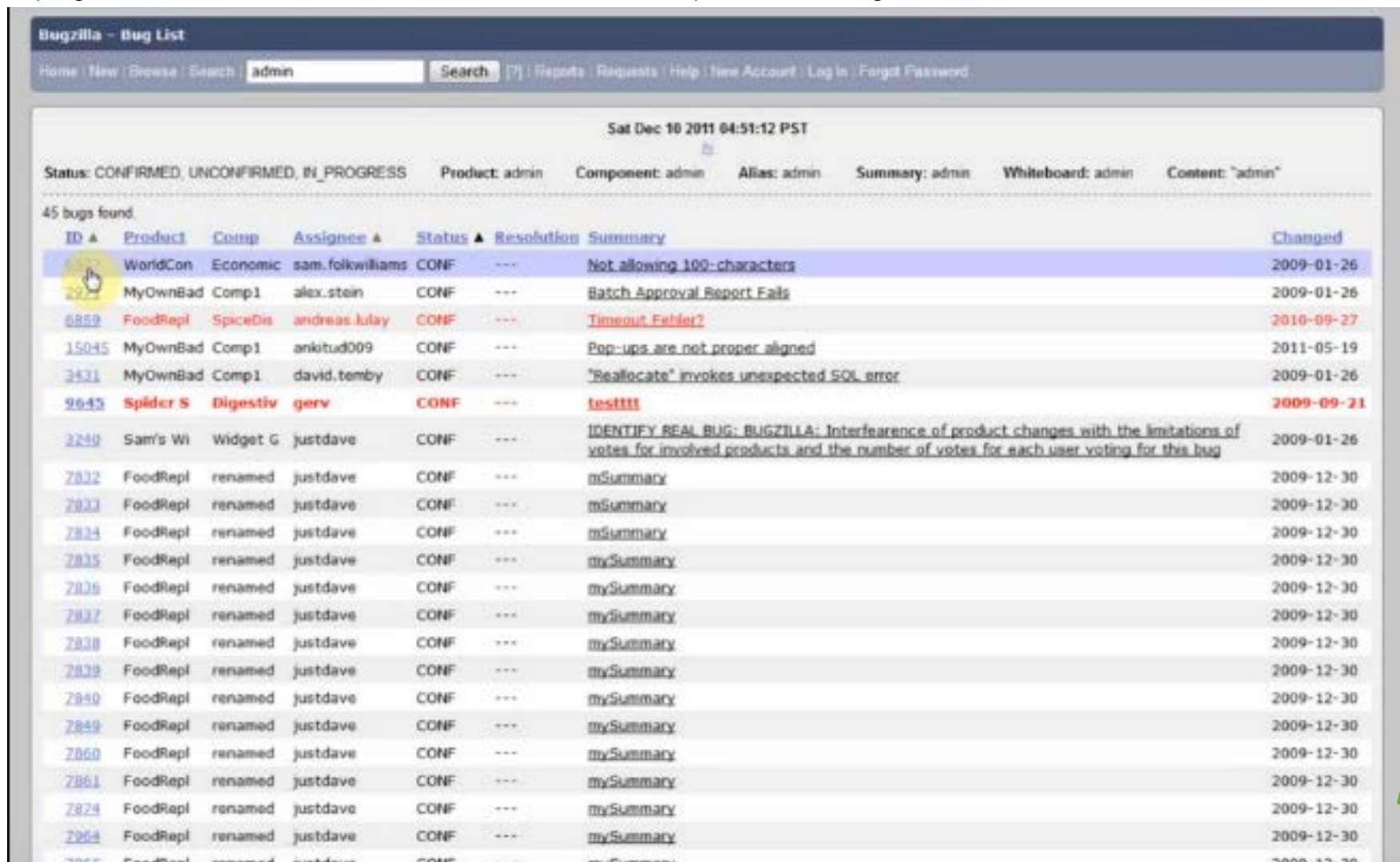
testing group

11:01 / 48:07 [Taskbar Icons]

Figura 5. Imagen ingreso de datos Bugzilla

Fuente: <https://www.youtube.com/watch?v=vBAKYRJgV64> minuto 11

A continuación, se puede apreciar una imagen de la trazabilidad de errores que se encuentran registrados. El programa también notifica al correo electrónico cada vez que se realiza algún cambio.



Bugzilla - Bug List

Home | New | Browse | Search | Search [7] | Reports | Requests | Help | New Account | Log In | Forgot Password

Sat Dec 10 2011 04:51:12 PST

Status: CONFIRMED, UNCONFIRMED, IN_PROGRESS Product: admin Component: admin Alias: admin Summary: admin Whiteboard: admin Content: "admin"

45 bugs found.

ID ▲	Product	Comp	Assignee ▲	Status ▲	Resolution	Summary	Changed
	WorldCon	Economic	sam.folkwilliams	CONF	---	Not allowing 100-characters	2009-01-26
2011	MyOwnBad	Comp1	alex.stein	CONF	---	Batch Approval Report Fails	2009-01-26
6859	FoodRepl	SpiceDis	andreas.kulay	CONF	---	Timeout Fehler?	2010-09-27
15045	MyOwnBad	Comp1	ankitud009	CONF	---	Pop-ups are not proper aligned	2011-05-19
2431	MyOwnBad	Comp1	david.temby	CONF	---	"Reallocate" invokes unexpected SQL error	2009-01-26
9643	Spider S	Digestiv	gerv	CONF	---	testttt	2009-09-21
3249	Sam's Wi	Widget G	justdave	CONF	---	IDENTIFY REAL BUG: BUGZILLA: Interference of product changes with the limitations of votes for involved products and the number of votes for each user voting for this bug	2009-01-26
7832	FoodRepl	renamed	justdave	CONF	---	mySummary	2009-12-30
7833	FoodRepl	renamed	justdave	CONF	---	mySummary	2009-12-30
7834	FoodRepl	renamed	justdave	CONF	---	mySummary	2009-12-30
7835	FoodRepl	renamed	justdave	CONF	---	mySummary	2009-12-30
7836	FoodRepl	renamed	justdave	CONF	---	mySummary	2009-12-30
7837	FoodRepl	renamed	justdave	CONF	---	mySummary	2009-12-30
7838	FoodRepl	renamed	justdave	CONF	---	mySummary	2009-12-30
7839	FoodRepl	renamed	justdave	CONF	---	mySummary	2009-12-30
7840	FoodRepl	renamed	justdave	CONF	---	mySummary	2009-12-30
7849	FoodRepl	renamed	justdave	CONF	---	mySummary	2009-12-30
7860	FoodRepl	renamed	justdave	CONF	---	mySummary	2009-12-30
7861	FoodRepl	renamed	justdave	CONF	---	mySummary	2009-12-30
7874	FoodRepl	renamed	justdave	CONF	---	mySummary	2009-12-30
7864	FoodRepl	renamed	justdave	CONF	---	mySummary	2009-12-30
7865	FoodRepl	renamed	justdave	CONF	---	mySummary	2009-12-30

Figura 6. Imagen ingreso de datos Bugzilla

Fuente: <https://www.youtube.com/watch?v=FUiZQ2UnUTg>, minuto 6:37.

Bugzilla Testopia

Testopia maneja extensiones para la interacción con Bugzilla. Básicamente esta herramienta permite la administración y seguimiento de casos de prueba. (Network, 2005-2017)

Generatedata

Herramienta para la ejecución de pruebas que puede ser actualizada por cualquier persona. Tiene una versión beta gratuita y también una versión completa. Es creada en PHP con javascript y usa

una base de datos MySQL para ayudar a generar datos aleatorios que faciliten las pruebas.

En esta herramienta se pueden configurar los datos de prueba que se requieren, por ejemplo si se requiere tener una lista de nombres, apellidos y edades, entonces se diligencian los campos y se indica el tipo de formato en el que se requiere dentro de los cuales están: html, Excel, xml, csv o sql.

The screenshot shows the Generatedata.com web application interface. At the top, there's a header with the logo and navigation links: 'Generar', 'Acerca de', 'Noticias', and 'Donar'. A language dropdown is set to 'Español'. Below the header, there's a text input field for the data set name, followed by a 'SALVAR' button and icons for CSV, Excel, and SQL. A section for 'DATOS ESPECÍFICOS DE PAÍS' has a dropdown set to 'Todos los países'. The main section is 'CONJUNTO DE DATOS', which contains a table with columns: Orden, Título de columna, Tipo de dato, Ejemplos, Opciones, Ayuda, and Elim. The table has four rows, each with a dropdown for 'Tipo de dato'. Below the table, there's a 'Añadir' button and a 'Fila(s)' input. A section for 'TIPOS DE EXPORTACIÓN' shows tabs for CSV, Excel, HTML (selected), JSON, LDIF, and 'Lenguaje de programación'. Below these are radio buttons for 'Formato de datos' (HTML selected) and a checkbox for 'Usar formato HTML personalizado'. At the bottom, there's a 'Generar' button with a '100' input for the number of rows, and radio buttons for 'Generar en la página', 'Nueva ventana/solapa', and 'Solicitar la descarga'. A 'Zip?' checkbox is also present.

Figura 7. Imagen configuración de datos Generatedata
Fuente: <http://www.generatedata.com/#t1>

A partir de esto, en la herramienta se pueden generar por ejemplo mil datos y pueden ser visualizados y manipulados:

A	B	C
Nombre	Apellido	Num doc
Sonia	Carver	1625052779999
Reese	Zamora	1614071245699
Harrison	Key	1637042603199
Winter	Sweeney	1643041213999
Summer	Henry	1612090381299
Hanae	Wagner	1668023043299
Kenyon	Santana	1661100705299
Quyn	Gonzales	1660110345999
Vielka	Mccormick	1696020975299
Nyssa	Hayden	1687012513399
Kieran	Rivera	1643051154099
Yardley	Schneider	1673090787899
Cole	Mayer	1635062121699
Damon	Weber	1691122793799
Dean	Jackson	1616031543799
Ori	Hoover	1672031167899
Debra	Mcmahon	1693071482599
Gil	King	1600031214499

Figura 8. Imagen reporte de datos de prueba generados- Generedata
Fuente: SENA

Soaptest

Es una herramienta que facilita la ejecución de pruebas de integración, APIs, webservices, entre otros.

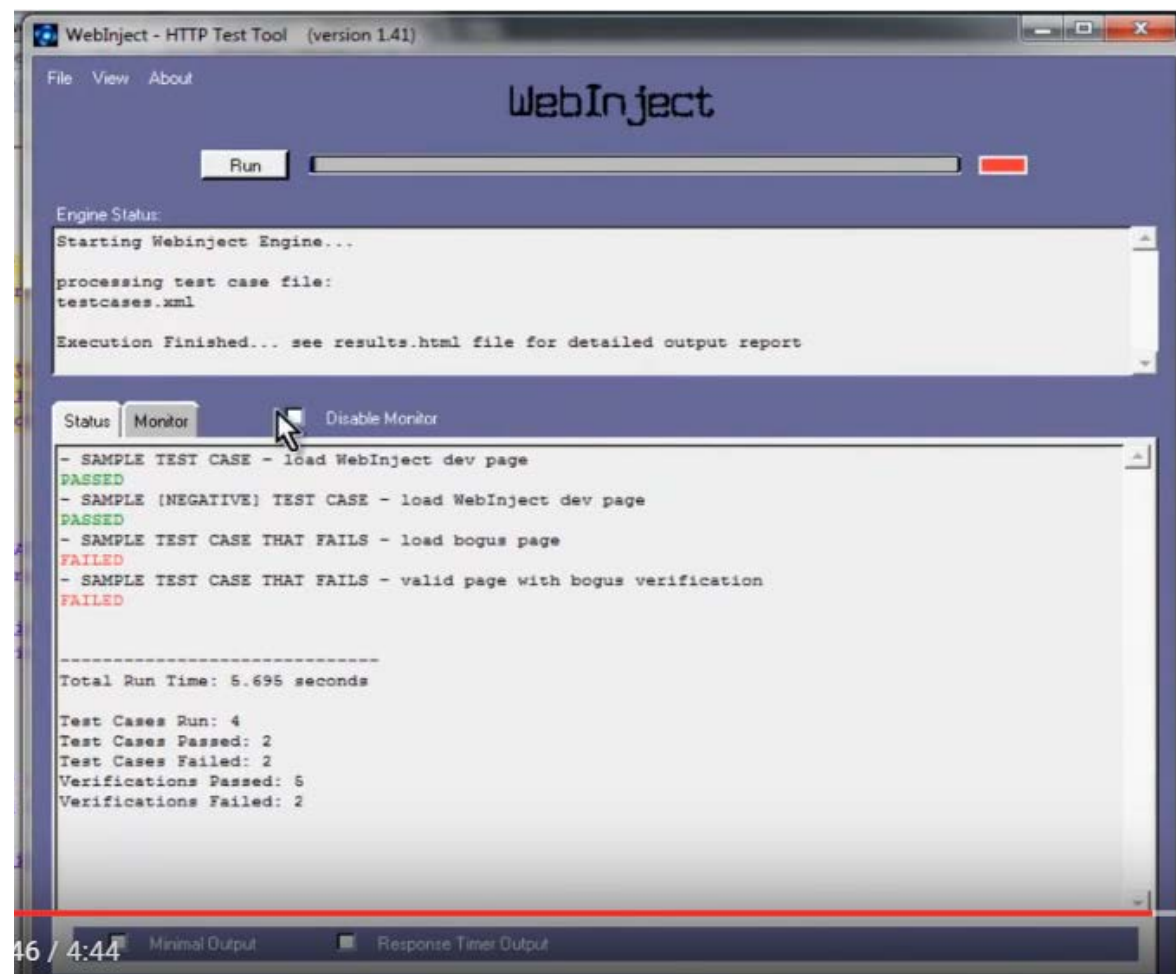
Esta herramienta permite la elaboración de pruebas funcionales, pruebas de desarrollo, pruebas de aplicaciones en la nube, pruebas de aplicaciones complejas.



Figura 9. Imagen Soaptest
Fuente: <https://www.parasoft.com/products/>

WebInject

Es una herramienta gratuita que facilita la automatización de pruebas funcionales, aplicaciones web y webservices.



Contiene un API en xml que permite evaluar los casos de prueba. Está escrita en Perl y puede ser instalada en Windows, Linux, Mac.

Figura 10. Imagen WebInject

Fuente:

<https://www.youtube.com/watch?v=FQsXzQAIEqQ>, minuto 3:44

WebLoad

Es una herramienta que utiliza javascript para generar escenarios de prueba. De esta manera permite simular el acceso de múltiples usuarios, lo cual permite evaluar el rendimiento del software.

De igual manera permite generar reportes para analizar el comportamiento de base de datos, servidores, sitio web y otros componentes utilizados durante las pruebas.

Permite realizar acciones de:

- Correlación
- Verificación
- Parametrización
- Vista de consola

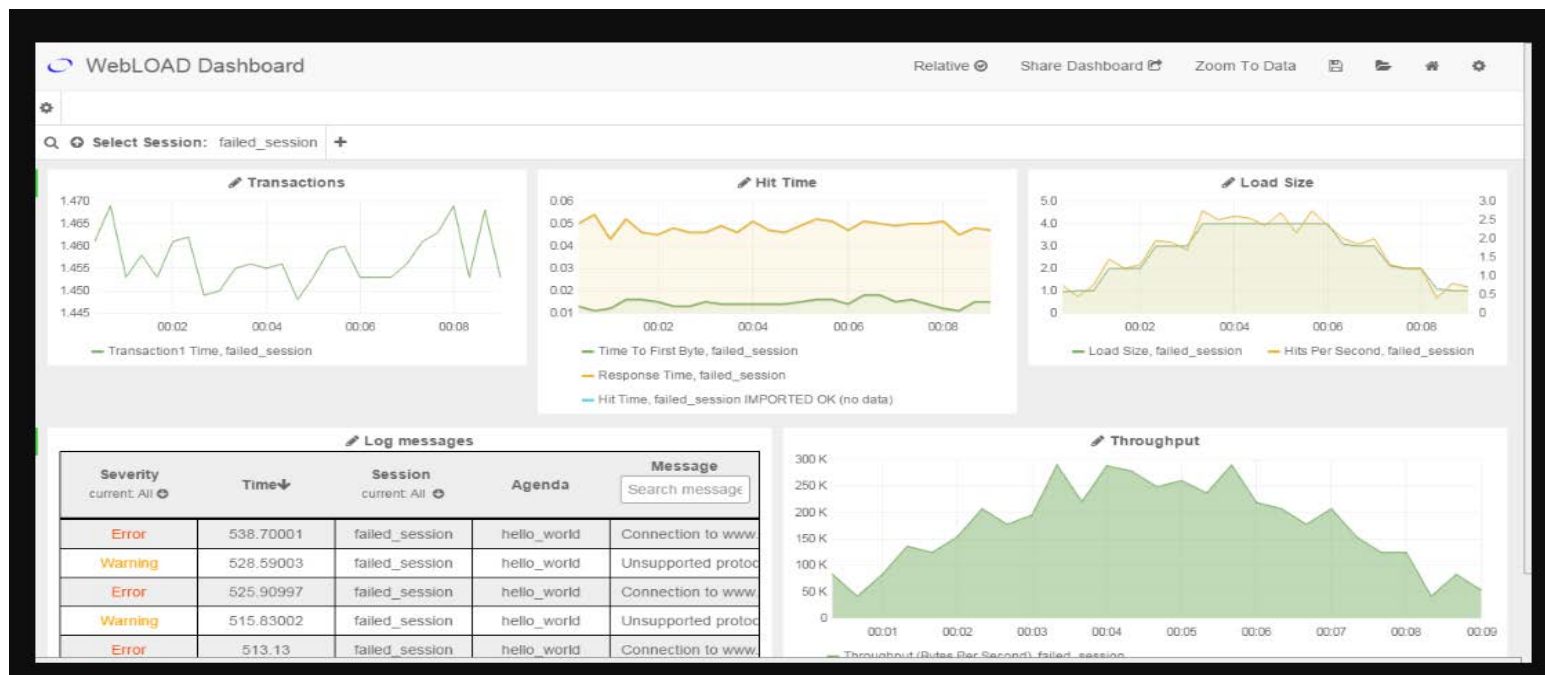


Figura 11. Imagen Webload

Fuente: <http://www.radview.com/solution/web-testing-tools/>

Jmeter

Jmeter es una herramienta que permite realizar pruebas de carga y rendimiento en el software, simulando una cantidad de usuarios específica que se puede configurar de acuerdo con las necesidades de las pruebas.

Se configura con el número de usuarios que van a realizar las pruebas, hora de inicio, hora de fin, se puede añadir gráfico de resultados.

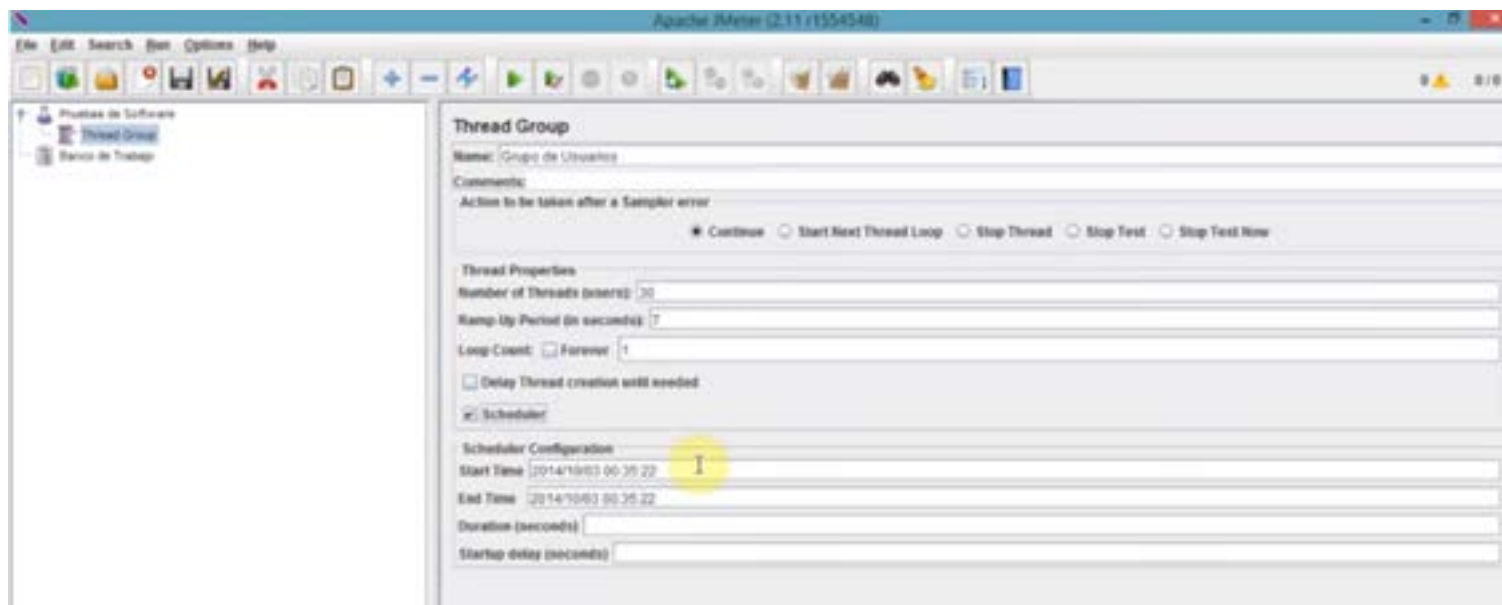


Figura 12. Imagen Jmeter

Fuente: <https://www.youtube.com/watch?v=1yJUyRWWOsg>

QA complete

Es una herramienta que facilita la gestión de pruebas; pues permite organizar los casos de prueba, ambientes de prueba, pruebas automatizadas, defectos y tareas de pruebas en el proyecto.



Figura 13. Imagen QAcomplete

Fuente: <http://www.testmanagement.com/qacomplete.html>

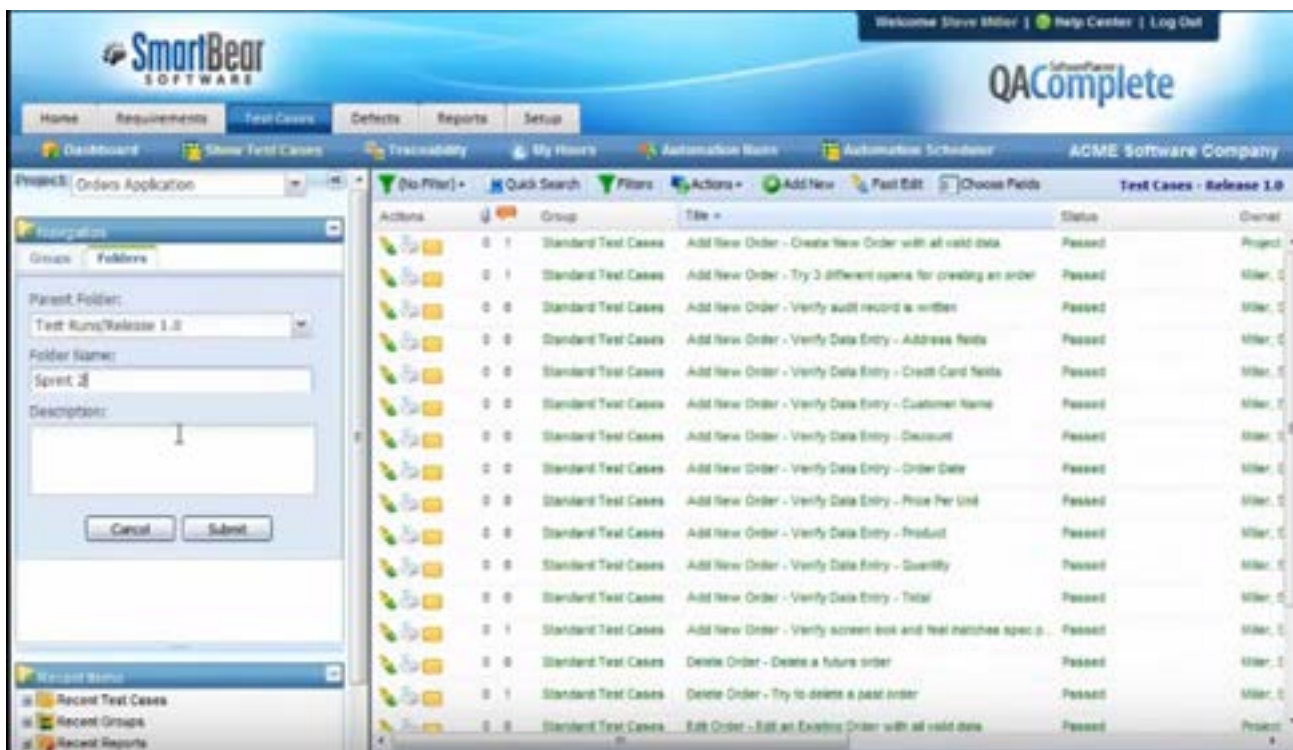


Figura 14. Imagen QAcomplete

Fuente: <http://www.testmanagement.com/qacomplete.html>

Testitool

Es una herramienta gratuita para la gestión de prueba, así permite generar casos de prueba, asignarlos, ponerles un estado, generar informes y generar instancias del plan de pruebas. Esta herramienta está hecha en PHP.

qaBook

Esta herramienta tiene una versión gratuita para la gestión de pruebas. También tiene otras versiones que contemplan acciones específicas de acuerdo con lo que se requiera:



Figura 15. Imagen QA BOOK
Fuente: <http://www.qabook.com/>

Wet

Es una herramienta gratuita para realizar pruebas funcionales que permite realizar todas las operaciones necesarias para probar aplicaciones web, tales como prueba de acceso en enlaces, ingreso de información en campos de texto, clic en los botones, entre otros.

Test Studio

Hay pocas herramientas que permiten realizar todo tipo de pruebas y que también permite la gestión de las mismas. Este es el caso de Test Studio. Es una herramienta que permite de manera completa la automatización de pruebas de interfaz de usuario, componentes, pruebas de integración, pruebas web, pruebas móviles y para aplicaciones de escritorio.

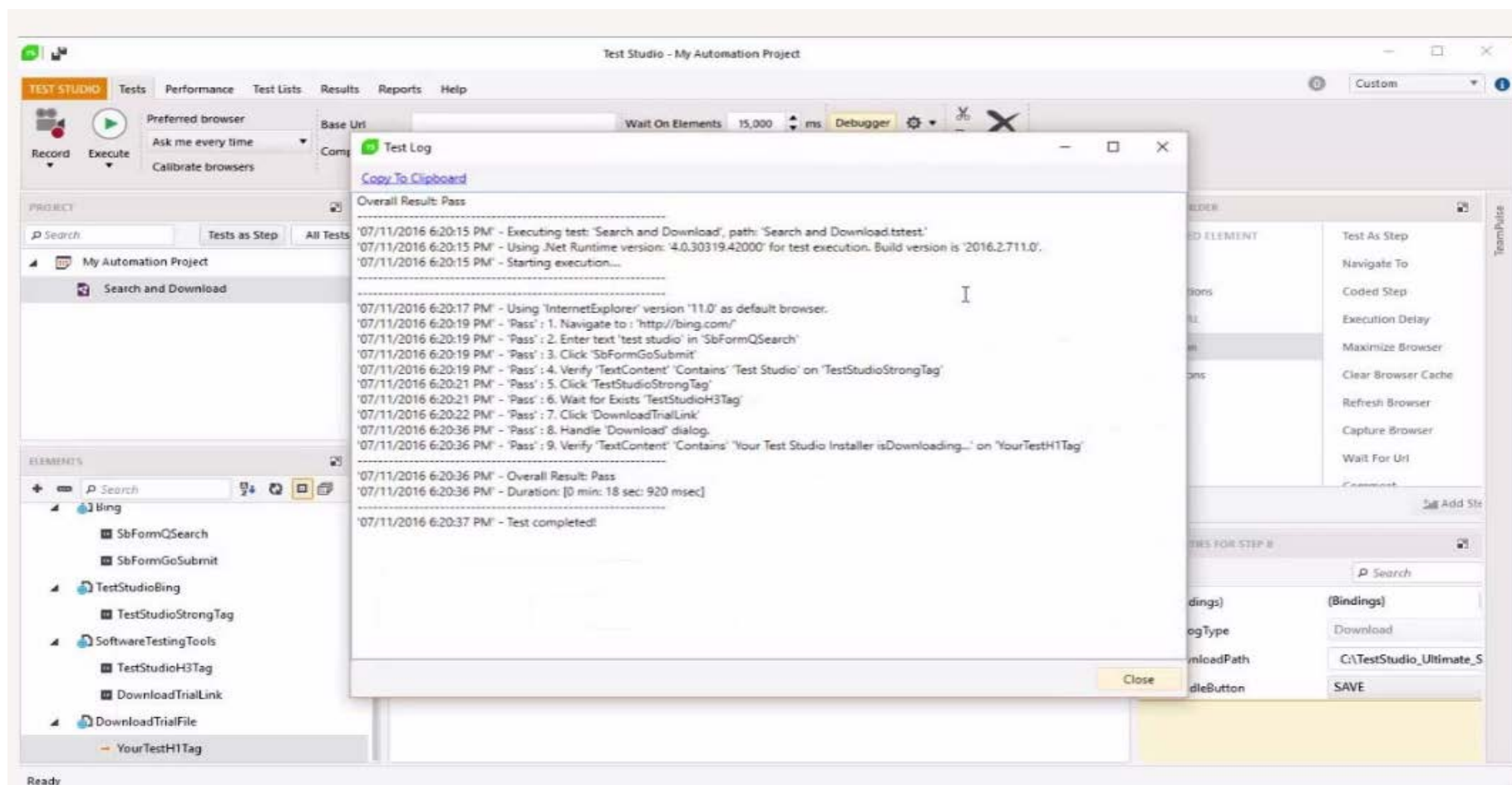



Figura 16. Imagen Test Studio
Fuente: <http://www.telerik.com/teststudio>



5. Material de apoyo

Para complementar los diferentes conceptos adquiridos en el presente documento, se recomienda a los aprendices consultar el documento anexo Roles para la gestión de pruebas.





6. Glosario

Herramientas de soporte de pruebas: software que facilita la ejecución de pruebas ya que permite automatizar algunas de las actividades a realizar.

Tipos de prueba: es la clasificación de las pruebas de software de acuerdo con los objetivos de las mismas y la metodología a utilizar.



7. Referencias bibliográficas

Marqu ez, J. (2015). ***Las mejores herramientas para realizar pruebas de software.***
Recuperado el 03 de junio de 2017 <https://es.slideshare.net/nanotechnology/las-mejores-herramientas-para-realizar-pruebas-de-software>



Créditos

Equipo de Adecuación Gráfica
Centro de Comercio y servicios
SENA Regional Tolima

Línea de Producción

Director Regional

Félix Ramón Triana Gaitán

Subdirector de Centro

Álvaro Fredy Bermúdez Salazar

Coordinadora de Formación Profesional

Gloria Ines Urueña Montes

Senior Equipo de Adecuación

Claudia Rocio Varón Buitrago

Experta Temática

Catalina Roperero Acero

Asesora Pedagógica

Ángela Patricia Frasser Castaño

Guionistas

Genny Carolina Mora Rojas

Jesús Bernardo Novoa Ortiz

Diseño y Diagramación

Diana Katherine Osorio Useche

Pedro Nel Cabrera Vanegas

Ismael Enrique Cocomá Aldana

Programadores

Davison Gaitán Escobar

Héctor Horacio Morales García



Creative commons

Atribución, no comercial, compartir igual.

Este material puede ser distribuido, copiado y exhibido por terceros si se muestra en los créditos. No se puede obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo comercial.

cc creative commons

