

Prueba Automática de Carga y Estrés

Automatic test of Load and Stress

Liudmila Sanchez Almenares

Universidad de las Ciencias Informáticas.

lalmenares@uci.cu

Resumen

Este trabajo propone un manual para realizar pruebas de cargas y estrés de manera automatizada por medio de la herramienta JMeter. Se describen brevemente aspectos vinculados con este tema como son: la calidad del software, tipos de pruebas, niveles de prueba y características de valoración de los resultados. Además se aborda una breve reseña de los antecedentes vinculados con el proyecto “Cuerpo Investigación Científicas Penales y Criminalísticas” en lo adelante CICPC, donde quedan reflejados objetivos, actividades, artefactos y otros trabajos realizados con anterioridad que están vinculados con esta investigación. De igual manera se expone una panorámica relacionada con la clasificación de las herramientas de automatización de las pruebas, y herramientas de prueba reconocidas mundialmente. Por lo demás se realiza una comparación entre estas herramientas y se justifica JMeter como herramienta para la automatización de pruebas de carga y estrés en el proyecto CICPC. Realizar las pruebas automatizadas implica introducir nuevas tareas en el proceso de pruebas de carga y estrés por tanto se realiza una descripción del proceso identificando roles, actividades y artefactos. Por último se aplica el proceso descrito a las pruebas de carga y estrés en las aplicaciones producidas por el equipo de desarrollo del CICPC, quedando estas como ejemplo de la utilización de este nuevo proceso y como validación del mismo.

Palabras clave: Calidad del Software, herramienta de prueba, prueba automática, prueba de carga y estrés, prueba del software.

Abstract

This document proposes a manual to perform load and stress testing automatically by means of using tool JMeter. Several aspects linked to this subject are briefly described, such as software quality, kinds of tests, testing levels, and results evaluation characteristics. Furthermore, a brief historical account of the precedents linked with project CICPC will be exposed, where objectives, activities, artefacts and other pieces of work linked to this investigation. Likewise, a wide view of the tests automation tools classification process will be exposed, as well as testing tools used worldwide. In addition, a comparison between these tools is made, and the use of JMeter as the tool for the load and stress testing automation in project CICPC is augmented. Performing automated tests implies to import new tasks in the load and stress testing process; therefore, a description of the process is made, identifying roles, activities and artefacts. Last but not least, the described process is applied to the load and stress tests in the applications produced by the CICPC's development team; leaving them as usage examples for this new process and it's validation.

Key words: Automated test, load and stress testing, testing tool, software quality, software testing.

Introducción

El objetivo de la fase de pruebas es detectar las anomalías que posee el software antes de su entrega al cliente, esta puede definirse como un elemento crítico para la garantía de la calidad del software. Conociendo la calidad del software como: “Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente

documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente” (Pressman, 1992).

El proyecto de modernización del Cuerpo de Investigaciones Científicas Penales y Criminalísticas, en lo adelante proyecto CICPC, es un proyecto de la Universidad de las Ciencias Informáticas que ha surgido con el fin de dar respuestas a dos solicitudes realizadas por el gobierno Bolivariano de Venezuela: Una nueva versión del Sistema de Investigación e Información Policial (SIIPOL) y el Portal Web de la organización. Las aplicaciones desarrolladas por este proyecto en la Universidad precisan de una rigurosa etapa de pruebas de carga y estrés por sus características. Además estas pruebas necesitan reflejar: la cualidad de la aplicación de reponerse ante indisponibilidad en alguno de sus servicios o el fallo de un ente externo ya sea eléctrico, problemas de red, hasta la velocidad de envío de paquetes de información entre el cliente y el servidor.

En la actualidad en la Universidad de las Ciencias Informáticas las pruebas carga y estrés se realizan de manera manual y la acción de probar el software requiere del empleo de gran cantidad de tiempo y recursos, por ello se ve afectada la calidad de los resultados de las mismas.

Si en las pruebas que se necesitan realizar en el Proyecto CICPC se utilizan los métodos que hasta el momento se conocen en la universidad, el resultado puede no ser efectivo ya que implicarían miles de personas para poder simular un escenario real y el empleo más recursos y tiempo. Otro problema relacionado con ello sería la precisión en cuanto a los resultados ya que el método manual puede traer consigo errores derivados del factor humano, además del poco tiempo con el que se cuenta para realizar este proceso.

Por tanto el objetivo general que se persiguió para el desarrollo del trabajo fue: Investigar y proponer herramientas para ejecutar pruebas de carga y estrés en el Proyecto CICPC, con ello se defiende la idea de que la utilización de una herramienta para la realización de las pruebas de carga y estrés a las aplicaciones del proyecto CICPC se realizan a través de una herramienta incurriría en una disminución de tiempo y recursos para este proceso y aumentaría la calidad de los resultados de estas pruebas.

Desarrollo

Para realizar una selección de las herramientas necesarias se realizó un estudio de las herramientas vinculadas al proceso de pruebas para detectar cuál de estos tipos de herramientas es necesario aplicar al proceso. Luego se analizó el proceso de pruebas a realizar en el Proyecto con las características específicas de sus aplicaciones. Se seleccionó una herramienta y se adaptó la utilización de esta a la definición del proceso de pruebas surgiendo como un nuevo proceso de este tipo en la Universidad.

1. Herramientas de automatización de las pruebas

1.1. Conceptos

Según la metodología RUP las herramientas de prueba se pueden categorizar según las funciones que realicen. Algunas designaciones de funciones típicas para herramientas son:

- **Herramientas de adquisición de datos** que adquieren datos para utilizar en las tareas de prueba. Los datos se pueden adquirir mediante la conversión, la extracción, la transformación o la captura de datos existentes, o mediante la generación de guiones de uso o especificaciones suplementarias.
- **Herramientas estáticas de medida** que analizan información contenida en los modelos de diseño, el código fuente u otros orígenes fijos. El análisis produce información en el flujo lógico, el flujo de datos o la métrica de calidad, como la complejidad, el mantenimiento o las líneas de código.
- **Herramientas dinámicas de medida** que realizan un análisis durante la ejecución del código. Las medidas incluyen la operación de tiempo de ejecución del código, como la memoria, la detección de errores y el rendimiento.

- **Simuladores o controladores** que realizan tareas que, por cuestiones de tiempo, gastos o seguridad no están disponibles para las pruebas.
- **Herramientas de gestión de pruebas** que ayudan en la planificación, el diseño, la implementación, la ejecución, la evaluación y la gestión de tareas de prueba o productos de trabajo.

Además las herramientas de prueba suelen caracterizarse como cajas blancas o cajas negras en función de cómo se utilicen, o la tecnología y los conocimientos necesarios para utilizarlas.

- **Las herramientas de caja blanca** dependen del conocimiento del código, los modelos de diseño y otro material de origen para implementar y ejecutar las pruebas.
- **Las herramientas de caja negra** dependen de los guiones de uso o la descripción funcional del destino de la prueba.

Las herramientas de caja blanca saben cómo procesa la solicitud el destino de la prueba, mientras que las herramientas de caja negra dependen de las condiciones de entrada y de salida para evaluar la prueba.

Además de las amplias clasificaciones de herramientas que se presentaron antes, las herramientas también se pueden clasificar según la especialización.

- **Las herramientas de grabación y reproducción** combinan la adquisición de datos con la medida dinámica. Los datos de prueba se adquieren durante la grabación de sucesos (conocida como implementación de la prueba). Más tarde, durante la ejecución de la prueba, los datos se utilizan para reproducir el script de prueba, que se utiliza para evaluar la ejecución del destino de la prueba.
- **Las herramientas de métrica de calidad** son herramientas de medida estática que realizan un análisis estático de los modelos de diseño o código fuente para establecer un conjunto de parámetros que describen la calidad del destino de la prueba. Los parámetros pueden indicar fiabilidad, complejidad, mantenimiento u otras medidas de calidad.
- **Las herramientas de supervisión** de la cobertura indican la completitud de la prueba mediante la identificación de la cantidad de destino de la prueba cubierta, en alguna dimensión, durante la prueba. Las clases típicas de cobertura son guiones de uso (basados en requisitos), nodo o ramificación lógica (basada en código), estado de los datos y puntos de función.
- **Los generadores de guiones de prueba** automatizan la generación de los datos de prueba. Los generadores de guiones de prueba utilizan o bien una especificación formal de las entradas de datos del destino de la prueba o bien los modelos de diseño y el código fuente para producir datos de prueba que prueben las entradas nominales, las entradas de errores y los guiones de límite.
- **Las herramientas del comparador** comparan los resultados de la prueba con los resultados de referencia e identifican las diferencias. Los comparadores se diferencian en su especificación para formatos de datos particulares. Por ejemplo, pueden basarse en píxeles para comparar imágenes de mapa de bits o en objetos para comparar las propiedades o los datos del objeto.
- **Los extractores de datos** proporcionan entradas para los guiones de prueba de orígenes existentes, incluidos bases de datos, secuencias de datos de un sistema de comunicación, informes o modelos de diseño y código fuente.

1.2. Resumen de Herramientas para Pruebas de Carga y Estrés

En vistas a encontrar una herramienta que fuera capaz de dar respuesta a esta situación se realizó una investigación encontrando cuatro herramientas. Estas herramientas cumplen con las características necesarias para ejecutar las pruebas de carga y estrés o apoyar el proceso de pruebas. Una breve descripción de ellas se muestra a continuación.

QALoad

QALoad de la empresa Compuware es una herramienta de automatización de pruebas de carga para web, Java, .NET, aplicaciones ERP (Planificación de recursos empresariales) y CRM (gestión de relaciones con los clientes y ambientes distribuidos). Simula miles de usuarios desempeñando transacciones de negocio clave de una aplicación para asegurar su desempeño y escalabilidad

previa a su puesta en producción. Con esta herramienta, los equipos de prueba pueden rápidamente detectar problemas, optimizar el desempeño de los sistemas y ayudar a asegurar un despliegue de aplicaciones exitoso. (10)

LoadRunner

LoadRunner es una herramienta para realizar pruebas de carga de Mercury Interactive que permite pre-ver el comportamiento y el rendimiento del sistema. Además, permite poner a prueba toda la infraestructura corporativa para identificar y aislar los posibles problemas mediante la simulación de la actividad de miles de usuarios. Realiza pruebas en toda la infraestructura corporativa, que comprende las soluciones e-business, ERP, CRM y las aplicaciones personalizadas, simulando la actividad de miles de usuarios, con lo que los equipos de desarrollo de aplicaciones y sitios Web pueden mejorar el rendimiento de las aplicaciones. Es la herramienta de pruebas de carga más escalable que permite simular la actividad de miles de usuarios con los mínimos recursos de hardware. Se integra a la perfección con las herramientas de gestión del rendimiento de Mercury Interactive. Los mismos scripts creados durante las pruebas pueden volverse a utilizar para monitorizar la aplicación una vez terminada su implantación. Presenta una interfaz API abierta, con la que los usuarios y otros fabricantes pueden integrar LoadRunner en sus propios entornos (11).

Quality Center

Permite automatizar los procesos de calidad, uniendo todos los componentes con las aplicaciones correctas para acelerar los tiempos de depuración. El resultado es una mejora impresionante en la calidad y en la consistencia de la aplicación. Ayuda a manejar y controlar el riesgo, mientras se desarrolla y prueba la aplicación. En todos los momentos del proceso, se tiene visibilidad de en dónde se encuentra el proyecto con respecto a calidad (requerimientos probados y satisfechos, pruebas ejecutadas, defectos y tendencias encontradas, etc.). Maneja y automatiza el proceso de entrega, con indicadores claves de rendimiento, a tiempo real, proporciona aplicaciones que automatizan todas las actividades claves en los procesos de calidad, y apoya a todas las personas y roles que necesitan involucrarse en los procesos de entrega. (12)

Esta herramienta no es una herramienta de pruebas de sistemas pero se ha decidido involucrarla en la investigación ya que la utilización de la misma en el manejo de la calidad del proceso de pruebas influye en la garantía de los resultados y del proceso en general, permitiendo una mayor organización del proceso e influye directamente en el rendimiento del equipo de trabajo.

JMeter

Es una herramienta Java desarrollada dentro del proyecto Jakarta, que permite realizar Pruebas de Rendimiento y Pruebas Funcionales sobre Aplicaciones Web.[15] Permite realizar pruebas Web clásicas, pero también permite realizar test de FTP, JDBC, JNDI, LDAP, SOAP/XML-RPC, y WebServices (en Beta). Permite la ejecución de pruebas distribuidas entre distintos ordenadores, para realizar pruebas de rendimiento. Además activar o desactivar una parte del test, lo que es muy útil cuando se está desarrollando un test largo, y se desea deshabilitar ciertas partes iniciales que sean muy pesadas o largas. Tiene la forma de generar un caso de prueba a través de una navegación de usuario. (15)

Para la elección de la herramienta candidata se analizaron los siguientes aspectos:

- Funcionalidades: Prestaciones de la herramienta, elementos funcionales, respuesta a los requerimientos del sistema.
- Modo de adquisición: Si el software se obtiene por medio de una licencia de pago o es gratuito.
- Materiales de apoyo: Si existe documentación de la herramienta que ofrezca una guía para su utilización.
- Entorno en el que fue desarrollado: Si es un software a código abierto desarrollado en el ambiente de software libre o si es un software del cual no se puede obtener información de su código o no se puede modificar y adecuar a los ambientes de trabajo del proyecto por ser un software propietario.

Se decidió utilizar para las pruebas Carga y Estrés la herramienta JMeter por las funcionalidades que presenta, por utilizarse en proyectos con características similares a los desarrollados por el Proyecto CICPC realizados en java, ser un software libre y gratis. Además se pueden utilizar muchas de sus funcionalidades para probar aspectos de las pruebas de Disponibilidad y Red.

Atendiendo a la clasificación expuesta anteriormente la herramienta JMeter es una se clasifica como dinámica de medida por su tipo de función, como manipula datos de entrada para la ejecución de las pruebas y necesita la comparación de los resultados para la evaluación, recurre al método de caja negra. Además se clasifica por su especialización en una herramienta de grabación y reproducción.

La utilización del JMeter supone un 95 % de tiempo menos para la realización de estas pruebas. La mayor inversión de tiempo que se necesita para la realización de las pruebas es la fase de estudio de los casos de uso críticos en la aplicación Web y la elaboración del plan de pruebas en la herramienta. Permite almacenar los resultados de la prueba y se generar gráficos que representan los aspectos que se han probado.

La posibilidad de contar con material para el estudio de la utilización de la herramienta, implica una garantía en el éxito de la ejecución de las pruebas y de la realización de un manual que permita a otros proyectos de gestión, identificar puntos de control para la elaboración de estas pruebas y la ejecución de las mismas además de la recopilación y análisis de los errores encontrados.

Otro aspecto a favor de esta herramienta es su desarrollo en el marco del software libre, política que apoya la universidad y que por las características existe un por ciento superior de confianza en la utilización de la misma ya que se puede contar con todo el código que genera a la herramienta.

Como desventaja de la herramienta se encuentra el alto consumo de rendimiento del CPU ya que es necesario para su mejor desempeño más de 512 mb de RAM.

2. Proceso de Pruebas con JMeter

Este epígrafe expone una propuesta de un manual para realizar las pruebas de carga y estrés de manera automatizada en la herramienta JMeter. Se definen las actividades a seguir por los trabajadores implicados en esta tarea, así como los requerimientos necesarios para la implementación de las mismas.

2.1. Proceso Automatizado de Carga y Estrés

Para las pruebas de carga y estrés automatizadas por JMeter define 5 importantes etapas en las que se enmarca esta descripción.

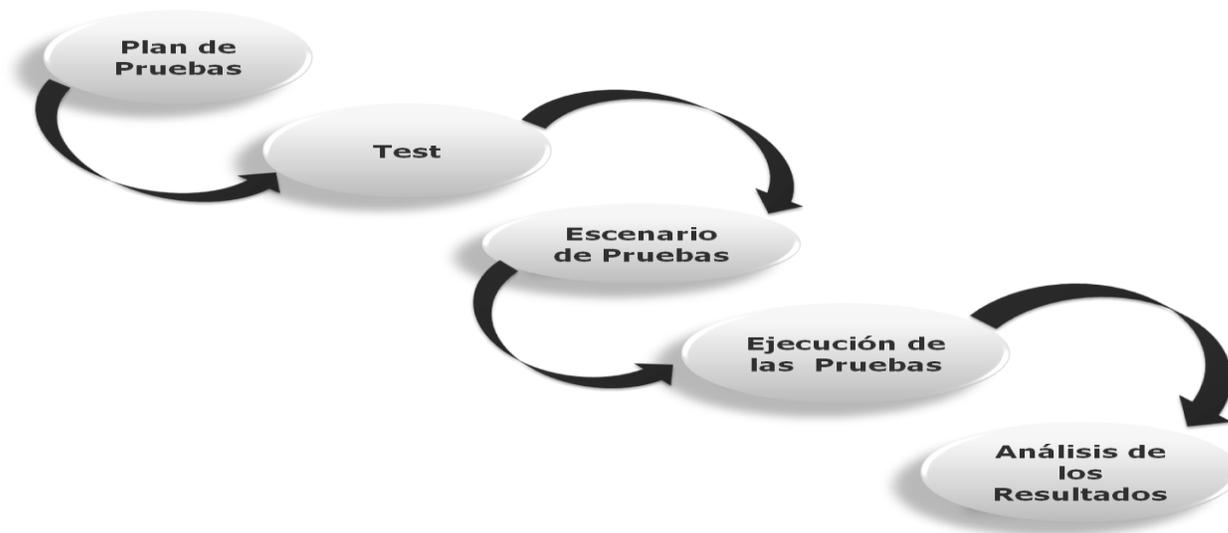


Fig. 1 Proceso de automatización de las pruebas de sistema No Funcionales de Carga y Estrés.

Estas cinco etapas están definidas en diferentes procesos y actividades que se encuentran distribuidos entre trabajadores del equipo de calidad, específicamente del equipo de pruebas. No obstante se pueden identificar otros trabajadores del proyecto que intervienen en esta etapa tales como el Administrador de Sistema y los Jefes de Modulo que representan a su vez un equipo de desarrolladores. Una descripción general del nuevo proceso se puede observar en la figura siguiente.

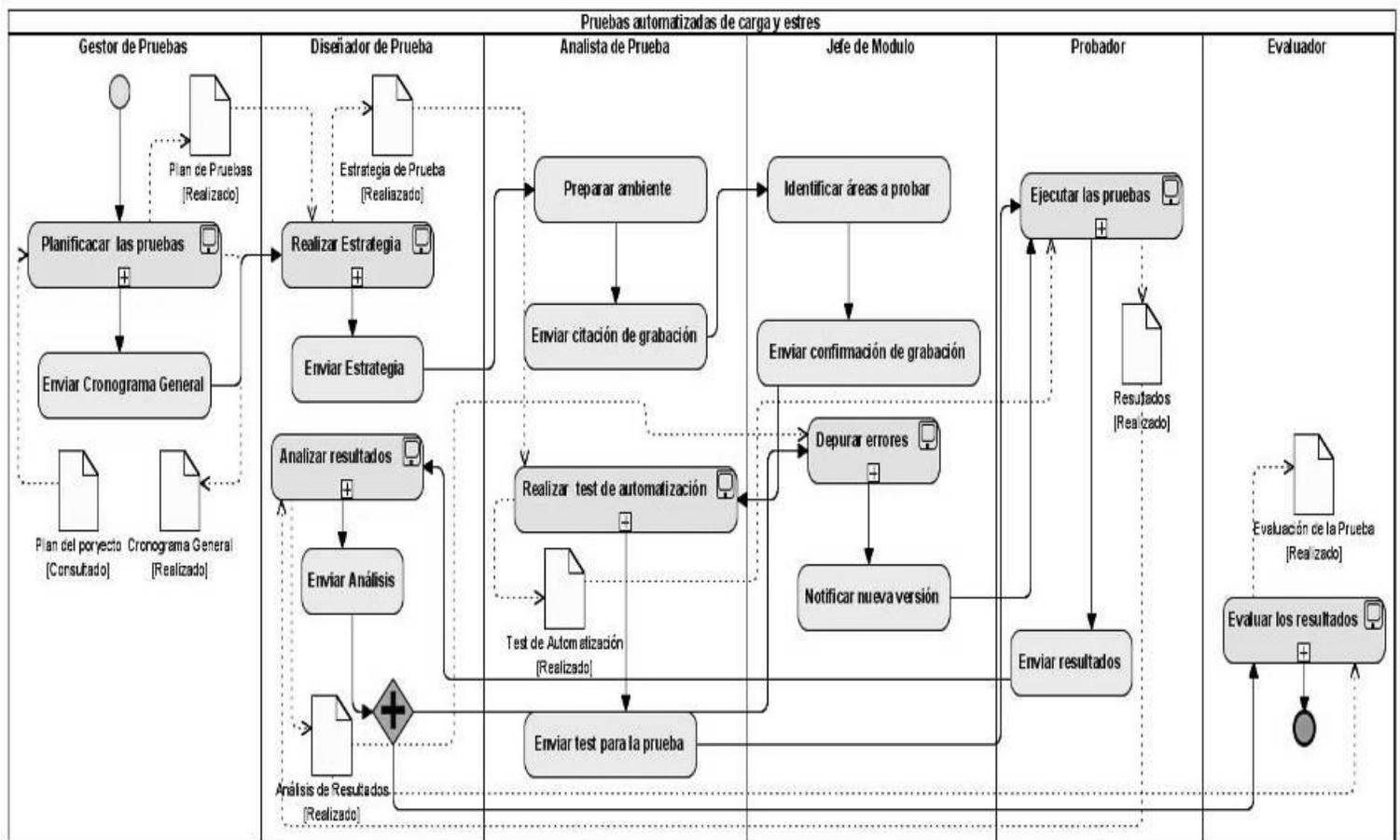


Fig. 2 Proceso de Automatización de las Pruebas de Sistema no Funcionales de Carga y Estrés.

El proceso de pruebas automatizadas de carga y estrés se inicia cuando se concibe el plan de pruebas. El gestor de pruebas envía al diseñador de pruebas una notificación donde se especifica el tiempo en el que encontrará enmarcado el proceso de automatización. El diseñador recibe el plan de pruebas y realiza la estrategia de pruebas y la envía al analista de pruebas el cual comienza a preparar el ambiente donde se realizarán los test y las ejecuciones de las pruebas mientras el jefe de módulo analiza los escenarios a probar. Posterior a ello se procede a la grabación de los escenarios y se agregan elementos de prueba de acuerdo a las especificaciones de la estrategia, eliminando elementos que se grabaron durante la navegación pero no serán utilizados. Una vez probado que el test funciona de la forma correcta se notifica a los probadores quienes comprueban que el entorno donde se ejecutarán las pruebas tenga las características señaladas para este tipo de pruebas y si existe algún problema notifica al analista de pruebas y suspende las pruebas. Si por el contrario se encuentran creadas todas las condiciones se ejecutan los test y se recolectan los resultados que muestra la herramienta. Estos resultados se envían al diseñador de pruebas que analiza los mismos comparándolo con los resultados esperados y envía un informe de análisis al jefe de módulo y al evaluador. El jefe de módulo depura los errores o deficiencias y notifica la creación de una nueva versión del software, mientras el evaluador valora los resultados de la prueba y extiende una evaluación del software. El proceso que se muestra a continuación expone las actividades, artefactos y el rol que componen el subproceso que realiza la estrategia de pruebas.

El proceso incluye otras actividades como pueden ser:

1. Explicar el proceso a los Jefes de Desarrollo: Esta actividad es fundamental ya que los desarrolladores son los mejores documentados sobre el sistema y son ideales para identificar los escenarios de concurrencia o de carga que podrían interesar para la grabación de las pruebas.

2. Explicar características de los escenarios a grabar: Los jefes de Desarrollo no siempre tienen conocimiento sobre las características de los escenarios de concurrencia o de carga, es responsabilidad del equipo de pruebas apoyar a los desarrolladores para que estos puedan identificarlos.

Aportes del trabajo:

Este trabajo tiene como aportes científicos la elaboración de un manual que ejemplifica los pasos a seguir en la herramienta JMeter para realizar pruebas de Carga y Estrés. Además se define un nuevo proceso para las pruebas de cargas el cual identifica nuevas actividades y artefactos que interactúan en el mismo. Y en su momento posibilitó las pruebas de Carga y Estrés en las aplicaciones realizadas por el Proyecto CICPC, mostrando sus resultados como aval del buen desempeño en este equipo de trabajo e identificando algunos riesgos que de persistir podrían incurrir en sobre carga del sistema o lentitud de sus prestaciones.

Conclusiones

Este trabajo ha tratado de reflejar la importancia que tienen las pruebas de carga y estrés de manera automatizada en aplicaciones webs, demostrando que la aplicación de una herramienta en este proceso, la definición de las misiones y tareas de manera planificada y organizada, que se necesitan realizar estas pruebas ha favorecido al equipo de producción del Proyecto CICPC ya que la inversión total de tiempo fue muy pequeña en comparación a la inversión que se necesitaría si se hubiera utilizado personal para simular esa cantidad de usuarios que era necesario para cada uno de las aplicaciones, además que el sistema de medición que ofrece la herramienta JMeter es mucho más exacta y extensa que los resultados que tradicionalmente se recogían de manera manual cuando estas pruebas se realizaban, aunque en una mayoría de aplicaciones estas pruebas no se llevaban a cabo con el objetivo de medir el tiempo, si no solo para saber la cantidad de usuarios que soportaba la aplicación de manera concurrente.

Con respecto a los resultados de estas pruebas en los productos del Proyecto CICPC se considera que han sido de gran importancia ya que gracias a estas pruebas se detectaron detalles que pueden tomarse en cuenta en futuras versiones para lograr una mejoría de la gestión de las peticiones.

Referencias Bibliográficas

1. -D'Onofrio, A.-p.D.L., "Probando software y números de versión". 2002
2. -Torres, I.H.M.C., Plan de Pruebas del proyecto CICPC. 2006.
3. - López, N.M.S., Proyecto Técnico 2007.
4. - (IBM), R., Rational Unified Process. 2006.
5. - Torres, I.Y.d.C.T.P.I.Y.M., Propuesta de manual de procedimiento de Pruebas de Sistema y su aplicación en el Proyecto CICPC. 2007, Universidad de las Ciencias Informáticas.
6. -Pressman, R.S., "Ingeniería del software. Un enfoque práctico". 2ª ed. 1992: McGrawHill
7. -Pressman, R.S., Ingeniería del Software. Un enfoque práctico. Quinta ed. 2002: s.l Mc Graw Hill.
8. -W, A.S., Método de Pruebas Orientada a Objetos para el Ciclo de Vida Completo (FLOOT). 2002.
9. -Tamayo, I.M.F.L.D., Requerimientos Suplementarios, proyecto CICPC. Version 1.0 ed. 2007.
10. -Compuware, QLoad. 2007.
11. -Compuware, Compuware. 2006.
12. -Mercuryinteractive.Com, LoadRunner.
13. -Noticias.Com, www.noticias.com/noticia. Febrero 2007.
14. -BamSol, www.bamsol.com/bamsol. 2007.
15. -Jakarta, jakarta.apache.org/jmeter/.

