

**Universidad de las Ciencias Informáticas**



# **Prueba Automática de Carga y Estrés en el Proyecto CICPC**

**Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas**

**Autor: Liudmila Sánchez Almenares**

**Tutor: Msc. Haydee María Cruz Torres**

**“Ciudad de la Habana” Junio 2008**

***“Año del 50 Aniversario de la Revolución Cubana”***

## Declaración de Autoría

Declaro ser autor del presente Trabajo de diploma y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de Junio del año 2008.

---

**Firma del Autor**

Liudmila Sánchez Almenares

---

**Firma del Tutor**

Msc. Haydee María Cruz Torres

*La verdadera medida de nuestra valía se compone de todos los beneficios que los demás han obtenido de nuestro éxito”.*

*Cullen Hightower*

*A mis abuelitos que me han enseñado lo suficiente para crecer, madurar y  
valerme por mí misma.*

*A mis tres padres que han sacrificado por ver hechos realidad mis sueños.*

*A mis hermanos y mi familia en general por ser tan especiales y llenar mi  
vida de buenos pensamientos.*

*A mi novio Adrian Santos Rodríguez, sin ti bebé todo esfuerzo habría sido  
imposible, no podemos andar por el mundo careciendo de un corazón. La  
seguridad de ser lo que soy es el reflejo de lo que veo en tus ojos*

## Agradecimientos

Agradezco de manera especial a mi familia por todo el apoyo que he recibido y el seguimiento de todo el trayecto de este trabajo, por creer en mí. A mi novio por ayudarme a salir adelante y mostrarme la mejor vía para lograr buenas cosas, ser paciente como la clave del éxito y tener Fe en las cosas que deseamos, por su amor.

Agradezco a mis amigos y a la familia de mi novio, que han sido mi familia todo este tiempo y que me han ayudado siempre que lo he necesitado, me han brindado la sonrisa que a veces me falta por el estrés. No tengo las hojas necesarias para nombrarlos a todos, ni palabras para decirles cuan agradecida me siento.

Agradecimientos especiales a Jorge Amado Soria, que además de considerarlo un amigo, es la persona que me ha apoyado en la realización de este trabajo, todo lo que está reflejado es parte de su conocimiento.

A mi tutora Haydee María Cruz Torres por el apoyo que ha brindado todo el tiempo y su preocupación por la calidad de este trabajo. A Nilet Soto López por aportar esta idea tan brillante que aportó que hoy está hecha realidad. A mi oponente Reynaldo Roselló Núñez por las críticas constructivas que han influido en la mejora de la calidad de este documento. Además a todos mis profesores que intervinieron en mi enseñanza, no solo en esta universidad si no también aquellos que me enseñaron a contar y a escribir.

A todos los que de alguna manera han aportado un granito de ayuda en esta investigación y en vida. A mi infinito Dios, por guiarme por el buen camino y darme las mejores armas de la vida: la Fe, el Amor y la Verdad.

## 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 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 claves:

Calidad del Software, Prueba del Software, Prueba Automática, Herramienta de Prueba, Prueba de Carga y Estrés.

# Índice

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA</b> .....	<b>1</b>
INTRODUCCIÓN .....	1
1.1. PROYECTO CICPC, ANTECEDENTES .....	1
1.2. CALIDAD DEL SOFTWARE .....	2
1.3. PROCESO DE PRUEBAS.....	3
1.4. NIVELES DE PRUEBA.....	6
1.5. TIPOS DE PRUEBAS DE SISTEMA .....	7
1.6. PLANIFICACIÓN DE LAS PRUEBAS .....	10
1.7. HERRAMIENTAS DE AUTOMATIZACIÓN DE LAS PRUEBAS .....	12
1.7.1. <i>Conceptos</i> .....	12
1.7.2. <i>Resumen de Herramientas para Pruebas de Carga y Estrés</i> .....	13
1.7.3. <i>Comparación de las Herramientas</i> .....	15
1.7.4. <i>Herramienta Seleccionada para Automatizar Pruebas de Carga y Estrés</i> .....	21
CONCLUSIONES DEL CAPÍTULO.....	21
<b>CAPÍTULO 2 PROCESO DE PRUEBAS CON JMETER</b> .....	<b>23</b>
INTRODUCCIÓN .....	23
2.1. PROCESO AUTOMATIZADO DE CARGA Y ESTRÉS.....	23
2.2. ETAPA DE PLANIFICACIÓN .....	30
2.2.1. <i>Plan de Pruebas</i> .....	30
2.2.2. <i>Organización del Equipo de Pruebas</i> .....	31
2.2.3. <i>Estrategia de pruebas de Carga y Estrés realizadas en JMeter</i> .....	32
2.2.4. <i>Cronograma</i> .....	33
2.2.5. <i>Recursos Requeridos</i> .....	35
2.2.6. <i>Documentación de los Resultados</i> .....	37
2.3. ELABORACIÓN DE LOS TEST .....	39
2.4. MANUAL PARA REALIZAR PRUEBAS DE CARGA Y ESTRÉS EN JMETER .....	40
2.5. ANÁLISIS DE LOS RESULTADOS .....	41
CONCLUSIONES DEL CAPÍTULO .....	42
<b>CAPÍTULO 3 PRUEBAS DE CARGA Y ESTRÉS EN EL PROYECTO CICPC</b> .....	<b>43</b>
INTRODUCCIÓN .....	43
3.1 PLANIFICACIÓN Y CONCEPCIÓN DE LAS PRUEBAS DE CARGA Y ESTRÉS DEL SIIPOL .....	43
3.1.1 <i>Estrategia de prueba</i> .....	43
3.1.2 <i>Realización de los Test</i> .....	50
3.1.3 <i>Ejecución de las Pruebas</i> .....	51
3.1.4 <i>Análisis de los resultados de las Pruebas</i> .....	52
3.2. PLANIFICACIÓN Y CONCEPCIÓN DE LAS PRUEBAS DEL PORTAL WEB .....	55
3.2.1. <i>Estrategia de prueba</i> .....	55
3.2.2 <i>Realización de los Test</i> .....	59

3.2.3. <i>Ejecución de las Pruebas</i> .....	59
3.2.4 <i>Análisis de los resultados de las Pruebas</i> .....	60
CONCLUSIONES DEL CAPÍTULO .....	61
<b>CONCLUSIONES</b> .....	<b>62</b>
<b>RECOMENDACIONES</b> .....	<b>63</b>
<b>REFERENCIAS BIBLIOGRÁFICAS</b> .....	<b>64</b>
<b>BIBLIOGRAFÍA</b> .....	<b>65</b>
<b>GLOSARIO DE TÉRMINOS</b> .....	<b>67</b>
<b>ANEXOS</b> .....	<b>70</b>



## Índice de Figuras

Fig. 1 Proceso de Pruebas.....	3
Fig. 2 Planificación de las pruebas.....	4
Fig. 3. Diseño de Pruebas.....	4
Fig. 4. Acondicionamiento del Entorno de Pruebas.....	5
Fig. 5. Ejecución de las Pruebas .....	5
Fig. 6. Evaluación de las Pruebas .....	6
Fig. 7 Herramientas de Automatización de Pruebas de Sistema No Funcionales.....	20
Fig. 8 Proceso de automatización de las pruebas de sistema no funcionales de Carga y Estrés .....	23
Fig. 9. Proceso de Automatización de las Pruebas de Sistema no Funcionales de Carga y Estrés .....	25
Fig. 10. Proceso de confección de la estrategia de pruebas automatizadas de carga y estrés .....	27
Fig. 11. Confección de los test de Carga y Estrés.....	28
Fig. 12. Ejecución de las pruebas .....	29
Fig. 13. Análisis de los resultados .....	30
Fig. 14 Análisis del rendimiento para 450 usuarios .....	54
Fig. 15. Análisis del rendimiento del sistema para 900 usuarios.....	54

## Índice de Tablas

Tabla 1.Comparación de Herramientas de Automatización de Pruebas de Sistema No Funcionales ...	17
Tabla 2 Cronograma.....	33
Tabla 3 Ejemplo de Cronograma.....	34
Tabla 4 Especificación de Recursos .....	36
Tabla 5 Ejemplo de Especificación de Recursos .....	36
Tabla 6 Reporte de Estado de las PC.....	37
Tabla 7 Ejemplo de Reporte de Estado de las PC .....	37
Tabla 8 Plantilla de Resultados de las Pruebas.....	39
Tabla 9. Especificación de recursos para las pruebas de Carga y Estrés del SIIPOL.....	44
Tabla 10. Reporte de estado de las PC.....	45
Tabla 11. Cronograma del Proceso de Pruebas Automatizadas del SIIPOL.....	50
Tabla 12. Cronograma de grabación de escenarios a probar .....	50
Tabla 13 Análisis de los Resultados del SIIPOL Para 450 usuarios.....	53
Tabla 14 Análisis de los Resultados del SIIPOL para 900 usuarios .....	55
Tabla 15. Cronograma de actividades del Portal Web del CICPC .....	58
Tabla 16. Resumen de Resultados de las pruebas de carga y estrés del Portal Web .....	60

## Introducción

Para evaluar la calidad de un software se tienen en cuenta una serie de aspectos bien estructurados en normas y en procesos diseñados para un sano empleo. Una de las aristas de las que trata la calidad se puede encontrar un proceso muy importante: Las Pruebas de Software.

Se ha desarrollado una gran expectativa sobre el tema de “Cómo probar software de manera eficiente”. Lo que ha provocado un movimiento de información y de trabajos de investigación al respecto.

Las pruebas de sistema, uno de los niveles de prueba, tienen la responsabilidad de detectar todos los errores del sistema en períodos tempranos enfocándose en los requerimientos funcionales y no funcionales que el cliente especificó en las primeras etapas de concepción del proyecto. Estas se definen en una escala muy amplia, recogidas en grupos determinados y dirigidos a los requisitos del sistema. Entre ellas se encuentran las siguientes:

- ✓ Las *Pruebas de Rendimiento* utilizadas para determinar si el ambiente soporta requisitos de carga de configuración o transición mediante pruebas a escenarios específicos del negocio.
- ✓ Las *Prueba de Stress*, provocan que el sistema o ambiente designado llegue a un punto de saturación (el agotamiento de un recurso: CPU, la memoria, etc.) para determinar si la conducta cambia o se dificulta en el sistema, aplicación o datos.
- ✓ Las *Pruebas de Red* se encargan de monitorear los componentes de red, determinar los requisitos de capacidad de la red, analizar el tráfico existente para identificar anomalías y caracterizar la transición de atributos entre ellos: la cantidad de datos transferidos y el número de usuarios concurrentes.
- ✓ Existen otras tales como: Pruebas de Seguridad, Pruebas de Soporte, Pruebas de Instalación, Pruebas Operacionales, Pruebas de Concurrencia, Pruebas de Portabilidad, Pruebas de Vulnerabilidad, Pruebas de Volumen Funcional, etc.

Se estima que las pruebas de software representan más de la mitad del costo de desarrollo de un programa. Hacer un sistema magistral cuesta mucho esfuerzo, probarlo cuesta más[1].

Aunque no se puede generalizar, en algunas ocasiones los desarrolladores tienen muy poco tiempo para realizar un programa y por tanto se afecta el tiempo que debe dedicar para probar que sus funcionalidades están lo suficiente infalibles para asegurar que el producto no tenga errores. En ocasiones las pruebas de sistema son dirigidas solo a los requisitos funcionales, obviando comprobar en qué medida se cumplen las expectativas de los requisitos no funcionales[2]. Estos ocurren por desconocimiento de la importancia del este proceso de pruebas o por la ausencia de una buena

estrategia en la que se encuentren detallados los procedimientos a seguir para cada una de ellas o por la imposibilidad de llevar a cabo pruebas de gran extensión en cortos períodos de tiempo.

El Proyecto de Modernización del Cuerpo de Investigaciones Científicas, Penales y Criminalística de la República Bolivariana de Venezuela (CICPC) desarrollado en la Universidad precisa de una rigurosa etapa de pruebas de carga y estrés por sus características particulares es necesario someter el software a situaciones de de concurrencia y de flujo de datos extremos. 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 se deriva de esta situación un problema científico: ¿Cómo realizar las pruebas de sistema no funcionales de carga y estrés de manera eficiente y eficaz? Para dar respuesta al mismo se toma como objeto de estudio el proceso de pruebas de sistemas no funcionales de carga y estrés. Teniendo como campo de acción herramientas de ejecución de pruebas de sistema no funcionales de carga y estrés, su aplicación en el Proyecto CICPC. Trazándose como objetivo general investigar y proponer herramientas que de forma automática ejecuten estas pruebas en el proyecto CICPC. Todo ello con el propósito de establecer alguna herramienta para la automatización de las pruebas de carga y estrés en el proyecto CICPC que logre disminuir el tiempo y los recursos empleados en estas pruebas y así obtener mayor calidad en los resultados. Lo cual supone que el uso de herramientas que automaticen el proceso en el proyecto incurrirá en una menor inversión de tiempo y recursos frente al método tradicional y conseguirá una mayor calidad en los resultados obtenidos.

Para llevar a cabo la investigación se concretaron las siguientes tareas:

- ❖ Investigar sobre la ejecución de estos tipos de prueba en la Universidad.
- ❖ Investigar sobre las herramientas libres que se pueden utilizar.

- ❖ Seleccionar las herramientas a utilizar haciendo un análisis comparativo con otras herramientas existentes.
- ❖ Confeccionar un manual que indique como utilizar la herramienta.
- ❖ Ejecutar las pruebas haciendo un uso de la herramienta y exponer los resultados de las pruebas ejecutadas en el proyecto CICPC.
- ❖ Colaborar con la asimilación y entrenamiento de las personas que utilizarán la herramienta.

Entre los **Métodos de Trabajo Científico** utilizados para llevar a cabo esta investigación se destacan los siguientes:

1. **Métodos generales:** El **método histórico-lógico** y el **dialéctico** para el estudio crítico de los trabajos anteriores, y para utilizar estos como punto de referencia y comparación de los resultados alcanzados.
2. **Métodos lógicos:** El método **analítico-sintético** al descomponer el problema de investigación en elementos por separado y profundizar en el estudio de cada uno de ellos, para luego sintetizarlos en la solución de la propuesta.
3. **Método Explicación:** El **método de explicación** porque se elaborará un manual para lo cual se requiere de una explicación del por qué y el cómo se hará.

Se han utilizado algunos de los **métodos empíricos** como son la **investigación** y la **encuesta** ya que a través de ellos es que se ha podido profundizar el estado inicial de esta problemática y todos los aspectos relacionados con el tema; además se han podido contactar personas que están involucradas en el campo de acción que han podido aportar información valiosa para la investigación.

Este documento está compuesto por tres capítulos:

En el Capítulo I expone el estado del arte del trabajo, mostrando los principales conceptos como los de pruebas de carga y estrés, herramientas de automatización y calidad. Además la investigación de herramientas candidatas y elección de una de ellas para realizar las pruebas en el Proyecto CICPC.

El proceso pruebas automatizadas de carga y estrés se muestra de manera estándar en el Capítulo II, recoge todos los elementos de la automatización de estas pruebas. Define además artefactos, roles y actividades. Quedan reflejados los objetivos, composición y misión del manual diseñado, como elemento de apoyo a la hora de trabajar con la herramienta de automatización de estas pruebas.

Por último en el Capítulo III se aplica el proceso definido en el Capítulo II a los productos realizados por el Proyecto CICPC. Sirve de ejemplo para la utilización de una herramienta para la realización de las pruebas de carga y estrés, explica los pasos necesarios para realizar estas pruebas y los entregables que se generan.

# Capítulo

# 1

## Fundamentación Teórica

### Introducción

Este capítulo se expone los temas fundamentales que sustentan la investigación. Se mencionan conceptos de calidad, pruebas y herramientas. Aborda de manera general el proceso de pruebas de software, enfatizando las pruebas de sistema no funcionales como tema inicial para este trabajo. Se muestra un estudio de las herramientas que existen para la ejecución de pruebas de carga y estrés, así como comparaciones entre ellas. Se fundamenta la herramienta seleccionada para la solución del problema.

#### 1.1. Proyecto CICPC, Antecedentes

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. Los detalles que originan este proyecto son de carácter confidencial y se encuentran documentados en el Proyecto Técnico [3].

El SIIPOL es realizado en la plataforma Java 2 Enterprise Edition (Java2EE) y base de datos Oracle. Mientras que el Portal Web fue realizado utilizando el sistema de administración de contenido para sitios Web Drupal, lenguaje PHP y base de datos MySQL. Por las características del proyecto se aplica la metodología RUP[4] como guía de desarrollo.

La misión principal de este proyecto es realizar estas aplicaciones para la institución (CICPC) mejorando la gestión de información policial. El equipo de pruebas de calidad interna del CICPC que fue creado para supervisar la calidad de los artefactos generados por el proyecto, está compuesto por un grupo de probadores, analistas de prueba, diseñadores de pruebas y un gestor de pruebas. Los artefactos más importantes que se utilizan en el aseguramiento de calidad en el entorno de prueba son:

- Cronograma General de las Pruebas: Expresa la fecha de inicio de las iteraciones de prueba según el tipo de prueba que se le realizan al software, fecha final para realizar estas tareas y los responsables de las mismas.

- Plan de Pruebas: Expone las pruebas que se realizan en cada aplicación, la misión, objetivos, alcances, herramientas involucradas en este proceso, responsables, artefactos y cronograma.
- Estrategia de pruebas: Define cómo se realizan las pruebas, cuando y los recursos necesarios para realizar el proceso.
- Casos de Prueba: Exponen procedimientos, juegos de datos y toda la información que se necesita para ejecutar las pruebas.
- Documentos de No Conformidad: Exponen los errores o anomalías encontrados en las pruebas.

Anterior a esta investigación se realizó una Propuesta de manual de procedimiento de Pruebas de Sistema y su aplicación en el Proyecto CICPC(2007)[5]. Este refleja el proceso de pruebas de sistema funcionales y sugiere las pruebas de sistema no funcionales entre ellas las pruebas de carga y estrés que deben realizarse en el proyecto.

## 1.2. Calidad del Software

Una empresa de software debe tener presente aspectos importantes de la calidad, todos ellos en función de las expectativas del cliente. La rapidez de la implementación del software y su calidad son aspectos de gran importancia. Este último ha sido estudiado por muchos expertos los cuales lo definen 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” [6].*

*“El conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas”.[7]*

El término calidad de software por sí solo no tiene validez, la calidad del software está dada en la definición y ejecución de aspectos como:

- ✓ Métodos y herramientas de análisis, diseño, programación y prueba.
- ✓ Inspecciones técnicas formales en todos los pasos del proceso de desarrollo del software.
- ✓ Estrategias de prueba multiescala.
- ✓ Control de la documentación del software y de los cambios realizados.
- ✓ Procedimientos para ajustarse a los estándares (y dejar claro cuando se está fuera de ellos).
- ✓ Mecanismos de medición (métricas).
- ✓ Registro de auditorías y realización de informes.

### 1.3. Proceso de Pruebas

El objetivo de la fase de pruebas es detectar las anomalías que posee el software.

Según Pressman en su libro Ingeniería del Software. Un enfoque práctico[7]: *“La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación”*.

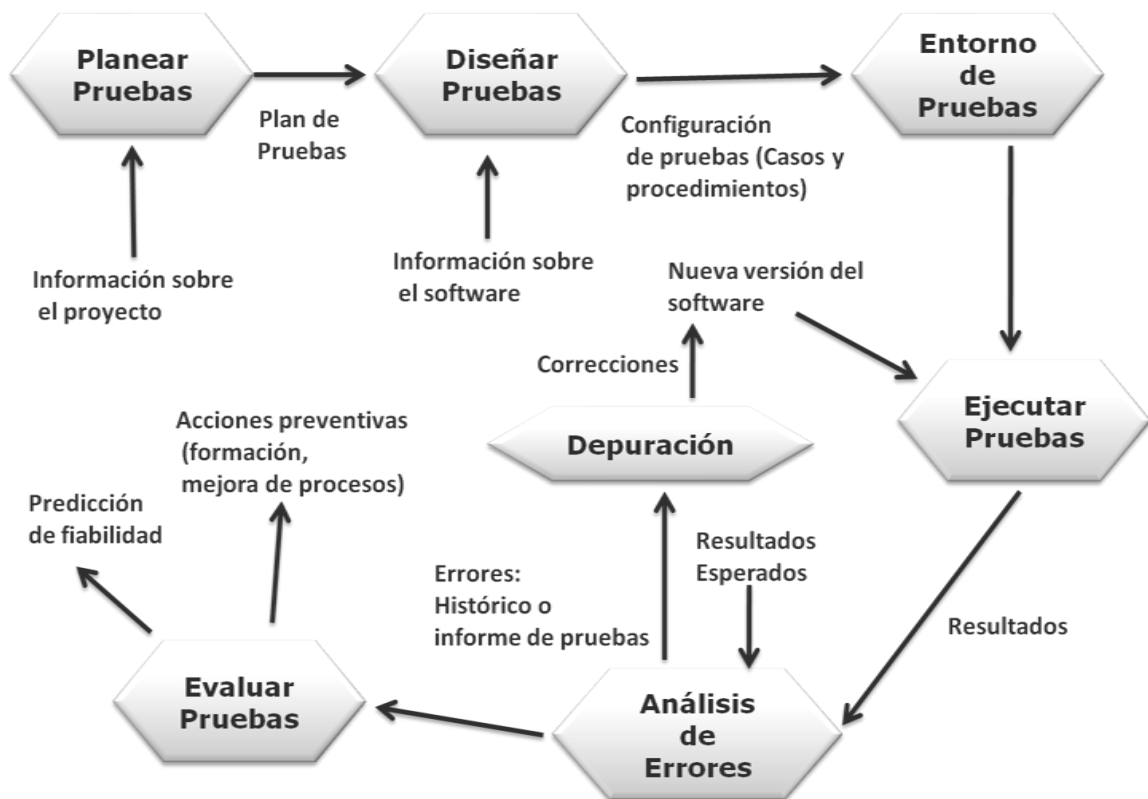
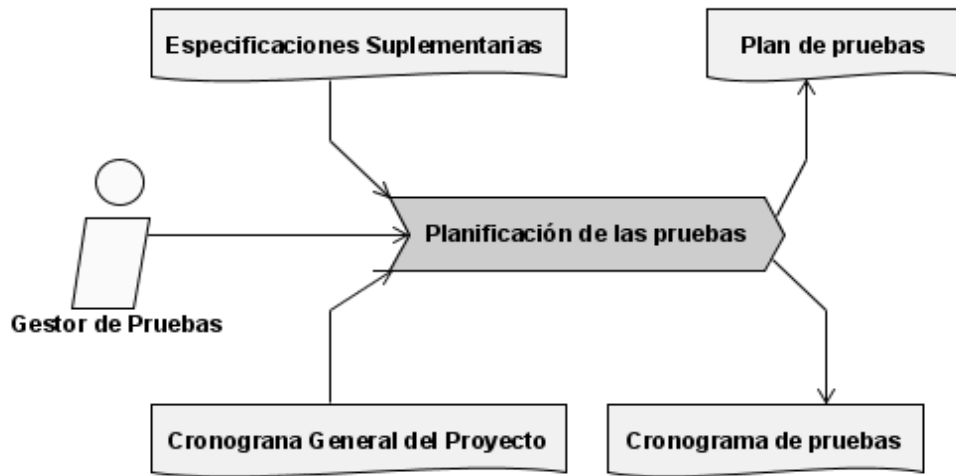


Fig. 1 Proceso de Pruebas

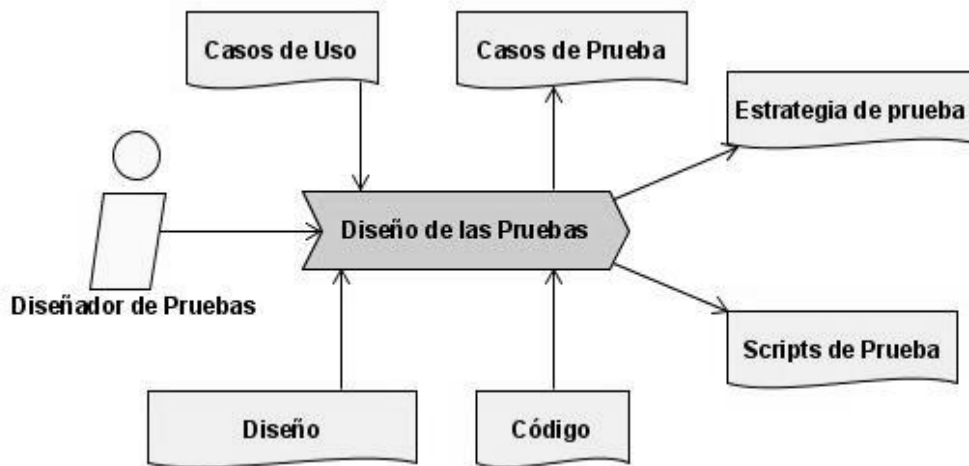
La figura 1 muestra el proceso de pruebas compuesto por un conjunto de procesos, siendo el inicial el más importante, debido a que se planean las pruebas que son necesarias ejecutar. Se concibe el Plan de Pruebas, la estrategia de pruebas y el cronograma. Todos estos artefactos se generan basadas en las Especificaciones suplementarias y el Cronograma General del proyecto. Estos documentos son realizados por el Gestor de Pruebas. (Ver fig.2)





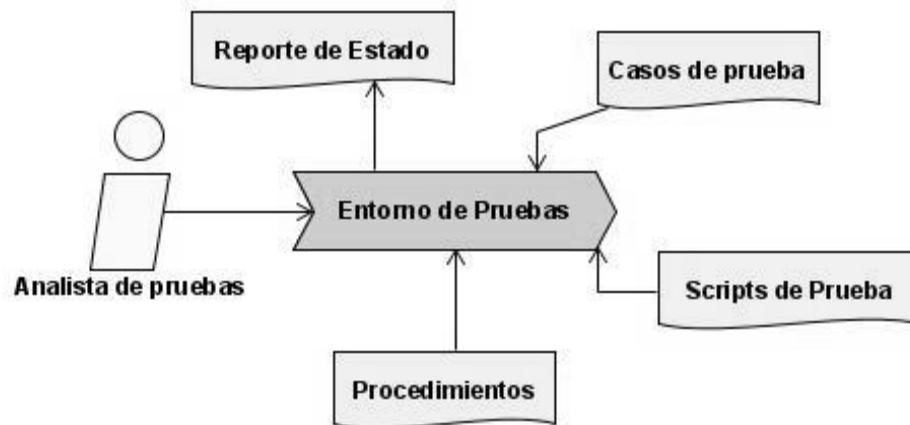
**Fig. 2 Planificación de las pruebas**

En diseño de las pruebas se crean los pasos que deben realizarse para obtener los casos de prueba o scripts de prueba si se utiliza una herramienta de automatización de pruebas. Se realiza la estrategia de pruebas y otros artefactos que brindan información necesaria para la ejecución de las pruebas. (Ver fig.3)



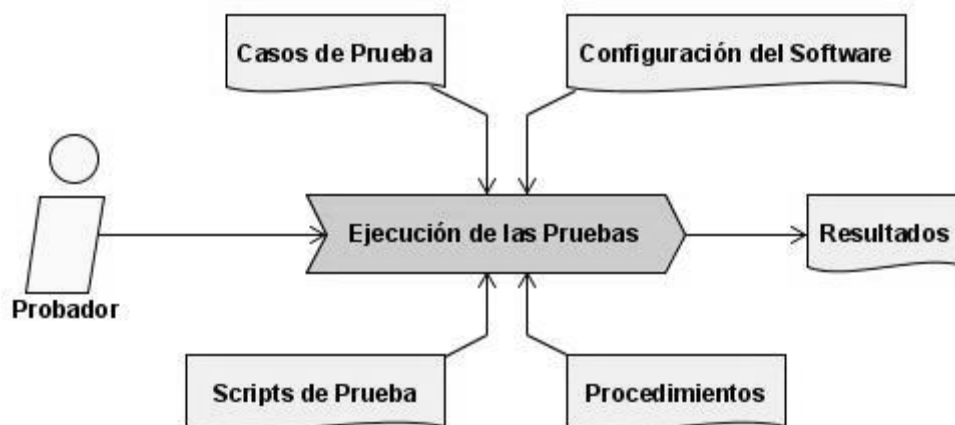
**Fig. 3. Diseño de Pruebas**

El entorno de pruebas es la preparación del escenario donde se ejecutarán las pruebas, interviene el analista de pruebas utilizando los artefactos y la información necesaria el entorno de pruebas. (Ver fig. 4)



**Fig. 4. Acondicionamiento del Entorno de Pruebas**

La ejecución se realiza después de alistar el entorno de las pruebas, es llevada a cabo por el probador utilizando los artefactos que definen los aspectos a probar en el software. (Ver fig.5)



**Fig. 5. Ejecución de las Pruebas**

Teniendo los resultados de las pruebas se realiza un análisis de los resultados, comparando estos con los resultados esperados y luego se evalúan estos resultados por el evaluador. (Ver fig.6)

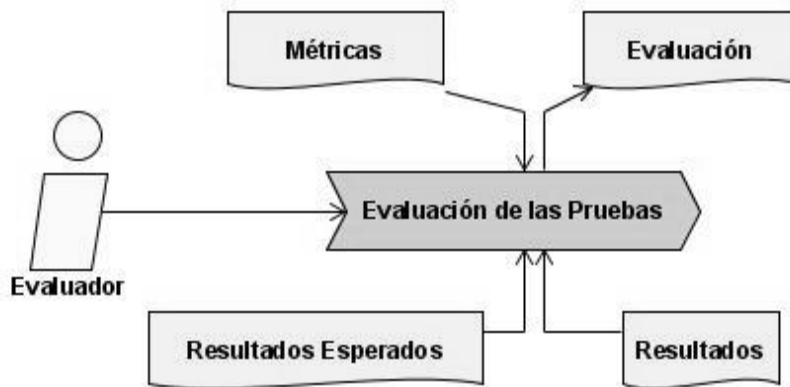


Fig. 6. Evaluación de las Pruebas

#### 1.4. Niveles de Prueba

En el proceso de pruebas al software de niveles en los cuales se ejecutan diferentes tipos de pruebas de acuerdo al desarrollo que se haya alcanzado en la aplicación. Estos niveles especifican qué tipos y métodos de pruebas se deben utilizar en cada uno de ellos y cuáles son sus objetivos.

Los niveles de prueba por los que transita un software son[5]:

❖ De Unidad

- Enfocadas al código fuente de los componentes.
- Para verificar todos los flujos de control.
- Primero pasa por la revisión del programador.

❖ De integración

- Prueba los componentes combinados para ejecutar una funcionalidad.
- Para verificar descubrir errores o incompletitud en las especificaciones de las interfaces de las clases.

❖ De sistema

- Prueba el software funcionando como un todo.
- Aceptable para cuando el software se encuentra en la Fase de Construcción.
- Trata de probar que los objetivos para los que fue construida la aplicación no se cumplen en su totalidad y que por tanto hay que cambiar cosas en la aplicación.
- Se usa como base los objetivos originales.
- No existe un método en específico, sino que se dan lineamientos, a la hora de preparar los casos de prueba.

- Se finaliza cuando se cumple el tiempo estimado y se han detectado N errores.
- ❖ De liberación
  - Prueba el software funcionando como un todo.
  - Trata de probar que los objetivos para los que fue construida la aplicación no se cumplen en su totalidad y que por tanto el software no está en condiciones de ser presentado al cliente y hay que hacer modificaciones a la aplicación.
  - Se realiza cuando el software funciona como un todo y está prácticamente listo para ser presentado al cliente.
- ❖ De aceptación
  - Prueba el software funcionando como un todo, se realiza dándole un uso real a la aplicación y es realizada por parte de los clientes, los errores que se encuentren en la misma son reportados como defectos.

Las pruebas del software se encuentran en cada etapa del desarrollo del software. En la medida que el trabajo de los desarrolladores va aumentando el volumen de la aplicación, las pruebas van cambiando de estrategias y técnicas, presentándose como una nueva etapa, estas están definidas en niveles que tienen nuevos objetivos, entornos y resultados.

## 1.5. Tipos de Pruebas de Sistema

Las pruebas de sistema están divididas en dos grandes grupos orientadas a la funcionalidad y a las respuestas del software en el ambiente en el que se debe ejecutar[8]:

### **Funcionales**

Enfocadas a los requisitos funcionales del software. Comprueban el correcto comportamiento de las acciones que debe realizar el sistema.

### **No funcionales**

Enfocadas a los requisitos no funcionales del proyecto. Definidas como[5]:

### **Prueba de Seguridad**

#### **Objetivo**

Verificar que los mecanismos de protección incorporados en el sistema realmente lo protegerán de accesos impropios.

#### **Metas**

Validar que en la aplicación:

- ✓ Los datos del sistema solo pueden ser accesibles por los autores debidamente autorizados.
- ✓ Las funciones del sistema solo pueden ser accesibles por los autores debidamente autorizados.
- ✓ Las funciones que atenten contra la integridad de los datos de negocios sean debidamente impedidas.

### **Prueba de Disponibilidad y Red**

#### **Objetivo**

Verificar el comportamiento de la aplicación cambiando la infraestructura de red al aplicar diferentes configuraciones, o retardos.

#### **Metas**

Validar que en la aplicación:

- ✓ No se reduzca la disponibilidad de los sistemas dentro de la solución, debido a la actividad de alguna persona o sistema. Ya sea, accidental o malintencionado.

### **Prueba de Rendimiento o Carga**

#### **Objetivo**

Verificar el rendimiento del software en tiempo de ejecución dentro del contexto de un sistema integrado y así como la verificación de la capacidad del sistema para manejar volúmenes de datos extremos de acuerdo al tiempo de respuesta establecido para el sistema para el sistema.

#### **Metas**

Validar en la aplicación:

- ✓ Comprobar los tiempos de respuesta del sistema en una cantidad limitada de escenarios de trabajo (a nivel de número de usuarios y número de transacciones), bajo una configuración de hardware y software constante.
- ✓ Comprobar el tiempo de respuesta al realizar una función.
- ✓ Comprobar el tiempo de respuesta al realizar accesos concurrentes a una determinada información.
- ✓ Atender múltiples solicitudes de parte de los actores que acceden a un mismo recurso.

### **Prueba de Compatibilidad**

#### **Objetivo**

Verificar el funcionamiento del sistema sobre diferentes componentes de software.

**Metas**

Validar en la aplicación:

- ✓ Sistemas Operativos.
- ✓ Navegadores.

**Prueba de Resistencia o Stress****Objetivo**

Verificar cómo se comporta el sistema bajo condiciones anormales:

**Metas**

Validar en la aplicación:

- ✓ Carencia de sistemas externos con los que interactúa el sistema.
- ✓ Aplicar carga excesiva de trabajo al sistema (extremas sobrecarga)
- ✓ Hardware no disponible.
- ✓ Recursos compartidos no disponibles.
- ✓ Además verifica qué hace el sistema cuando el usuario no hace lo que supuestamente debe hacer.

**Prueba de Usabilidad****Objetivo**

Determinar si la organización de los contenidos y las funcionalidades que se ofrecen desde la aplicación son entendidas y utilizadas por los usuarios de manera simple y directa.

**Metas**

Validar en la aplicación:

- ✓ Para poder ver las páginas adecuadamente necesita utilizar un navegador compatible con estándares Web.
- ✓ Proporcionar al usuario información relacionada con el estado actual del sistema.

**Prueba de Fiabilidad****Objetivo**

Verificar la probabilidad de que ese sistema funcione o desarrolle una cierta función, bajo condiciones fijadas y durante un período de tiempo determinado.

**Meta**

Validar en la aplicación:

- ✓ El sistema deberá estar disponible las 24 horas.
- ✓ El tiempo permitido que el sistema puede estar fuera de operación después de un fallo.

## 1.6. Planificación de las Pruebas

### Objetivo

El objetivo global de esta estrategia debe alcanzarse. Por ejemplo, para una de las pruebas de sistema, este objetivo puede ser una declaración de que todos los requerimientos funcionales deben comportarse de acuerdo a lo esperado, o como quedó documentado.

### Técnica

Especifica cómo los casos de prueba serán desarrollados, el instrumento o herramienta usado para almacenarlos y donde pueden ser encontrados; cómo ellos serán ejecutados y los datos que serán usados.

### Entorno de Prueba

Se detallan las condiciones de hardware y configuración bajo las cuales se deben realizar las pruebas.

### Proceso

Breve descripción del proceso que se realiza.

### Casos de Prueba

Se realiza una lista detallada o una referencia a los casos reales de prueba.

### Criterios de Término

Registrar los criterios que serán usados para determinar la aprobación o rechazo de pruebas y la acción que debe ser tomada con base en los resultados de la prueba.

### Herramientas

Documentar los instrumentos o herramientas que serán empleados para las pruebas. Citar al proveedor, la versión y el número de la Mesa de Ayuda para pedir el apoyo, si fuera necesario.

### Recursos Requeridos

Identificar los roles y las responsabilidades que serán requeridas para la ejecución del Plan de Pruebas asociado.

### Plan de proyecto

Parte del cronograma del proyecto que abarca la etapa de pruebas.

### Calendario y Plazos.

Se documenta el plazo en el cual la aplicación a probar estará disponible para pruebas y el tiempo estimado para ejecutar los casos de prueba. Se especifica si se proporcionará partes construidas, sobre una base regular durante el ciclo de prueba, o cuando se espera que los componentes del sistema estén listos para pruebas.

### Definición de los Entregables

Se añade en una lista cualquier entregable asociado con el esfuerzo de pruebas y donde las copias de

estos entregables o documentos pueden ser localizados, esto incluye el Plan de Pruebas en sí mismo, escenarios para prueba, casos de prueba y el plan de proyecto.

### **Seguimiento y Reporte de Defectos**

Se documenta el instrumento y el proceso usado para registrar y rastrear los defectos. Se añade en una lista todos los informes que serán generados incluyendo repositorios, frecuencias, mecanismos de entrega y ejemplos. Se identifican los recursos involucrados en el proceso de seguimiento.

Se describe la categoría o clasificación que se usará para identificar o priorizar defectos. Las siguientes son categorías se utilizaron en las pruebas funcionales realizadas al caso de estudio que sirvió de preparación y entrenamiento para todo el equipo del CICPC[5], para priorizar o calificar defectos:

- **Crítico:** Denota una función inutilizable que causa un término anormal o una falla general, o cuando un cambio en un área de la aplicación causa un problema en otra parte.
- **Severo:** Una función no actúa como fue requerido o diseñado, o un objeto de interfaz no trabaja como se muestra.
- **Advertencia:** La función trabaja, pero no tan rápidamente como esperado, o no se ajusta a las normas y convenciones.
- **Cosmético:** No crítico para el funcionamiento de sistema: palabras con mala ortografía, formato incorrecto, mensajes de error vagos o confusos o advertencias.

Por las características de los resultados de las pruebas automatizadas de carga y estrés se toman las siguientes calificaciones para identificar los resultados:

- **Crítico:** La aplicación no soporta la cantidad de usuarios estimados para la cual fue diseñada e implementada.
- **Severo:** En caso de falla de la aplicación, esta no es capaz de reponerse en el tiempo definido para la reposición del sistema y esta no puede lograrlo por sí misma.
- **Cosmético:** Una petición no es encontrada. O sea falla la petición por la ausencia del elemento al que esta hace referencia o por un mal funcionamiento de la aplicación.
- **Advertencia:** Los tiempos de respuestas de las aplicaciones son superiores a los tiempos de respuestas esperados. (Teniendo en cuenta el ambiente en el que se encuentra instalado el servidor).



## 1.7. Herramientas de automatización de las pruebas

### 1.7.1. Conceptos

Según la metodología RUP[4] 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.

### 1.7.2. Resumen de Herramientas para Pruebas de Carga y Estrés

Para el equipo de pruebas de calidad interna del proyecto CICPC sería ineficiente realizar las pruebas de carga y estrés de manera manual y prácticamente imposible comprobar si se cumplen los requerimientos descritos por el cliente, en algunos si para uno de ellos se necesitaría al menos 900 personas para su verificación[9].

En vistas a encontrar una herramienta que fuera capaz de dar respuesta a esta situación se realizó una investigación encontrando 4 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].

Compuware (NASDAQ: CPWR) es líder mundial en proporcionar software y servicios profesionales que permiten a las empresas gestionar sus negocios y maximizar el valor de sus activos de TI. Las soluciones de Compuware aceleran el desarrollo, aumentan la calidad y mejoran el rendimiento de los sistemas críticos, mejorando su eficiencia, el control de costos y la productividad de sus empleados a lo largo de toda la organización. Fundada en 1973, Compuware trabaja para las mayores organizaciones de TI del mundo, incluyendo a más del 90 por ciento de las empresas del Fortune 100[11].

### **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[12].

Mercury Interactive empresa que realizó la herramienta Load Runner es una compañía especializada en Business Technology Optimization (BTO), española que recientemente ha sido comprada por HP. [13]

### **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. [14]

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]

JMeter 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.

Se necesita tener una Máquina Virtual Java 1.3 o superior instalada en el sistema operativo Windows XP Professional en este caso.

### **1.7.3. Comparación de las Herramientas**

Es necesario efectuar una comparación donde se compendien aspectos importantes en las herramientas que sirvan de apoyo a una justa elección para la elección de la herramienta que se aplicará en este proceso de pruebas.

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.

La comparación de estas herramientas se encuentra en la tabla siguiente, donde muestran de manera explícita cada una de las características.

Tabla 1. Comparación de Herramientas de Automatización de Pruebas de Sistema No Funcionales

Herramientas	Funcionalidades	Modo de adquisición	Materiales de apoyo	Entorno en el que fue desarrollado
<b>QALoad</b>	<ul style="list-style-type: none"> <li>▪ Automatiza pruebas de carga para web, Java, .NET, aplicaciones ERP y CRM</li> <li>▪ Simula miles de usuarios</li> <li>▪ Detecta problemas, optimizar el desempeño de los sistemas y ayudar a asegurar un despliegue de aplicaciones.</li> </ul>	<ul style="list-style-type: none"> <li>▪ La adquisición de esta herramienta es a través de la compra de la licencia a la empresa propietaria.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Poca existencia de manuales para el modo de empleo y difícil acceso a los mismos ya que son vendidos en conjunto de la licencia de la herramienta.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Esta herramienta es desarrollada por empresas propietarias fuera del marco del software libre.</li> </ul>
<b>LoadRunner</b>	<ul style="list-style-type: none"> <li>▪ Realiza pruebas de carga</li> <li>▪ Permite pre-ver el comportamiento y el rendimiento del sistema.</li> <li>▪ Permite la simulación de la actividad de miles de usuarios.</li> <li>▪ Permite la reutilización de los scripts generados para los casos de prueba.</li> <li>▪ Presenta una interfaz API abierta que permite integrar la herramienta a</li> </ul>	<ul style="list-style-type: none"> <li>▪ Esta permitido la utilización de manera gratuita de algunas de sus funcionalidades como la elaboración del plan de pruebas de manera gratuita por 15 días hábiles después de los cuales es necesario la obtención de la licencia por medio de la compra a la</li> </ul>	<ul style="list-style-type: none"> <li>▪ Existe documentación sobre la herramienta ya que en la Web origen permiten bajar un beta de la aplicación, permitiendo su utilización por 15 días hábiles y este ejecutable trae anexo un tutorial para el estudio de su utilización.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Esta herramienta es desarrollada por empresas propietarias fuera del marco del software libre.</li> </ul>

## FUNDAMENTACIÓN TEÓRICA

	otros entornos de trabajo.	empresa propietaria Mercury Interactive.		
<b>Quality Center</b>	<ul style="list-style-type: none"> <li>▪ Permite automatizar los procesos de calidad.</li> <li>▪ Maneja y automatiza el proceso de entrega, con indicadores claves de rendimiento, a tiempo real.</li> <li>▪ Proporciona aplicaciones que automatizan todas las actividades claves en los procesos de calidad.</li> <li>▪ Reduce los riesgos de depuración de la aplicación.</li> </ul>	<ul style="list-style-type: none"> <li>▪ La obtención de esta herramienta es a través de la compra de la licencia.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Se puede bajar documentación desde la Web que hace referencia a la misma.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Esta herramienta es desarrollada por empresas propietarias fuera del marco del software libre.</li> </ul>
<b>JMeter</b>	<ul style="list-style-type: none"> <li>▪ Permite realizar Pruebas de Rendimiento y Pruebas Funcionales sobre Aplicaciones Web. Se definen otras funcionalidades a continuación de esta tabla.</li> </ul>	<ul style="list-style-type: none"> <li>▪ No se necesita la utilización de una licencia ya que se permite bajar de manera gratuita el software desde la Web que representa al proyecto Jakarta</li> </ul>	<ul style="list-style-type: none"> <li>▪ Existe una amplia documentación sobre la utilización el modo de empleo de la herramienta y los requerimientos no funcionales que ella precisa.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Esta herramienta esta fue realizada en el marco del software libre.</li> </ul>

Estas son otras funcionalidades que brinda la herramienta JMeter:

- Permite realizar test de FTP, JDBC, JNDI, LDAP, SOAP/XML-RPC, y WebServices (en Beta).
- Ofrece la posibilidad que el propio usuario desarrolle en Java un “Controller” a medida, cumpliendo una interfaz Java, y depositando el .jar correspondiente al desarrollo en el directorio “lib” de JMeter lo que permite reducir el consumo de rendimiento del CPU.
- Permite realizar pruebas distribuidas en distintos ordenadores que actuarán como clientes simulando varios hilos que harán función de usuarios.
- Permite generar un caso de prueba es a través de una navegación de usuario.

Tipos de informes que puede generar:

- Assertion Results: Muestra la URL de cada petición e indica los errores que se produzcan (Assertions que no se han cumplido) en el test.
  - Graph Full Results: Simplemente muestra el tiempo.
  - Graph Results: Muestra un gráfico con los tiempos medio, desviación, throughput, etc. de la ejecución del plan de prueba.
  - Mailer Visualizar: Permite enviar un e-mail si el plan de pruebas falla o no, o supera un determinado valor de fallos o éxitos.
  - Simple Data Writer: Vuelca los resultados a un fichero.
  - Spline Visualizer: Gráfico de tiempos como spline.
  - Aggregate Report: Muestra una tabla con una fila por cada URL solicitada, indicando el tiempo min, max, medio, etc. Es una tabla que totaliza por URL.
  - View Results in Table: Muestra una tabla con todas las respuestas, la URL, tiempo y resultado de ejecución de cada una de ellas.
  - View Results in Tree: Muestra un árbol con todas las respuestas y sus tiempos.
- Permite controlar todos los parámetros de una request http:
    - Método: GET o POST.
    - Path del recurso a pedir.
    - Redirección automática: Seguir las redirecciones indicadas por el resultado de la petición.
    - Use KeepAlive: Mantener la conexión viva entre distintas peticiones
  - Permite envío de parámetros en la request.
  - Permite envío de un fichero adjunto a la request.



- Permite especificar el número de threads (hilos de ejecución) en paralelo, así como el tiempo de arranque de cada uno, y número de iteraciones que hará cada uno de ellos (puede marcarse como infinito).
- Se puede planificar (scheduler, función automática para el arranque de la aplicación) la ejecución de la prueba indicando la hora de arranque y parada, o la duración del test en segundos y el tiempo de arranque del mismo.
- Permite la reutilización de los scripts realizados para la prueba.



**Fig. 7 Herramientas de Automatización de Pruebas de Sistema No Funcionales**

Además se analizó la herramienta Quality Center que no es una herramienta de automatización de pruebas de carga y estrés pero ha sido objeto de estudio por ser una herramienta que apoya el proceso de las pruebas y su utilización influye directamente en los resultados satisfactorios de las pruebas. Según la clasificación anterior, es una herramienta 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.

#### **1.7.4. Herramienta Seleccionada para Automatizar Pruebas de Carga y Estrés**

Se ha decidido utilizar para las pruebas Carga y Estrés la herramienta JMeter por las funcionalidades que presenta, por utilizarse en proyectos 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 en el epígrafe 1.7.1 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, supone 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 512mb de RAM. Para una mejor utilización del rendimiento del CPU es necesario utilizar la herramienta Ant, libre y gratuita, que facilita la ejecución de estos test de manera útil pero la utilización de la misma supone restricciones en cuanto a la graficación de los resultados y acciones directas que se pueden realizar en la ejecución de estos test en la herramienta JMeter.

### **Conclusiones del Capítulo**

En este capítulo se ha realizado descripción del proceso de pruebas, en específico los tipos de pruebas de sistema no funcionales con el objetivo de ubicar al lector en el campo que se enmarca el estudio de este trabajo.

Se muestran además las herramientas que han sido encontradas a lo largo de la investigación, no quiere decir ello que estas sean las únicas herramientas existentes, ya que es imposible dominar la totalidad de herramientas de este tipo utilizadas en todo el mundo. No obstante estas herramientas que se presentan son las de mayor utilización en el exterior del país y existe una amplia documentación que hace referencia a la utilización de las mismas para las pruebas de proyectos realizados en J2EE. La comparación de las herramientas ha demuestra que la herramienta JMeter es la mejor candidata para la automatización de las pruebas de sistema no funcionales para el proyecto CICPC. Ya que posee varias características que abogan por un exitoso empleo en dicho proceso.

# Capítulo 2

## Proceso de Pruebas con JMeter

### Introducción

Se definen subprocesos, actividades y artefactos que intervienen en la planificación, concepción y realización de las pruebas. Cada uno de estos elementos, están estrechamente relacionados y definidos sobre la base del aseguramiento de la calidad del proceso de pruebas.

Este capítulo 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. Este manual está fundamentado sobre estudios previos en la fase de pruebas descrito en la Metodología RUP[4], la calidad en las mismas según Pressman[7] y además la norma ISO 8402[16].

### 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.

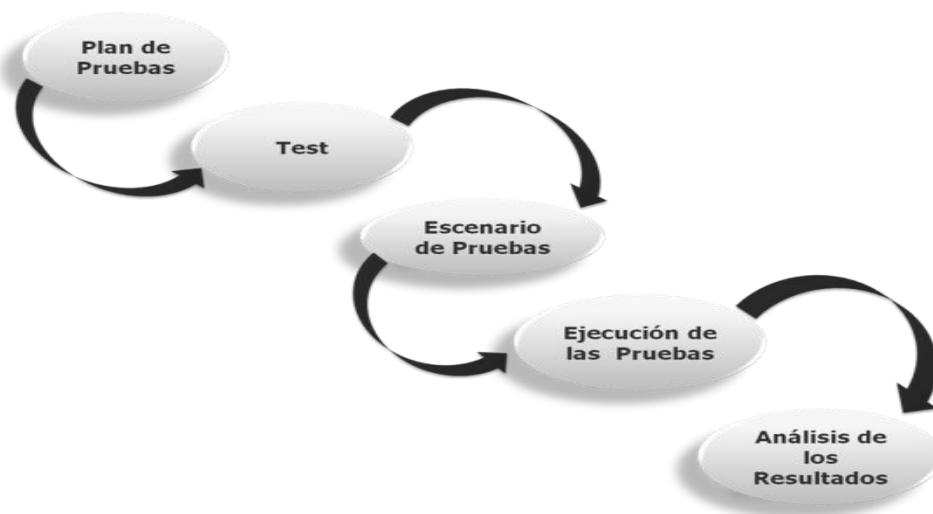


Fig. 8 Proceso de automatización de las pruebas de sistema no funcionales de Carga y Estrés

Estas 5 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.

La figura 9 muestra las etapas reconocidas por la herramienta JMeter para la planificación, concepción y aplicación de las pruebas. Estas etapas están descritas a lo largo de este capítulo en el que se pretende mostrar un método eficiente para la aplicación de herramientas automáticas al proceso de pruebas de carga y estrés. Este proceso se encuentra modelado en la figura 3.

## PROCESO DE PRUEBAS CON JMETER

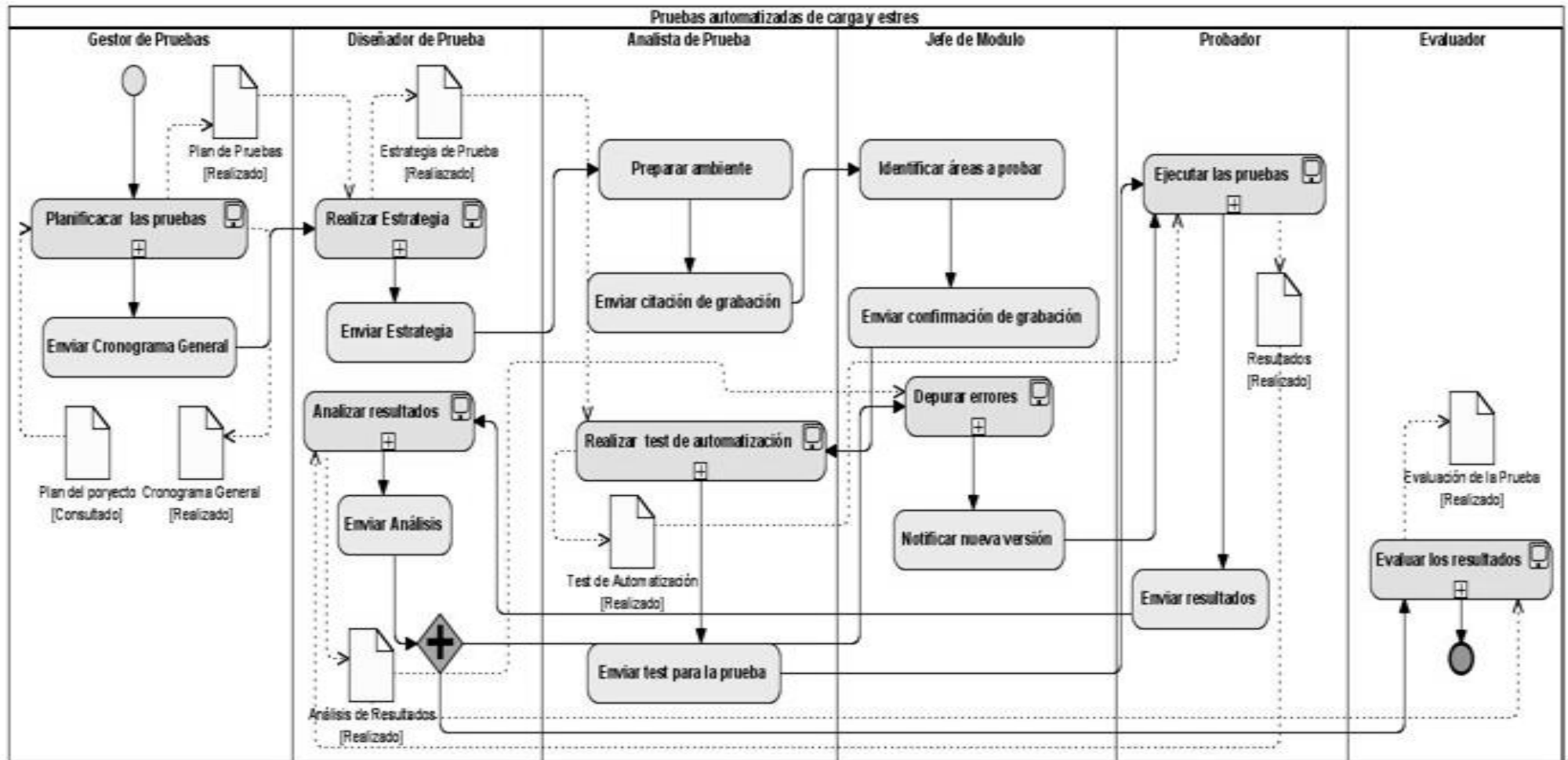
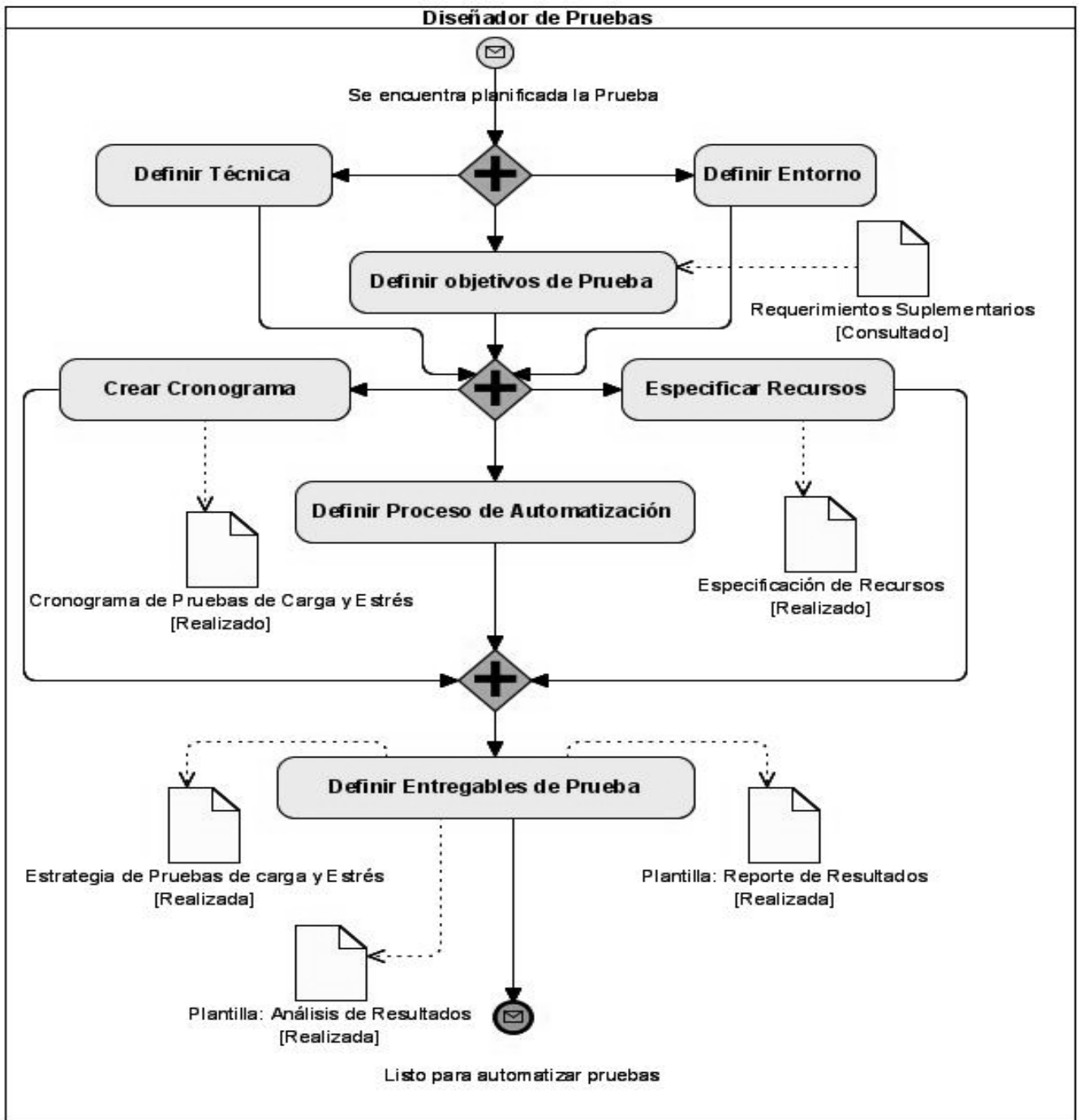


Fig. 9. Proceso de Automatización de las Pruebas de Sistema no Funcionales de Carga y Estrés

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 en 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.

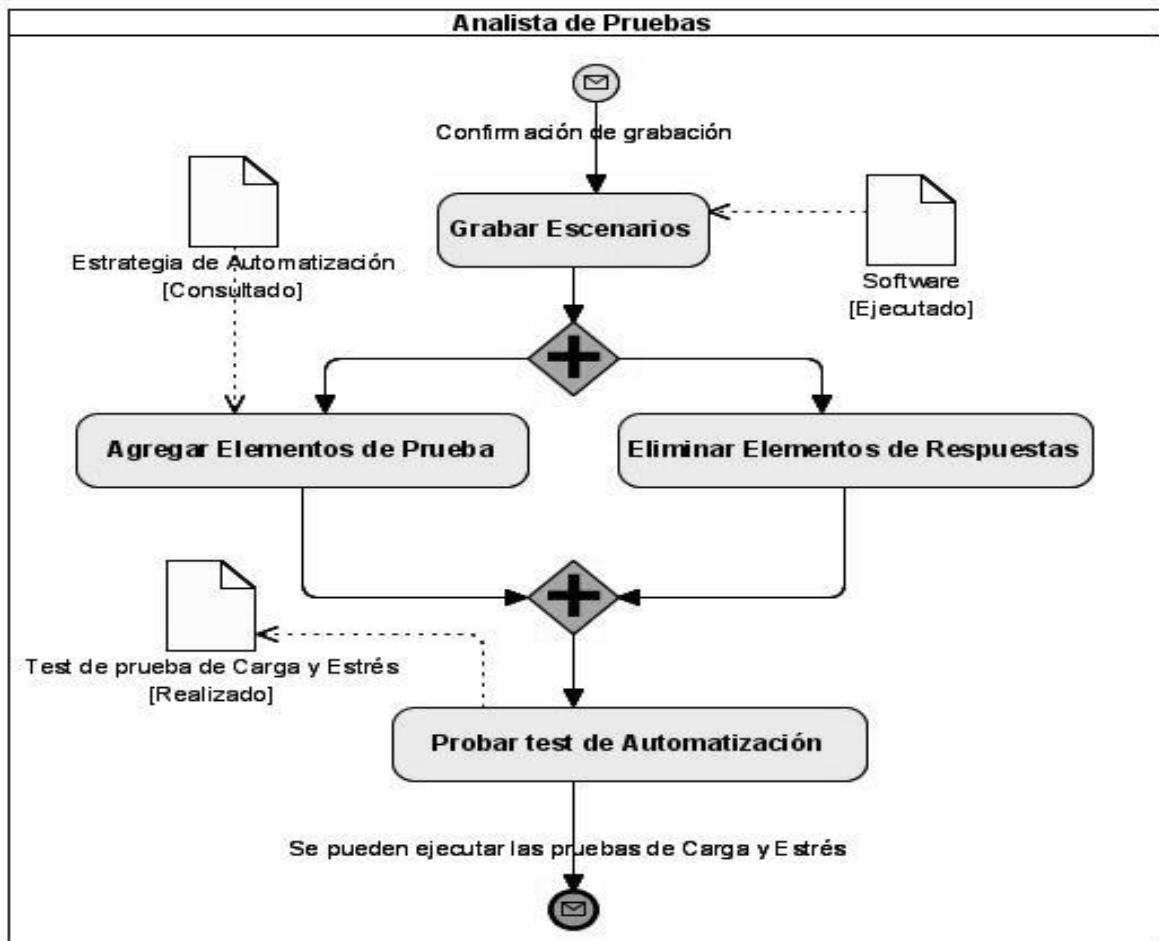


**Fig. 10. Proceso de confección de la estrategia de pruebas automatizadas de carga y estrés**

Este proceso se inicia cuando el diseñador recibe una notificación que especifica la planificación de las pruebas de carga y estrés. Para la confección de la misma se define la técnica que se utilizará, los objetivos para la cual está orientada, además se detalla el entorno en el que se realizará. Cuando se encuentran precisos estos elementos se crea un documento especificando los recursos necesarios

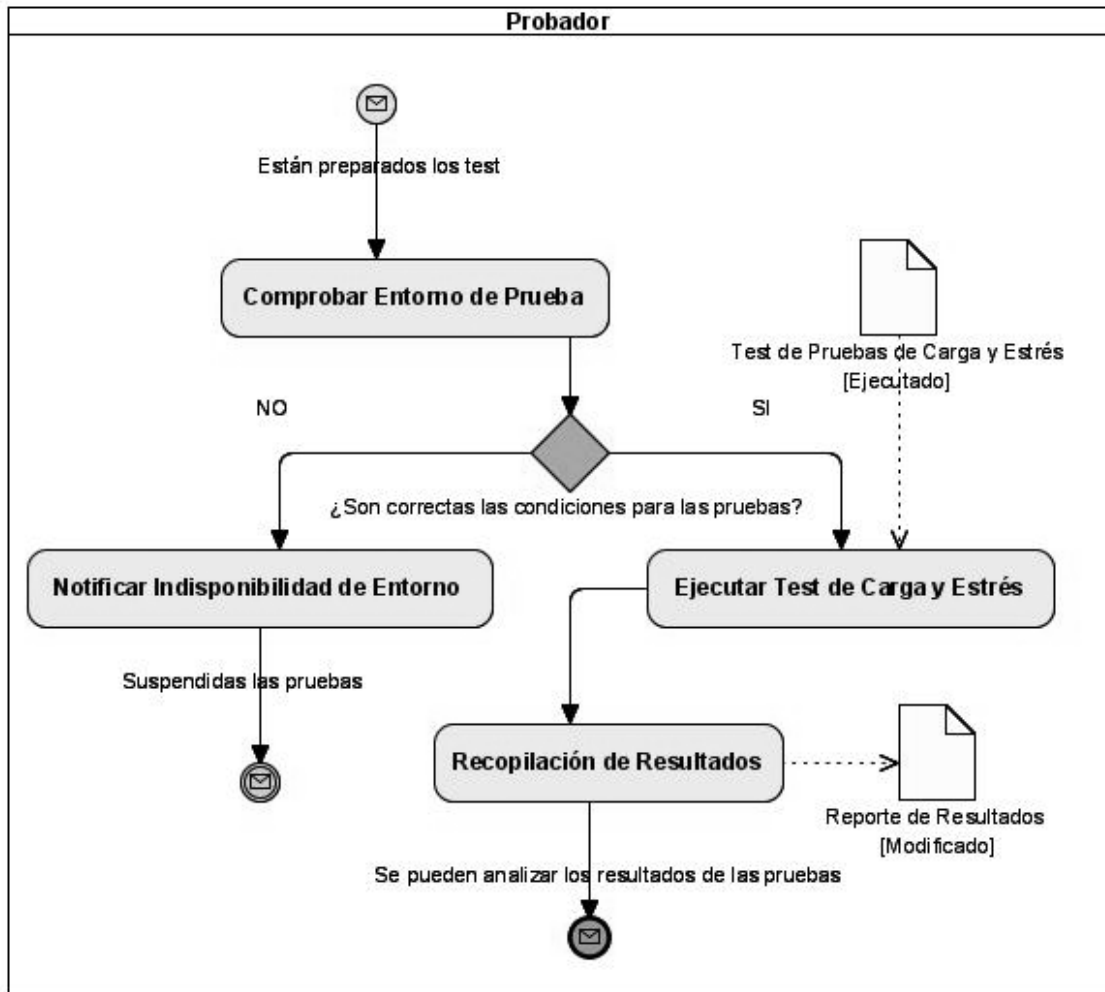


para el desarrollo del proceso y a su vez un cronograma que plantea fechas y horarios para cada una de las actividades que se ejecutarán en lo adelante además del cómo se realizará el proceso, cada paso y cada especificación para la confección de los test. Para finalizar el proceso se definen los entregables que intervendrán en lo adelante y se envía el documento de estrategia al analista de pruebas para que proceda a la confección de los test.



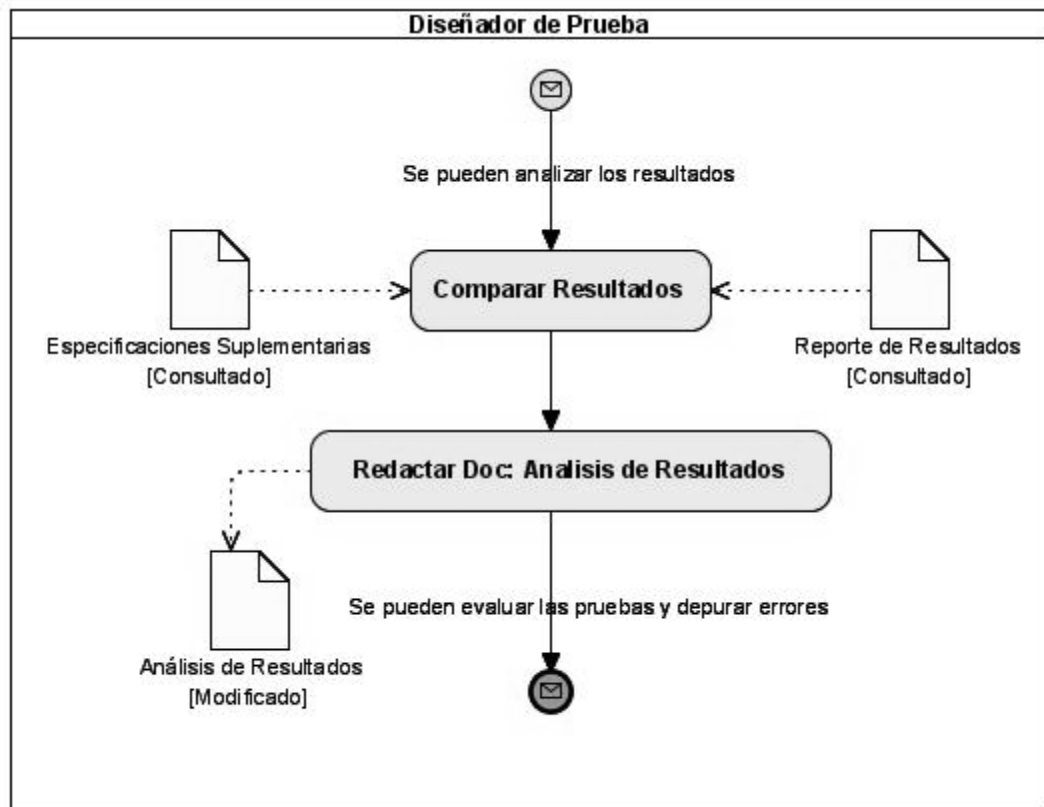
**Fig. 11. Confección de los test de Carga y Estrés**

Recibiendo la notificación por parte del jefe de modulo listo para la grabación, se procede a la navegación de la web y la grabación de los escenarios. El analista de prueba elimina los elementos que se graban como resultado de las respuestas del servidor y no son necesarias para la prueba y se agregan otros elementos de la herramienta. Se prueba con un mínimo de usuarios la funcionalidad del test hasta que esté listo para ejecutarse en las pruebas.



**Fig. 12. Ejecución de las pruebas**

La ejecución de los test inicia cuando el probador recibe una notificación para la realización de la prueba. Este comprueba que se encuentra la aplicación disponible y que ninguno de los elementos del entorno de pruebas fue cambiado o no se encuentra disponible. Si existe algún problema que dificulte la ejecución de las pruebas este informa al analista de pruebas y suspende esta actividad hasta recibir una nueva notificación. Si por el contrario están creadas todas las condiciones para ejecutar las pruebas se procede a esta actividad y se recogen los resultados obtenidos de la herramienta enviándolos al diseñador de pruebas junto a una notificación que puede analizar los mismos.



**Fig. 13. Análisis de los resultados**

El análisis de los resultados no es más que la comparación de los resultados obtenidos en las pruebas con los resultados esperados especificados en el documento de especificaciones Suplementarias. Por tanto el documento que realiza el diseñador de prueba al realizar este análisis tendrán las características de los resultados de acuerdo al los criterios: critico, severo, advertencia y cosmético, además de las observaciones de las pruebas. Este documento se envía al jefe de módulo y al evaluador de la prueba.

Los procesos de evaluación y depuración de errores no se describen en este proceso ya que estos se encuentran implementados en otras investigaciones de los cuales son objetivos.

## **2.2. Etapa de Planificación**

Para el desarrollo de la etapa de planificación de las pruebas se han tenido en cuenta aspectos descritos en el Plan de Pruebas del proyecto CICPC[2].

### **2.2.1. Plan de Pruebas**

En el plan de pruebas se definen una serie de tareas a desarrollar en las cuales se asegura que se cumpla que:

1. Se han identificado las fases de la prueba.
2. Se han establecido los requisitos de validación de los requerimientos.
3. Se han comprobado pronto las funciones importantes.
4. Es consistente el plan de pruebas con el plan del proyecto.
5. Han quedado definido los tiempos de prueba en el plan de pruebas.
6. Están disponibles los recursos necesarios para las pruebas.
7. Se encuentra establecido un mecanismo para recoger los resultados de las pruebas.
8. Se encuentra definido un mecanismo para el análisis de los resultados de las pruebas.

### **Propósito**

El propósito es dejar de forma explícita y organizada la documentación del alcance, enfoque y recursos requeridos para la realización de las pruebas. Además de definir responsables en cada una de las tareas que se deben realizar para llevar a cabo este proceso.

### **Alcance**

Preferiblemente este documento debe ser desarrollado por el mayor exponente del equipo de calidad o Gestor de pruebas, con los conocimientos requeridos para definir cuáles son los tipos de pruebas que se aplicaran al software. Para la realización de esta tarea se debe consultar el documento *Especificaciones suplementarias* en el cual aparecen los requerimientos no funcionales aplicables al sistema. El plan de pruebas debe ser actualizado para cada una de las iteraciones de prueba, definiendo en cada una de las etapas los riesgos potenciales y los recursos necesarios para llevar a cabo cada una de ellas.

## **2.2.2. Organización del Equipo de Pruebas**

El equipo de pruebas debe estar compuesto por:

**Diseñador de prueba:** Planifica, diseña y evalúa los resultados de la prueba. Es el responsable de la retroalimentación con el equipo de desarrolladores de los resultados de las pruebas y la planificación de las iteraciones necesarias para el completamiento y eficacia del proceso.

**Analista de Prueba:** Es el responsable de la elaboración de los componentes de prueba para los procedimientos que puedan ser automatizados. Está implicado en la realización de los test en la herramienta para la automatización de las mismas.

**Probadores:** Ejecutan los scripts y registran los resultados de las pruebas.

**Evaluadores:** Expertos en el proceso de pruebas que utilizan factores numéricos para exponer un resultado positivo o negativo de la ejecución de las pruebas y el trabajo del equipo de pruebas. Para la realización de esta tarea se aplican métricas específicamente diseñadas para calcular todos los

aspectos necesarios para exponer el valor de los resultados y expresar criterios de la evaluación de pruebas.

**Jefe de módulo:** Representante del equipo de desarrollo.

### **2.2.3. Estrategia de pruebas de Carga y Estrés realizadas en JMeter**

Dentro del documento *Plan de Pruebas* es importante dejar detallado una estrategia de pruebas en particular para cada tipo de pruebas que se vaya a desarrollar.

#### **Objetivo**

El objetivo general está orientado a verificar el cumplimiento de la implementación de los requerimientos. Se recomienda consultar detenidamente el documento en el que están especificados los requisitos suplementarios para constatar los importantes en este proceso.

#### **Técnica**

Para la elaboración y construcción de los casos de prueba se debe tener una información minuciosa de todos los casos de uso implicados en la prueba. Al ser desarrolladas las pruebas en la herramienta JMeter los casos de pruebas serán definidos dentro de un Plan de pruebas dentro de la aplicación del proyecto es necesario tener el nombre del servidor o IP y el puerto, la dirección URL de las páginas, el protocolo que se estará utilizando y el método que se utilizará para la petición HTTP.

#### **Entorno de Prueba**

Utilizando JMeter es importante especificar los requerimientos funcionales y no funcionales de dicha aplicación, y bajo qué circunstancias la aplicación se ejecuta incluyendo la cantidad de personal asociada a la misma. Si las pruebas serán ejecutadas en varias PC, especificar el servidor de la aplicación y las características de la misma para que esta se pueda ejecutar a modo servidor.

#### **Casos de Prueba**

Los casos de prueba deben definirse de acuerdo con la prueba que se vaya a realizar en la herramienta. Es importante definir cuáles serán las páginas críticas que se necesitan probar, el camino lógico a seguir en cada uno de los módulos del proyecto. Para la elaboración de los mismos se puede usar el modo grabación de la aplicación. Hacer una lista detallada o una referencia a los casos reales de prueba que serán utilizados para poner en práctica el plan. Detallar los casos de usos o páginas críticas en las que es importante comprobar el comportamiento de la aplicación para situaciones detalladas en los requerimientos como importantes y significativas en general. Describir todos los escenarios posibles por los cuales debe transitar el usuario para realizar determinada acción.

#### **Proceso**

El proceso de la ejecución de las pruebas depende de la forma en que vayan a ser ejecutadas las pruebas. JMeter muestra opciones para realizar pruebas automatizadas de manera planificada con un

tiempo de arranque y de parada de las pruebas. Mientras que se puede hacer de manera instantánea utilizando la opción Arrancar o Ctrl+R y Arrancar Remoto. Para cada uno las pruebas se ejecutaran haciendo peticiones HTTP y realizando una comparación con código previamente insertado en la aplicación de prueba y las respuestas de los códigos de las páginas. Como resultado final se obtiene un listado de tiempos con los cuales se formulan gráficos, si se desea obtener una imagen que exprese el análisis de los resultados. Estas pruebas pueden ejecutarse cuantas veces se quiera y sus resultados son visibles en pocos minutos.

### 2.2.4. Cronograma

En el cronograma de las pruebas de carga y estrés se planifican las fechas de las actividades que se realizan en cada una de ellas, sin afectar el tiempo en el que aparecen programadas en el *Cronograma General del Plan de Pruebas del Proyecto*[17] para ello debe hacerse una estimación de tiempo por cada tarea que debe ejecutarse. Este cronograma puede ser muy realista si los métodos que se utilizan son optimistas y cimentados en experiencias anteriores. En este se debe especificar cada una de las tareas o subprocesos que se realizarán para alcanzar la ejecución de las pruebas y el tiempo asignado a cada una de ellas así como los responsables en cada tarea. Con la realización de este pequeño documento es mucho más fácil agregar las tareas al *Cronograma General del Plan de Pruebas*. La elaboración de del cronograma inicial es muy importante para la obtención de una estimación de tiempo.

El mismo tendrá el siguiente formato, los datos adjuntos son un ejemplo:

**Tabla 2 Cronograma**

Tarea	Tiempo de desarrollo (Días)	Responsable
[Tarea o subproceso a realizar. Deben ponerse en el orden en el que van a ser ejecutadas por concepto de organización.]	[Cantidad de días necesarios para la realización de la tarea]	[Rol responsable de la tarea. Nombre y apellidos de la persona]
<b>Tiempo total</b>	[Suma total de los días necesarios para las pruebas]	

Ejemplo de un cronograma para la estimación de tiempo en el desarrollo de la automatización de las pruebas:

**Tabla 3 Ejemplo de Cronograma**

Tarea	Fecha (Inicio-Fin)	Tiempo de desarrollo (Días)	Responsable
Elaboración del Plan de Pruebas	14/05/2008 - 18/05/2008	30	Diseñador de Pruebas o Administrador de Pruebas.
Realizar los scripts de prueba	15/05/2008-20/05/2008	20	Analista de Pruebas
Actualizar Plan de Prueba	17/05/2008	1	Diseñador de Pruebas
Enviar notificación de realización de las pruebas	17/05/2008	1	Analista de Pruebas
Ejecución de las pruebas	21/05/2008 - 22/05/2008	2	Probador
Análisis y evaluación de los resultados	23/05/2008 - 24/05/2008	2	Evaluador
<b>Total</b>		57	

**Criterios de Término**

Existen varios criterios que la aplicación de automatización de pruebas brinda al usuario. Se debe hacer una comparación de los resultados obtenidos con los resultados esperados, formuladas en las especificaciones suplementarias del proyecto y luego someter esos resultados de ser necesario al criterio de expertos o usar métricas que definan de manera cuántica el criterio general de las respuestas de las pruebas.

**Herramientas**

Especificar la versión del JMeter que ha de ser utilizado, herramientas auxiliares y sus versiones.

### 2.2.5. Recursos Requeridos

En el proceso de planeación, elaboración y ejecución de las pruebas automatizadas se ven implicados de manera directa el diseñador de prueba, el analista de pruebas y el probador. El equipo de pruebas debe ser asistido por algún arquitecto, preferiblemente el arquitecto principal, el cual tiene el dominio total de las rutas críticas de la aplicación y el comportamiento de la misma.

Las pruebas automatizadas necesitan al menos una persona para la realización del Plan de Pruebas en el que queda plasmada la estrategia de prueba. Como mínimo una persona será responsable de la elaboración de los scripts de pruebas, pudiendo ser realizado por módulos independientes por más de un analista de pruebas. Al ejecutar las pruebas en la herramienta pueden ser asistidas como mínimo por una persona que utilizará los scripts de pruebas y almacenará el resultado de cada uno de ellos. Es necesaria la implicación del administrador de sistema para la colocación de la herramienta en las computadoras de los analistas de prueba y en las que se llevarán a cabo las pruebas. Estas actividades se pueden realizar en una misma computadora, pero se pueden usar más de dos si se utiliza una estación de JMeter como servidor para alguna ocasión en que se quieran simular los usuarios desde diferentes direcciones IP.

Para la ejecución de los scripts de prueba debe encontrarse disponible el software al que se le aplicaran las pruebas en condiciones equivalentes a las que será implantado.

El tiempo en el que deben estar listos estos recursos debe ser el mínimo, ya que la mayor inversión de tiempo se encuentra en el planeamiento y desarrollo de los scripts y la ejecución de las pruebas es muy breve.

El documento de *Especificación de Recursos* debe ser elaborado por el diseñador de prueba o por el equipo de pruebas en conjunto. Este documento se debe confeccionar antes que el equipo de pruebas pueda ejecutar las pruebas y estén elaborados los scripts. El mismo será entregado al administrador de sistema que será el encargado del acondicionamiento del entorno de las pruebas, supervisado por el analista de pruebas y al terminar esta tarea se debe entregar un *Reporte de Estado de las PC* en el que describe el porcentaje de cumplimiento de su tarea.

La especificación de los recursos define las tareas y los recursos necesarios para la realización de las mismas, la cantidad de personas necesarias para llevarlas a cabo y la fecha límite para el establecimiento del entorno de esos recursos. Se recomienda utilizar el siguiente formato para la plantilla de *Especificación de Recursos*.



**Tabla 4 Especificación de Recursos**

Tarea	Recurso	Cantidad de personas	Fecha Límite
[Tarea o subproceso a realizar]	[Maquinas computadores, papel de oficina, impresora, software necesarios para la ejecución de las pruebas]	[Para cada tarea especificar la cantidad de personas que intervendrán en su realización]	[Fecha para la cual deben estar creado el ambiente para trabajar en la elaboración y ejecución de las pruebas, debe estar definidas cantidad de personas en cada una de las tareas o actividades]

**Tabla 5 Ejemplo de Especificación de Recursos**

Tarea	Recurso	Cantidad de personas	Fecha Límite
Realizar los scripts de prueba	PC: 1, 512 Mb RAM o superior. Software: Word Office, JMeter, Java Virtual Machine 1.3 o superior.	1	30/04/2008
Ejecución de las pruebas	PC: 4, 512 Mb RAM o superior. Software: Word Office, JMeter, Java Virtual Machine 1.3 o superior. Navegadores: Opera, Internet Explorer, Firefox.	1	15/05/2008
Análisis y evaluación de los resultados	PC: 2, Software: Word Office	2	15/05/2008

El reporte de las PC que debe entregar el administrador de sistemas al analista de pruebas debe tener el siguiente formato, ya que el objetivo es expresar la situación de cada una de las PC solicitadas para la elaboración y ejecución de las pruebas.

**Tabla 6 Reporte de Estado de las PC**

No. PC	Descripción general del estado de la PC	Por ciento de Cumplimiento
[Número que identifica la máquina computadora que será utilizada]	[Estado de la PC, refiriéndose a capacidad libre en disco, RAM, software instalado, estado de las cuentas de los usuarios que utilizaran esas máquinas.]	Comparación entre los recursos especificados en el documento <i>Especificación de Recursos</i> y la situación actual de las maquinas computadoras.

**Tabla 7 Ejemplo de Reporte de Estado de las PC**

No. PC	Descripción general del estado de la PC	Por ciento de Cumplimiento
18 lab 207	P4, 0.99 GB de RAM. Office, JMeter y Java Virtual Machine instalados y listos para su utilización.	10/04/2008 100 % de cumplimiento de la tarea.
20 lab 207	P4, 0.99 GB de RAM. Office, JMeter y Java Virtual Machine instalados y listos para su utilización.	10/04/2008 100 % de cumplimiento de la tarea.
08 lab 208	P4, 512 MB de RAM. Office instalado, faltan algunos componentes como JMeter y Java Virtual Machine.	10/04/2008 33.33 % de cumplimiento de la tarea.

## 2.2.6. Documentación de los Resultados.

### Seguimiento y Reporte de Defectos.

Los resultados que se recolectan de las pruebas son analizados por los desarrolladores del software y el ingeniero de pruebas. De estos resultados se destacan los aspectos más importantes que pueden interferir en gran medida en el rendimiento del sistema. Tomándose dediciones por parte del equipo de desarrollo para mejorar esos resultados. Estos resultados se recolectan en el documento Análisis General de los Resultados donde se especifican los aspectos generales que influyen de alguna manera en un buen rendimiento del sistema o bien puedan interferir en los resultados del mismo en próximas versiones del producto. Además se expone un criterio por parte de los especialistas de

calidad en comparación de los resultados y las especificaciones que expresa en qué medida el sistema cumplen con lo pactado en las especificaciones suplementarias, sirviendo esto de base para el análisis realizado por el equipo de evaluación.

La base de los seguimientos está centrada en la reincidencia de los elementos encontrados en las pruebas los cuales deben ser depurados antes de la siguiente iteración de pruebas. Estos están controlados por las versiones de las ejecuciones de pruebas en cada versión del producto.

Los resultados deben ser guardados en una plantilla definida en la concepción de la planificación de las pruebas. Esta plantilla recogerá los resultados obtenidos de la ejecución de los scripts en el JMeter u alguna otra herramienta de automatización de las pruebas.

Cuando el esfuerzo de prueba esté terminado, documente los resultados y mediciones. Identifique cualquier discrepancia entre el plan y la puesta en práctica real y describa adecuadamente como aquellas discrepancias fueron manejadas.

**Formato de la *Plantilla de Resultados de las Pruebas Automatizadas*:**

Nombre del Documento: [Plantilla de Resultados de las Pruebas]

Nombre del módulo: [Nombre del modulo o sub modulo al que se le haya aplicado la prueba]

Versión: [Versión del documento, para cada iteración en el mismo módulo]

Tipo de prueba ejecutada: [Se especifica el nivel de complejidad de la prueba, simple o en profundidad, dependiendo si la prueba ha sido solo una llamada a las paginas o la transmisión de datos a ellas.]

Control de versiones [Se debe especificar la fecha en que ha sido realizada la prueba, la iteración de la prueba y los nombres y apellidos de la persona que haya ejecutado las pruebas y recopilado los resultados]

Tabla de contenidos [Breve índice del contenido de la Plantilla]

Aspectos reflejados en la prueba [Resultado final de las pruebas expuesto en las respuestas de aserciones. Los aspectos siempre tendrán algún valor por lo que se puede decir que estas pruebas no es en realidad un detector de errores si no un medidor del estado general del software]

**Descripción de los aspectos mostrados en la tabla de la Plantilla de Reporte de los Resultados**

Muestras: [Cantidad de páginas (Hilos) que simulan la cantidad de usuarios. Que están interactuando con el sistema desde la misma URL]

Media: [Media de tiempo en milisegundos en las que las páginas que se cargaron de manera satisfactoria]

Mediana: [Tiempo promedio que han tardado en cargarse las paginas]

Min: [Tiempo mínimo que ha demorado en cargarse una página]

Max: [Tiempo Máximo que ha tardado en cargarse una página]

Línea 90 %: [90 por ciento del tiempo en el que las páginas que se cargaron de manera satisfactoria]

%Error: [Por ciento de error de las paginas que no se llegaron a cargar de manera satisfactoria]

Kb/Seg: [Velocidad de carga de las paginas]

Observaciones: [En este campo se especifica cualquier eventualidad que ocurra durante la prueba o alguna recomendación]

**Tabla 8 Plantilla de Resultados de las Pruebas**

URL				
Muestras	Media	Mediana	Min	Max
Línea 90%	% Error	Rendimiento	KB/Seg	
Observaciones				

### 2.3. Elaboración de los Test

En estos momentos la etapa de decidir lo que se necesita probar y quienes lo llevarán a cabo ya está realizada, la etapa actual es la que define el cómo se llevarán a cabo esas pruebas. En esta tarea se ve implicado directamente el ingeniero de componentes, trabajador del equipo de pruebas, especializado en la utilización de la herramienta. El cual realizará los scripts de prueba.

Para describir cómo llevará a cabo las pruebas, el analista de pruebas realiza un análisis previo apoyándose en las Especificaciones de Casos de Uso y en los desarrolladores del proyecto, los cuales tienen la lógica de la utilización de las páginas.

Para ello utilizará sobre todo la lógica del mapa de navegación de cada una de los Casos de Uso. Con el objetivo de realizar esta tarea de forma organizada para al terminar los scripts y estos sean implementados, los resultados sean entendibles. Es preferible implementar las pruebas de forma tal que estos resultados queden conformados por los módulos, para que estos estén expresados de manera organizada y entendible.

Para la confección de los test se deben tener en cuenta las siguientes pautas para lograr que este logre cumplir con las expectativas de la prueba:

- Conciso: Es tan simple como sea posible y no más simple.

- Auto verificador: Basta con los informes de la prueba sus propios resultados; no necesita ninguna interpretación humana.
- Repetible: La prueba puede funcionar muchas veces sin la intervención humana.
- Robusto: Probar produce el mismo resultado ahora y por siempre. Las pruebas no son afectadas por los cambios en el ambiente externo.
- Suficiente: Las pruebas verifican todos los requisitos del software que es probado.
- Necesario: Todo en cada prueba contribuye a la especificación del comportamiento deseado.
- Claro: Cada declaración es fácil de entender.
- Eficiente: Funcionamiento de pruebas en una cantidad de tiempo razonable.
- Específico: Puntos de cada falta de la prueba apuntando a un pedazo específico de funcionalidad quebrada.
- Independiente: Cada prueba puede funcionar por sí mismo o en una habitación con un sistema arbitrario de otras pruebas en cualquier orden.
- Conservable: Las pruebas deben ser fáciles de entender, de modificarse y de extender.

## 2.4. Manual para realizar Pruebas de Carga y Estrés en JMeter

### Propósito

El manual que muestra el proceso de realización de test para la concepción y ejecución de las pruebas de sistema no funcionales de Carga y Estrés se ha realizado con el propósito de especializar estos procesos de manera que se comiencen a tomar los mismos con la seriedad y la importancia que merecen. Apoyando a su vez la mejoría y eficacia de sus resultados, eliminando los errores derivados de la intervención del factor humano al participar menos personal en la elaboración y ejecución de las mismas. Se pretende además nivelar los conocimientos de los especialistas que trabajan en este campo, mostrándoles una vía más factible para la realización de su trabajo logrando mayor éxito en sus resultados. Con esto se desea contribuir en el ahorro de recursos y de personal implicado en estos procesos favoreciendo el desarrollo del país al mostrar una mejor utilización de las Tecnologías de la Información y las Comunicaciones.

### Alcance

Este trabajo está dirigido a todos los especialistas de Calidad que trabajan en el proceso de pruebas, específicamente a los especialistas implicados en las pruebas de Carga y Estrés.

### Estructura del Manual “Cómo Realizar Pruebas de Carga y Estrés en JMeter”

El documento está compuesto por 4 capítulos:

- Capítulo I Introducción: Se realiza una breve reseña del trabajo, mostrando antecedentes y una orientación general del documento para ubicar los usuarios en cuanto al contenido.
- Capítulo II Recursos y Componentes: Este capítulo expone los requerimientos necesarios para la utilización del JMeter y los componentes de los cuales dispone la herramienta para la confección de los test.
- Capítulo III Elaboración de Escenarios de Prueba Simples: Debido a los diferentes planos en los que se puede ver una aplicación web se ha decidido trazar dos caminos en los cuales el Ingeniero de Pruebas pueda identificar la mejor opción para la comprobación de sus requisitos vinculados con los aspectos de carga y estrés. Los escenarios simples se indican en mayor medida para aquellas aplicaciones que no tienen un gran flujo de datos entre el usuario y el servidor.
- Capítulo IV Elaboración de Escenarios de Prueba para Software de Gestión: Debido al intercambio de datos entre el usuario y el servidor de la aplicación además de características especiales de seguridad, se deben utilizar elementos adicionales en la elaboración de los test por ello este capítulo expone para este tipo de aplicación los pasos a seguir para la realización de los mismos.

**Recomendaciones:**

Se pueden incorporar otros elementos que sean de gran utilidad en el proceso de pruebas, no solo el de pruebas de carga y estrés, sino también para las pruebas JDBC, o sea pruebas a la base de datos. Hay que tener en cuenta que este manual ha sido el resultado de una investigación y el trabajo en un proyecto determinado, por tanto aun puede nutrirse de nuevas experiencias y mejores prácticas. Además extender su utilización hacia otros proyectos de similares características.

**2.5. Análisis de los Resultados**

Para realizar un análisis de los resultados, se deben comparar los resultados obtenidos en la ejecución de los escenarios de prueba y los requerimientos pactados en el documento de *Especificaciones Suplementarias* del proyecto en cuestión. Es necesario, la utilización de artefacto *Resultados de las Pruebas Automatizadas*. La comparación de los índices recogidos en dicho artefacto, es utilizada como resultado final de las pruebas por el quipo de Evaluación de las Pruebas del Software los cuales exponen una calificación final mediante la utilización de métricas y métodos de evaluación.

Para realizar un eficiente análisis de los resultados es necesario además tener en cuenta el tiempo que se demora cada una de las transacciones que son en realidad las que vera el usuario. Analizar por qué unas demoran más que otras y si estos tiempos pueden ser mejorados. En caso de fallar alguna de las

peticiones analizar sus causas. Verificar que las peticiones que se estén realizando al servidor sean realmente necesarias y ver que no sean redundantes.

En estos momentos en el proyecto, se encuentran definidas las características de la red y su topología, el tipo de servidor y el enlace de conexión en el que se despliega. Pero aun no se ha calculado el resultado real del comportamiento de la aplicación bajo estas características. Todos estos aspectos relacionados con el ambiente donde se encuentra el servidor y las características de red, influyen directamente en los resultados de las pruebas de carga y estrés, debido a que el comportamiento de la aplicación está estrechamente vinculado a los cambios de estas especificaciones y en muchos de estos casos las respuestas que se esperan no están definidas para un tipo de condiciones en las cuales se debe desempeñar el software. Por tanto el análisis de los resultados está estrechamente vinculado al comportamiento esperado de la aplicación con estas especificaciones.

## **Conclusiones del Capítulo**

En este capítulo se abordó el proceso propuesto para la realización de pruebas de carga y estrés con la mayor calidad posible. Se abordan las actividades que se realizan dentro del proceso, los artefactos que se generan e intervienen en las mismas, responsables y cronogramas. Los ejemplos que se mencionaron sirven como una guía para la aplicación de esta nueva práctica.

Si desde la concepción del proyecto se definen los responsables, herramientas, actividades y artefactos implicados en el proceso, se garantiza la organización del entorno donde se desarrollarán las pruebas. Si se especifica el tiempo necesario para cada una de las acciones de este proceso y los responsables de cada una de ellas se puede decir que el equipo está preparado para iniciar las pruebas y que el éxito de las pruebas dependerá solo del desempeño profesional de quienes intervienen en ellas.

# Capítulo 3

## Pruebas de Carga y Estrés en el proyecto CICPC

### Introducción

Como se ha demostrado a lo largo de todo este documento este trabajo ha surgido por la necesidad de verificar los requerimientos pactados con el cliente durante la concepción del proyecto de modernización del CICPC desarrollado en la facultad 8. Por tanto los principales esfuerzos han sido orientados a una mayor organización y calidad en los procesos tan necesarios como las pruebas de carga y estrés, en los productos realizados por el equipo de trabajo del proyecto CICPC.

Este capítulo describe por tanto cómo se ha aplicado el proceso y la utilización de la herramienta JMeter y cómo se han analizado los resultados obtenidos por los mismos. Se realiza una breve reflexión que indica cuánto ha aportado la utilización de una herramienta que automatiza las pruebas de carga y estrés brindando resultados confiables y mayor eficiencia al proceso.

### 3.1 Planificación y Concepción de las Pruebas de Carga y Estrés del SIIPOL

#### 3.1.1 Estrategia de prueba

##### Objetivo

Es objetivo de estas pruebas es obtener un índice de resultados de comportamiento del sistema en determinadas condiciones, o sea, al realizar las pruebas de estrés y carga a la aplicación se obtienen resultados de un comportamiento que puede ser muy cercano a un comportamiento real, con ello se espera realizar un análisis en el equipo de desarrollo para en caso de ser resultados poco favorables realizar acciones que mitiguen las deficiencias. Teniendo estos resultados se puede prever errores futuros y respuestas del sistema ante cantidades específicas de usuarios y ante altos niveles de concurrencia y grandes volúmenes de información.

##### Técnica

La técnica utilizada es la técnica de caja negra ya que, como se define en RUP[4]. Por tanto por ser una prueba que necesita la entrada de datos y la validación de ellos contra resultados esperados es que se define que la técnica utilizada para este proceso de pruebas es de caja negra. Para la realización de estas pruebas se utiliza la herramienta de automatización JMeter, la cual ha sido



seleccionada como candidata para realizar estas por las características que ella presenta y por las opciones que brinda para su manipulación.

### Entorno de Prueba

La preparación del entorno de prueba es responsabilidad del analista de pruebas, quien debe garantizar que el administrador de sistema y el arquitecto principal realicen las actividades necesarias para la ambientación del entorno de las pruebas.

Para la grabación de los escenarios de prueba se necesitan los siguientes datos recogidos en el artefacto *Especificación de Recursos*. Acondicionar los recursos tanto de software como hardware es responsabilidad del administrador del sistema y del arquitecto principal.

**Tabla 9. Especificación de recursos para las pruebas de Carga y Estrés del SIIPOL**

Tarea	Recurso	Cantidad de personas	Fecha límite
Elaboración del plan de pruebas	1PC. M Office.	1	14/05/2008
Realizar los test en la herramienta	1PC. 0,99mb de Ram. M Office. Java Virtual Machine 1.3 o superior. JMeter. Navegador IExplorer.	1	15/05/2008
Ejecución de las pruebas	1PC. 0,99mb de Ram. M Office. Java Virtual Machine 1.3 o superior. JMeter. Navegador IExplorer. Se necesita además que se encuentre corriendo el servidor de la aplicación.	1	21/05/2008
Análisis de los resultados	1PC. M Office.	3	23/05/2008

La respuesta que se ha obtenido por parte del administrador de sistema es la siguiente, la cual se encuentra recogida en el artefacto *Reporte de estado de las PC*:

**Tabla 10. Reporte de estado de las PC**

No. PC	Descripción general del estado de la PC	Porcentaje de Cumplimiento
PC 27 lab 207.	1PC. 0,99mb de Ram. M Office. Java Virtual Machine 1.3 o superior. JMeter. Navegador IExplorer.	100%

### Proceso

El proceso de automatización de las pruebas consta de 4 etapas fundamentales.

1. Grabación de los escenarios de prueba.
2. Confección de los test.
3. Ejecución de las pruebas y recolección de los resultados.
4. Análisis de los resultados.

### Grabación de los Escenarios de Prueba.

La grabación de los escenarios de prueba se conformó por 5 grabaciones independientes en las cuales se solicitó un representante de cada equipo de desarrollo con conocimientos sobre las páginas críticas del módulo. Esta grabación se realizó por parte del analista de pruebas utilizando el JMeter en modo servidor utilizando un puerto para la aplicación indicándole además que reunir cada uno de los grupos de peticiones en controladores por separado y luego presionando arrancar.

Para ello se seleccionó en las opciones de internet del navegador y se especificó la dirección y el nuevo puerto que a utilizar para la grabación de los escenarios, apuntando al puerto del servidor de la aplicación JMeter.

Teniendo configurado el navegador se accedió a la dirección del servidor web al que se le realizan las pruebas y se navegó a través del módulo.

El representante del módulo grabado debe haber tenido en cuenta las siguientes situaciones para identificar las áreas importantes para este proceso:

- Tareas o panoramas altamente redundantes. Manejadoras importantes flujos de datos.
- Tareas repetidoras que pueden causar fallas del sistema por parte del personal que lo utiliza.

- Casos o panoramas bien desarrollados y bien entendidos que se identifiquen como prioridad en el uso del software.
- Las áreas relativamente estables que a causas de un excedente puedan traer problemas a la aplicación.

Luego de haber terminado esta grabación quedaron recogidas todas las peticiones que se realizan en cada uno de los módulos para llevar a cabo las acciones que un usuario ejecuta normalmente en una situación real.

### **Confección de los test.**

Posterior a las grabaciones se añadieron elementos significativos en la herramienta como son los listeners, las aserciones, la cantidad de hilos, se organizaron las peticiones en diferentes controladores y se especificaron la cantidad de veces que se necesita la ejecución de estas peticiones durante la prueba. Este es el momento en el que el analista de pruebas expuso con la ayuda de la herramienta de prueba una modelación de un escenario real de actividad en el software que se prueba. Para ello debe haber realizado previamente un estudio sobre las áreas que son objeto de las pruebas y el comportamiento de los usuarios en las mismas.

Las los test que se confeccionan para probar el SIIPOL deben tener la siguiente estructura jerárquica en los elementos ya que según la clasificación en cuanto al tipo de test se define como complejo:

- Grupo de Hilos
- Valores por defecto para Petición HTTP
- Gestor de Cookies HTTP
- Modificador de Re-Escritura HTTP
- Aserción de respuesta
- Controlador Only Once
  - Extractor de Expresiones Regulares
  - Petición HTTP
- Controlador Loop
  - Controlador Simple
    - Petición HTTP
- Controlador Only Once
  - Controlador Simple
    - Petición HTTP
- Ver Árbol de Resultados

- Informe Agregado

### **Descripción del Manejo de los Elementos**

Debido a la utilización de ACEGI para la seguridad del SIIPOL se trazó una estrategia en la cual se le otorgan permisos únicos a un determinado usuario con el que se identifica la aplicación JMeter ante la aplicación web. Por tanto en el elemento Grupo de Hilos se especificó un único usuario tratado por el arquitecto de seguridad, ya que la aplicación no permite autenticarse más de un interesado a la vez con la misma contraseña y usuario. Por tanto se separó la petición inicial HTTP donde la herramienta JMeter se autentica ante la aplicación web en un controlador Only Once, realizándose esta petición una única vez.

Para potenciar la carga al sistema se utilizó en este caso un controlador Loop, especificando en el mismo la cantidad de veces que se necesitan realizar las peticiones al servidor web. Dentro de esta se anidan las peticiones HTTP, cada una de ellas organizadas en el controlador simple con el cual fueron grabadas y serán tantas como sean necesarias para realizar acciones concretas del sistema.

Para finalizar la prueba y no crear conflictos de usuario en el sistema se utilizó un controlador Only Once que contiene en su interior los controladores simples que agrupan las peticiones referentes al cierre de sesión.

Los elementos que se encuentran al inicio del test como son: Valores por defecto para Petición HTTP, Gestor de Cookies HTTP, Modificador de Re-Escritura HTTP. Se utilizan en la gestión de la información que se envían entre las peticiones de la herramienta de pruebas y el sistema. Los valores por defectos para las peticiones HTTP que se definieron son:

- Dirección IP o nombre del servidor donde se encuentra el SIIPOL
- Tipo de protocolo que se utiliza
- Puerto por el que se accede

El Gestor de Cookies estuvo apuntando al tipo de política compatible y en el modificador de re-escritura se le pasaron los parámetros expuestos en el manual *Cómo Realizar Pruebas de Carga y Estrés en JMeter*[18].

Teniendo estos elementos se adicionó la Aserción de Respuesta y se especifica en el patrón aprobar que la respuesta obtenida del servidor web de cada una de las peticiones de la herramienta de prueba coincida con el código 200 que es el código que se recibe del servidor cuando la petición ha sido respondida correctamente.

Para terminar la conformación del test se agregaron los listeners, en este caso se incorporó un Informe Agregado y un Ver Árbol de Respuestas, con el objetivo de cubrir todos los resultados de la prueba y

monitorear los errores que se puedan encontrar. Una vez probado todos los elementos y verificado su comportamiento se guardó el plan de pruebas y este se encontró listo para su ejecución.

### **Ejecución de las pruebas y recolección de los resultados.**

Para la ejecución de la prueba se comprobó el funcionamiento de la versión del SIIPOL a la que se le realizó la prueba. Además esta se encontró en situaciones reales ya que el rendimiento de una aplicación y los tiempos de respuestas de las peticiones están definidos por la capacidad del servidor donde se encuentra instalada la aplicación, el tipo de enlace, el tipo de conexión y de red. Estas especificaciones quedaron reflejadas en el resultado de las pruebas y son importantes a tener en cuenta.

La prueba se especificó para simular entre 450 y 900 usuarios debido a que es lo que se requiere en esta primera versión del SIIPOL. En futuras versiones deben realizarse estas pruebas simulando entre 5000 y 7000 usuarios accediendo a la aplicación.

### **Casos de Prueba**

La prueba tubo por cada iteración dos resultados que se registraron por cada uno de los módulos o equipos generándose 12 resultados de prueba para cada iteración de pruebas que se realizó. Los resultados fueron registrados teniendo en cuenta los siguientes módulos:

- Análisis de la Información
- Gestión Administrativa
- Investigación Penal
- Investigación Criminalística
- Investigación Forense
- SIIPOL

En general los casos de prueba estarán definidos dentro de las grabaciones de cada uno de los módulos y el SIIPOL de manera general.

### **Criterios de Término**

Al terminar esta investigación quedaron pendientes los criterios que se deben tener en cuenta para situaciones en las que se decida terminar los ciclos de pruebas están encaminados al rendimiento del sistema debido a que no se ha realizado una nueva versión del software. Por tanto debe tenerse en cuenta para la nueva iteración de pruebas los siguientes aspectos:

1. ¿Quedaron eliminadas las peticiones redundantes e innecesarias?
2. ¿Quedaron analizadas las peticiones que suponen mayor carga para el sistema y se han mejorado hasta su estado más simple?

## Herramientas

Para la realización de las pruebas automatizadas se utilizó la herramienta JMeter versión 2.3.

## Recursos Requeridos

Los roles implicados en los subprocesos y tareas de las pruebas de carga y estrés en el SIIPOL se definieron de la siguiente manera:

- Realización de la estrategia de pruebas (*Cronograma, Listado de recursos, método de ejecución*): Analista de Prueba.
- Análisis de escenarios de uso del sistema que deben probarse, por módulos: Jefes de módulo.
- Grabación de los escenarios de carga y concurrencia. Ingeniero de componentes de prueba.
- Concepción de los test: Ingeniero de componentes de prueba.
- Ejecución de las pruebas: Ingeniero de componentes de prueba.
- Recolección de los resultados: Ingeniero de componentes ó probador.
- Análisis de los resultados de las pruebas: Analista de prueba.
- Depuración de errores: Jefe de módulo.
- Evaluación de las pruebas: Equipo de evaluación.

## Calendario y Plazos

En el cronograma asociado a estas pruebas se definen las tareas que se realizaron acompañadas de sus responsables con fecha en la que deben ser ejecutadas. Las tareas a realizar pactadas en el cronograma se encontraron dentro del marco del cronograma general del proyecto, para no interferir en otras tareas del proyecto que se estuvieran realizando concurrentemente. Para tener un mayor control de las tareas que se realizaron se especificaron en el mismo artefacto Cronograma un cronograma donde se define el horario y la fecha en la que se cumplieron las grabaciones a cada módulo. Este artefacto ha sido realizado según la muestra presentada en el Capítulo II. Estas se ven reflejadas en las tablas que se muestran a continuación.

Tabla 11. Cronograma del Proceso de Pruebas Automatizadas del SIIPOL

Inicio	Fecha (Inicio-Fin)	Tiempo de desarrollo (Días)	Responsable
Elaboración de Grabación de las pruebas	22/05/2008	1	Analista de Prueba
Realizar los test en la herramienta	23/05/2008	1	Analista de Prueba
Ejecución de las pruebas	23/05/2008	1	Probador
Análisis de los resultados	24/05/2008	1	Diseñador de Prueba
Evaluación de las pruebas	25/05/2008	1	Evaluadores
<b>Total (Días)</b>		<b>5</b>	

Tabla 12. Cronograma de grabación de escenarios a probar

Módulos a Grabar	Fecha	Hora
Análisis de Información	22/05/2008	8:30 AM
Investigación Penal	22/05/2008	2:00 PM
Investigación Criminalística	22/05/2008	3:00PM
Investigación Forense	22/05/2008	5:00 PM
Gestión Administrativa	22/05/2008	8:30 PM

### Definición de los Entregables.

El proceso pruebas de carga y estrés automáticas generó un total de 5 artefactos, los mismos se encuentran en el repositorio del proyecto.

### 3.1.2 Realización de los Test

Para el proceso de grabación de los escenarios de carga y concurrencia de cada uno de los módulos se contó con un total de 5 grabaciones realizándose de manera satisfactoria en todo caso.

Para realizar estas grabaciones fue necesario contar con la existencia de una versión estable del software. Se obtuvieron los siguientes resultados durante la grabación:

- El módulo Investigación Penal con un total de 322 peticiones anidadas en 35 controladores lógicos.
- El módulo Investigación Forense con un total de 243 peticiones anidadas en 23 controladores lógicos.
- El módulo Investigación Criminalística con un total de 379 peticiones anidadas en 39 controladores lógicos.
- El módulo Análisis de la Información con un total de 456 peticiones anidadas en 35 controladores lógicos.
- El módulo Gestión Administrativa con un total de 160 peticiones anidadas en 17 controladores lógicos.
- Un total de 1560 peticiones en todo el sistema.

Teniendo este resultado de las grabaciones se prosigue a la elaboración de los Test. Para ello se incorporaron un total de 155 Informes Agregados, uno por cada controlador que se utilizó para monitorear los resultados de las peticiones de manera más directa además para ello se contó con un elemento Árbol de Resultados.

Para realizar la prueba de concurrencia se especificó en el grupo de hilos un total de 450 usuarios como usuarios medios y para 900 usuarios como sobrecarga del sistema. Mientras que para realizar las pruebas de carga se especificaron para cada usuario una cantidad de repeticiones en cada controlador Loop que contiene las peticiones de envío y respuestas de datos al sistema. Así quedó conformado el plan automatizado de prueba.

### **3.1.3 Ejecución de las Pruebas**

Para la ejecución satisfactoria de las pruebas automatizadas se realizaron las tareas definidas en la estrategia de pruebas que definen una serie de pasos a seguir como parte del aseguramiento de la calidad del proceso.

1. Se comprobó que el SIIPOL se encontraba disponible y funcionando de manera estable.
2. Se comprobó que las condiciones en las que se realizaron las grabaciones no hayan sido cambiadas.
3. Se comprobó que la herramienta JMeter se encontraba instalada en la PC que sirve como cliente y que los requerimientos de la misma respondan a las necesidades de la herramienta,



según lo especificado en el documento de Reporte de estado de las PC entregado por el administrador de sistema.

Se ejecutaron de manera general un total de 12 pruebas, para analizar de manera independiente los resultados de los módulos y de manera general el resultado del SIIPOL como la integración de cada uno de ellos.

El tiempo invertido durante el registro de los resultados de las pruebas fue de 2 horas y 34 minutos, la ejecución de las pruebas solo 15 minutos. Como la herramienta JMeter no ofrece una vía para guardar los resultados, estos se recogen en un documento y por tanto se incurre en más tiempo que el que se necesita para ejecutar las pruebas.

Como resultado del proceso se registraron un total de 12 documentos de resultados de las pruebas, generado uno para cada módulo para cada cantidad de usuarios específicos y el SIIPOL como un todo para 450 y 900 usuarios respectivamente.

Los 12 resultados obtenidos se encuentran en el repositorio del proyecto[19]. Un ejemplo de ellos se puede ver en el anexo 4.

### 3.1.4 Análisis de los resultados de las Pruebas

Uniendo todas las partes que conforman el SIIPOL que se graban como áreas críticas se cuenta con un total de 70417 peticiones realizadas al servidor para 450 usuarios y 140834 para 900. Respondidas en un promedio de 16 milisegundos cada una, además la herramienta mostró con este tiempo que por cada usuario la aplicación puede gestionar 43 y 42 peticiones respectivamente para cada cantidad de usuarios lo cual supera la cantidad de peticiones necesarias para realizar la transacción más pesada que consta de 40 peticiones. Teniendo en cuenta que estos resultados se han recogidos en condiciones de velocidad de red de 100 MBps y las características del hardware servidor reflejadas en el anexo 5 y como software se utilizó: Apache Tomcat/5.5.20 con una memoria máxima de 1820.50 MB y un máximo de hilos soportados concurrentemente de 450. Este resultado demuestra estar por debajo de los resultados históricos reflejados en los antecedentes del Proyecto Técnico. Sin embargo analizando que las condiciones de red en los que estos se reportaron son 10 veces más rápidas que las que se contaba para la prueba se concluye que el resultado ha sido exitoso y más aun cuando se verifica en el documento de Especificaciones Suplementarias se llega a la conclusión que el software cumple con lo pactado con el cliente ya que este solo especificó que las peticiones tuvieran un tiempo de respuesta menor que un segundo.

Por tanto este software se califica según a las características de los Resultados de las pruebas automatizadas de carga y estrés como:

- **Advertencia:** Ya que los tiempos de respuestas de las aplicación son inferiores a los

tiempos de respuestas esperados además de los señalamientos por utilizaciones innecesarias de la petición estilo/estyle.ccs. No obstante hay que señalar que estos tiempos mejorarán en dependencia a las condiciones de red y ancho de banda con las que se cuenta.

Los principales señalamientos de las pruebas:

Repetición constante de la petición estilos/estyle.css.

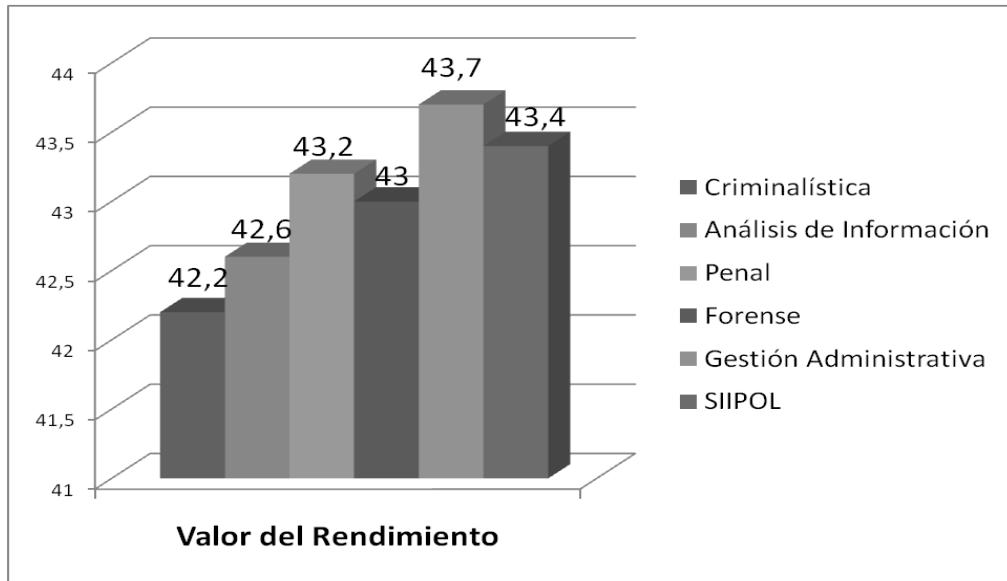
Esta petición tiene un pequeño tiempo de respuesta por lo que en estos momentos no afecta los resultados del rendimiento, pero deben controlarse la utilización de las mismas ya que para futuras versiones del software que incluya otras funcionalidades puede interferir en la velocidad y por tanto en el rendimiento de la aplicación.

Los resultados que se obtuvieron para 450 usuarios fueron los siguientes:

**Tabla 13 Análisis de los Resultados del SIIPOL Para 450 usuarios**

Datos	Inv. Forense	Inv. Penal	Inv. Criminalística	Análisis de la Información	Gestión Administrativa	SIIPOL
<b>Muestras</b>	11635	13624	17573	16016	11567	70417
<b>Media</b>	16	17	21	16	15	15
<b>Mediana</b>	16	16	16	16	16	16
<b>Min</b>	0	0	0	0	0	0
<b>Max</b>	1000	9172	1672	16	1000	9172
<b>Línea 90 %</b>	16	16	31	16	16	16
<b>%Error</b>	0.0	0.0	0	0.0	0.0	0.0
<b>Rendimiento</b>	43.0/seg	43.2/seg	42.2/seg	42.6/seg	43.7/seg	43.4/seg
<b>Kb/seg</b>	233122.4	301162,6	462543.8	418690.9	182137.4	1326612.1

Para 450 usuarios los resultados quedaron graficados de la siguiente forma:

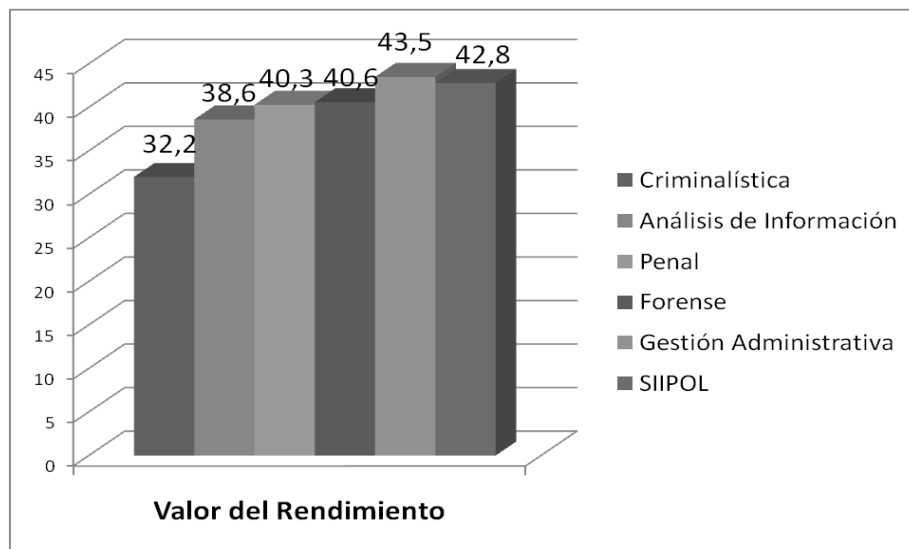


**Fig. 14 Análisis del rendimiento para 450 usuarios**

El comportamiento del rendimiento del sistema, que se muestra en la figura anterior se corresponde a los datos de los siguientes módulos de izquierda a derecha:

Criminalística, Análisis de la información, Penal, Forense, Gestión Administrativa, SIIPOL.

Para 900 usuarios los resultados quedaron graficados de la siguiente manera:



**Fig. 15. Análisis del rendimiento del sistema para 900 usuarios**

Orden de aparición de izquierda a derecha de los módulos:

Criminalística, Análisis de Información, Penal, Forense, Gestión Administrativa y el sistema SIIPOL. Ya que en la escala de grises no se infiere a cuál de ellos pertenece el valor.

Los resultados que se obtuvieron para 900 usuarios fueron los siguientes:

Tabla 14 Análisis de los Resultados del SIIPOL para 900 usuarios

Datos	Inv. Forense	Inv. Penal	Inv. Criminalística	Análisis de la Información	Gestión Administrativa	SIIPOL
<b>Muestras</b>	23270	27248	35146	32032	23134	140834
<b>Media</b>	16	16	14	16	16	15
<b>Mediana</b>	18	18	16	15	15	16
<b>Min</b>	0	0	0	0	0	0
<b>Max</b>	1000	8172	1359	1125	1000	8172
<b>Línea 90 %</b>	18	16	16	15	31	16
<b>%Error</b>	0.0	0.0	0.0	0.0	0.0	0.0
<b>Rendimiento</b>	40.3/seg	40,6/seg	32.2/seg	38.2/seg	43.5/seg	42.8seg
<b>Kb/seg</b>	466244.8	60235.2	925087.6	837381.8	364274.8	265322.2

## 3.2. Planificación y concepción de las pruebas del Portal Web

### 3.2.1. Estrategia de prueba

#### Objetivo

Las pruebas de carga y estrés al Portal Web del proyecto CICPC se realizan con el objetivo de descubrir ausencias de respuestas del servidor y medir los tiempos de respuestas del servidor de la aplicación, para encontrar en caso de encontrar resultados negativos, los aspectos que consumen mayor parte de rendimiento del sistema y con ello trazar estrategias de programación que ayuden a mejorar esas ineficiencias.

#### Técnica

La técnica utilizada fue la técnica de caja negra. Por ser una prueba que precisa la entrada de datos y la validación de ellos contra resultados esperados es que se define que la técnica utilizada para este proceso de pruebas es de caja negra. Para este tipo de pruebas se trazó una estrategia en la que convergen las páginas que reportan mayor cantidad de usuarios concurrentes en el sistema. Para realizar estas pruebas se utilizó la herramienta JMeter que brinda una serie de funcionalidades y ofrece registros de los aspectos relacionados con las peticiones que se realizan al sistema.

#### Entorno de Prueba

El entorno de prueba ha sido acondicionado por el analista de pruebas, dando respuesta a la Especificación de Recursos para las Pruebas de Carga y estrés del Portal Web. Todo ello queda reflejado en el artefacto Reporte de las PC. Estos documentos se encuentran en el repositorio del proyecto[19].

### **Proceso**

El proceso de automatización de las pruebas del Portal Web al igual que el SIIPOL constó de 4 etapas fundamentales.

1. Grabación de los escenarios de prueba.
2. Confección de los test.
3. Ejecución de las pruebas y recolección de los resultados.
4. Análisis de los resultados.

### **Grabación de los escenarios de prueba.**

La grabación de los escenarios de prueba se conformó por 2 grabaciones independientes en las cuales se solicitó un miembro del equipo de desarrollo del portal, con los conocimientos necesarios para realizar una navegación de manera satisfactoria en la aplicación. Estas grabaciones recogieron de manera general toda la navegación del sitio. Una de las grabaciones se encontró dirigida a la navegación plena y la otra grabación se realizó con el objetivo de recoger todas las peticiones que se generan al realizar acciones entre el servidor del Portal Web y el servidor Integrity del SIIPOL.

### **Confección de los test**

La confección de los test para las pruebas de cargas y estrés del Portal Web se realizó utilizando el método de pruebas sencillas, ya que las páginas que se consideraron críticas tienen un acceso de usuario sin contraseña. Por tanto la política que se sigue para la elaboración de estos test es mucho más sencilla, consiste en la navegación plena sin autenticación, ya que la mayor carga es la de usuarios estándar.

Los test que se elaboraron para probar el Portal Web del CICPC tuvieron la siguiente estructura jerárquica en los elementos:

- Grupo de Hilos
- Valores por defecto para Petición HTTP
- Gestor de Cookies HTTP
- Aserción de respuesta
- Controlador Loop
  - Controlador Simple
    - Petición HTTP

- Ver Árbol de Resultados
- Informe Agregado

### **Descripción de los elementos de prueba**

En el grupo de hilos se especificaron la cantidad de hilos que simulan la cantidad de usuarios necesarios para la prueba fueron 55. Y en el Controlador Loop se añadieron las repeticiones que fueron necesarias para las peticiones HTTP, de esta forma se intensificó la carga al servidor Web. Los valores por defectos para las peticiones HTTP fueron:

- Dirección IP o nombre del servidor donde se encuentra el Portal Web.
- Tipo de protocolo que se utiliza.
- Puerto por el que se accede.

Los demás elementos se añaden según se especifica en el manual que define cómo realizar los casos de prueba y ejecutarlos en la herramienta JMeter [18] .

### **Ejecución de las pruebas y recolección de los resultados**

Para la ejecución de las pruebas se comprobó el funcionamiento de la aplicación, la disponibilidad del servidor Web y las características necesarias para la ejecución de la herramienta de pruebas.

Se ejecutó la prueba de manera repetitiva hasta que se logró encontrar el nivel máximo de usuarios que el servidor soporta como conexiones y de esta manera se comprobó la capacidad que tiene el mismo de reponerse, Además se obtuvo un registro de los errores que muestra el sistema al denegar respuestas a las peticiones de los usuarios.

### **Casos de Prueba**

La prueba se realizó con un único caso de prueba siguiendo la estructura planteada en la confección de los test. Este test recoge toda la actividad del portal web, se realiza un único caso de prueba por la pequeña actividad que muestra la aplicación.

### **Criterios de Término**

Debido a que el sistema no se encuentra en las condiciones reales en que será implantado, la prueba se confecciona en la base de obtener una serie de resultados que permiten mejorar y detectar problemas que interfieran en el rendimiento del sistema. Estas pruebas dejarán de realizarse si se cumple que:

1. Se han eliminado peticiones redundantes e innecesarias.
2. Se han analizado las peticiones que suponen mayor carga para el sistema mejorando la transferencia de datos.
3. Los tiempos de respuestas de las peticiones son mínimos.

### **Recursos Requeridos**

Los roles implicado en los subprocesos y tareas de las pruebas de carga y estrés en el Portal Web se definen de la siguiente manera:

- Realización de la estrategia de pruebas (*Cronograma, Listado de recursos, método de ejecución*): Diseñador de Prueba.
- Análisis de escenarios de uso del sistema que deben probarse, Líder del Portal Web.
- Grabación de los escenarios de carga y concurrencia. Analista de Pruebas.
- Concepción de los test: Analista de prueba.
- Ejecución de las pruebas: Probador.
- Recolección de los resultados: Probador.
- Análisis de los resultados de las pruebas: Diseñador de prueba.
- Depuración de errores: Líder del Portal Web.
- Evaluación de las pruebas: Equipo de evaluación.

### Calendario y Plazos

A continuación se muestra el Cronograma del portal Web del CICPC:

**Tabla 15. Cronograma de actividades del Portal Web del CICPC**

Inicio	Fecha (Inicio-Fín)	Tiempo de desarrollo (min)	Responsable
Elaboración de Grabación de las pruebas	22/05/2008	30	Analista de Pruebas
Realizar los test en la herramienta	22/05/2008	30	Analista de Pruebas
Ejecución de las pruebas	22/05/2008	10	Probador
Análisis de los resultados	22/05/2008	60	Diseñador de Pruebas
<b>Total (min)</b>		130	

En el cronograma asociado a estas pruebas se definen las tareas que se realizan acompañadas de sus responsables con fecha en la que deben ser ejecutadas. Las tareas a realizar pactadas en el cronograma deben encontrarse siempre dentro del marco del cronograma general del proyecto, para no interferir en otras tareas del proyecto que se estén realizando concurrentemente.

### Definición de los Entregables

Los artefactos generados en este proceso por su gran extensión se encuentran en el repositorio del Proyecto.

### **Seguimiento y Reporte de Defectos**

Los resultados que se recolectaron durante las pruebas fueron analizados por los desarrolladores del software y el ingeniero de pruebas. De estos resultados se destacan los aspectos más importantes que pueden interferir en el rendimiento del sistema. Los resultados negativos del portal web están condicionados a las características de hardware en que se encontraba ejecutándose, por tanto se ha tomado la decisión realizar una nueva iteración de pruebas en condiciones similares a las reales.

El error de 0.09 por ciento ocasionado por la imagen que no se mostró fue corregido por parte de los desarrolladores del portal.

### **3.2.2 Realización de los Test**

La realización de los test quedó conformada por un único test compuesto por 35 controladores lógico que recogen de manera general 130 peticiones que conforman la navegación más significativa del sistema. La página de inicio muestra 55 peticiones. Cargando los elementos que sirven como plantilla al resto de las páginas de la navegación del sitio las cuales hacen una llamada de una media de 4 a 5 peticiones.

### **3.2.3. Ejecución de las Pruebas**

Las pruebas fueron realizadas al portal que presenta a las siguientes características de software:

- Servidor web: Apache 2.2.4
- Servidor de Base de Datos: mysql 5.0
- Lenguaje utilizado: php 5.2.5.

Con el siguiente hardware de servidor:

- Disco Duro: ST380817AS (80 GB, 7200 RPM, SATA)
- Memoria del sistema: 1015 MB (PC3200 DDR SDRAM)
- Procesador: Intel Pentium 4 630, 3000 MHz (15 x 200)

Para la ejecución satisfactoria de las pruebas automatizadas se realizaron las tareas definidas en la estrategia de pruebas que definen una serie de pasos a seguir como parte del aseguramiento de la calidad del proceso.

1. Se comprobó que el Portal Web se encontraba disponible y funcionando de manera estable.
2. Se comprobó que las condiciones en las que se realizaron las grabaciones no fueron cambiadas.



- Se comprobó que la herramienta JMeter se encontraba instalada en la PC que sirve como cliente y que los requerimientos de la misma respondían a las necesidades de la herramienta, según lo especificado en el documento de *Reporte de estado de las PC* entregado por el administrador de sistema.

Se ejecuta el test de pruebas teniendo como cantidad de usuarios límites 55 usuarios. Con esta cantidad de hilos se grabó un resumen de los resultados de las peticiones del test y se obtienen las condiciones máximas en las que puede funcionar de manera correcta el sistema.

Los resultados quedan expresados en el documento *Resultado de las pruebas automatizadas al Portal Web* en el anexo 4.

### 3.2.4 Análisis de los resultados de las Pruebas

La cantidad de peticiones que generó la grabación de la navegación del portal web fue de 55 peticiones mostrando los resultados que aparecen en la tabla 15. Los resultados no fueron satisfactorios debido a que este se encontraba instalado en un servidor con deficientes recursos lo cual limitaba la cantidad de conexiones y la velocidad de respuesta de las peticiones.

Analizando los resultados generales de las pruebas al Portal Web se obtiene un total de: 5335 peticiones para un total de 55 usuarios mostrando los datos que se muestran a continuación.

**Tabla 16. Resumen de Resultados de las pruebas de carga y estrés del Portal Web**

	Muestra	Media	Mediana	Línea 90 %	Min	Max	% Error	Rendimiento	Kb/seg
<b>Total</b>	5335	7010	110	18.985	0	415516	0.009	4.090/seg	80022,7

De esta prueba se destacan las siguientes observaciones a tener en cuenta por el equipo de desarrollo del portal para lograr una mayor eficiencia.

- El sistema muestra un soporte de concurrencia aproximadamente 5335 peticiones simultáneas. Esto significa que con las prestaciones del servidor en el que se encuentra instalado puede soportar un máximo de 55 usuarios concurrentes ya que con 70 no responde las peticiones. Mostrando un tiempo de recuperación de alrededor de 60 segundos restableciéndose de manera eficiente sin intervención del factor humano.
- La imagen /portalcicpc/files/img/logo.png nunca llegó a cargarse. Por lo que esa petición es fallida. Esta fue la causa principal del 0,009 % de error.
- El portal ha sido probado en condiciones inferiores a las reales en las que se debe desplegar. Para la cantidad de usuarios que soporta en estos momentos con las prestaciones

anteriormente mencionadas se considera satisfactorio el rendimiento que esta la aplicación. En general las peticiones tienen un tiempo de respuesta de 18.985 milisegundos lo cual demuestra que ha sido implementado siguiendo los requerimientos establecidos en los que se plantean buenas prácticas de programación.

## **Conclusiones del Capítulo**

Las pruebas de carga y estrés realizadas de manera automática a las aplicaciones de software del Proyecto CICPC arrojaron resultados simbólicos ya que marcan el comienzo de un historial del comportamiento de cada una de las aplicaciones. Sirvieron para demostrar la calidad de los productos y en qué medida estos cumplen con las expectativas del cliente. También son importantes ya que gracias a ellas se pudieron detectar anomalías como los errores por fallo de respuesta de una petición que incurren en el futuro en problemas para el rendimiento del sistema.

Las pruebas en general se realizaron de manera eficiente con una buena organización y todos los procedimientos quedaron reflejados en los entregables que se generaron en cada una de las etapas que conforman este trabajo. Lo que apunta favorablemente a la definición del proceso como una buena práctica de concepción, elaboración y ejecución de las pruebas de carga y estrés de manera automatizada.

Los señalamientos realizados en el documento de análisis son objeto de discusión y debate en el seno del equipo de desarrollo, como retroalimentación del proceso y por tanto se espera que la próxima versión de cada uno de estos sea mucho más eficiente teniendo en cuenta que se han detectado los puntos débiles y los aspectos que pueden incurrir en resultados negativos en las respuestas de las aplicaciones.

Con este capítulo además se demuestra que las pruebas de carga y estrés en la herramienta JMeter no son pruebas esquemáticas, ya que pueden realizarse de diferentes vías dependiendo el tipo de aplicación y los objetivos principales de cada uno de ellos, dependiendo de esto el grado de complicación que presentan cada una de las vías de solución para la realización de los test.

## 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.

Este trabajo hace una descripción total del proceso de prueba de carga y estrés destacando los subprocesos implicados directamente con el trabajo de investigación, se definieron los artefactos necesarios para la correcta planificación, recopilación de datos, métodos de procedimiento para las pruebas, roles implicados y sus respectivas responsabilidades en cada una de las actividades que lo conforman.

Este trabajo presentó un manual que muestra el uso de la herramienta JMeter para la elaboración de las pruebas de carga y estrés. La disponibilidad de este manual supone la completitud de los conocimientos necesarios para realizar estas pruebas de manera automatizada. Ya que este manual está orientado al entendimiento de la herramienta y la importancia de cada uno de los elementos que ella brinda en la elaboración de las pruebas.

Con esta investigación y el manual, fácilmente cualquier especialista implicado en el rol de las pruebas puede aplicar este proceso a su entorno de trabajo y obtener mejores resultados en menos tiempo y con menos recursos, ya que los conocimientos que aquí se reflejan se han expresado de manera organizada ejemplificando cada una de ellos y mostrándolos de manera amena y sencilla.

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. Estos resultados se han obtenido de manera rápida y concreta, ofreciendo un estado real de cada una de las aplicaciones y por tanto sirven para demostrar el nivel de calidad que tienen estas aplicaciones, estos resultados son observados y sometidos a evaluación por el equipo de evaluación del proyecto y ha sido favorable para el SIIPOL mientras se toma el resultado del Portal web como un resultado posible a mejorar cuando este se implante en su medio real.

## Recomendaciones

- ✓ Continuar trabajando en la profundización de los conocimientos sobre el proceso de pruebas, en específico sobre el resto de las pruebas de sistema no funcionales y buscar herramientas que automaticen las mismas.
- ✓ Profundizar el estudio sobre la herramienta JMeter ya que la misma brinda otras funcionalidades que pueden servir para otros tipos de pruebas de sistema no funcionales.
- ✓ Aplicar métodos definidos en la asignatura de teleinformática para utilizar los resultados de estas pruebas en la estimación de resultados para otras condiciones de hardware y red.
- ✓ Exigir a los equipos de desarrollo la utilización de este proceso haciendo uso del JMeter dentro de su proceso de pruebas.
- ✓ Publicar el manual confeccionado para que pueda ser utilizado en otros proyectos productivos de la Universidad.

## 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, Univercidad 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](http://www.noticias.com/noticia). Febrero 2007.
14. BamSol, [www.bamsol.com/bamsol](http://www.bamsol.com/bamsol). 2007.
15. Jakarta, [jakarta.apache.org/jmeter/](http://jakarta.apache.org/jmeter/).
16. Norma ISO 8402 UNE. p. 30
17. Soto, N.P., Cronograma General del Plan de Pruebas del Proyecto 2006.
18. Almenares, L.S., Cómo Realizar Pruebas de Carga y Estrés en JMeter. 2007.
19. Almenares, L.S., <http://10.35.11.32/svn/repn/Expediente> 2007.
20. Almenares, L.S., Estrategia de Pruebas de Carga y Estres. 2007.
21. Almenares, L.S., Test.jmx. 2007.

## Bibliografía

### 1979

Myers (1979). "Fase de Elaboración. FT Prueba (procedimientos genéricos y aplicación de algunos tipos de pruebas simples)."

### 1994

Mañas, J.A., *Prueba de Programas*. 1994.

### 1995

Davis (1995). "Fase de Elaboración. FT Prueba (procedimientos genéricos y aplicación de algunos tipos de pruebas simples)."

Pressman, R. S. (1995). "Ingeniería del software, un enfoque práctico, Mcgraw Hill."

### 1998

Pressman. R. S. (1998). "Ingeniería del software. Un enfoque práctico. McGrawHill".

### 2001

John Watkins (2001). Rational Software. "A Process for Selecting Automated Testing Tools"

Jim Heumann (2001). Rational software."Generating Test Cases From Use Cases"

Brian Bryson (2001). Senior Technical Evangelist. IBM Rational. "Enabling Cost-Effective. Comprehensive Component Testing"

Vincent Encontre (2001). Rational Software. Testing Embedded Systems: Do You Have The GuTs for It?

Dawn Haynes (2001). Automated Testing Evangelist. Rational Software. "Automated Testing: A Silver Bullet?"

Laura Rose (2001). Quality Assurance Manager.Rational TestManager. "Getting the Most from Your Automated Testing Tools"

Sam Guckenheimer (2001). Senior Director,Technology Automated Software QualityGroup. Rational Software. The Revolution in Software Testing

Brian Bryson (2001). Senior Technical Evangelist. IBM Rational. "The Key to Successful Automated Testing: Planning".

Erik Hatcher (2001). "Automating the build and test process".

### 2002

Sam Guckenheimer (2002). Senior Director of Technology for Automated Test. Rational Software "Test-First Design, Exploratory Testing, and Agile Process: An Interview with Brian Marick".

Ingenieria del Softwre III (2002). Universitat de les Illes Balears."Calidad en la Ingeniería del Software."

Fernando Berzal (2002). "Técnicas útiles en el desarrollo del software."

Diego Lucio D'Onofrio (2002). "Probando software y números de versión".

### **2003**

José A. Mañas (2003). Dept. De Ingeniería de Sistemas Telemáticos Universidad Politécnica de Madrid. "Calidad de las Pruebas"

Pete Herzog (2003). "OSSTMM 2.1.Manual de la Metodología Abierta de Testeo de Seguridad".

### **2004**

Scott W. Ambler (2004). Traducido al español por Carlos Méndez (2006) "Método de Pruebas Orientada a Objetos para el Ciclo de Vida Completo (FLOOT)".

José A. Mañas (2004). Dept. De Ingeniería de Sistemas Telemáticos Universidad Politécnica de Madrid "Prueba de Programas".

José A. Mañas (2004). Dept. De Ingeniería de Sistemas Telemáticos Universidad Politécnica de Madrid. " Pruebas de Aceptación".

Cesar Javier Acuña (2004). Ingeniería del Software I. Universidad Rey Juan Carlos. "Pruebas de Software".

Diego Vallespir (2004). Universidad de la República, Grupo de Ingeniería de Software (Gris), Montevideo, Uruguay. "Generación Automática de Casos de Prueba Unitarios para Objetos"

Javier J. Gutiérrez, Darío Villadiego, María J. Escalona, Manuel Mejías (2004). Departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla. "Aplicación de La programación orientada a aspectos en el diseño e implementación de pruebas funcionales".

Ivar Jacobson, G. B., James Rumbaugh (2004). "El proceso unificado de desarrollo de software." 2.

Jacobson Ivar, B. G., Rumbaugh James (2004). "El proceso unificado de desarrollo de software"1.

### **2005**

Michael Kölling (2005). Mærsk Institute, University of Southern Denmark. "Pruebas de Unidad en BlueJ".

María José Roca V. ParqueSoft (2005). "Pruebas de Integración de Productos: Un enfoque práctico".

### **2006**

Sasha Valdés Jiménez (2006). "Modelo de ingeniería de requerimientos".

Martin Fowler (2006). "Continuous Integration".

### **2007**

Wikipedia, I.e.I. (2007). "Rational Unified Process."

José María Toribio Vicente (2007). "JMeter".

Mercury Interactive Corporation (2007). "Tutorial HP Load Runner".

## Glosario de Términos

1. **Ant:** es una herramienta que usa un dialecto en XML para ejecutar las tareas que como desarrollador se le ordene. La naturaleza de las tareas puede ser muy diversa, desde copiar archivos a algún directorio, generar archivos (SWFs, XMLs o incluso clases) "rápidamente" o transferirlas por FTP a un servidor remoto. Es solo un ejemplo del poder de ANT.
2. **Artefacto:** es una información que es utilizada o producida mediante un proceso de desarrollo de software.
3. **API (Application Programming Interface - Interfaz de Programación de Aplicaciones):** es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.
4. **Calidad de Software:** conjunto de propiedades inherentes a un objeto que le confieren capacidad para satisfacer necesidades implícitas o explícitas.
5. **Caso de Prueba:** conjunto de condiciones o variables bajo las cuáles el analista determinará si el requisito de una aplicación es parcial o completamente satisfactorio.
6. **Caso de Uso:** es una técnica para especificar el comportamiento de un sistema: es una secuencia de interacciones entre un sistema y personas u otros sistemas de hardware y software que usa alguno de sus servicios.
7. **Caso de Uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos.
8. **CICPC:** Cuerpo de Investigaciones Central Policial y Criminalísticas.
9. **CP:** Casos de Prueba.
10. **Control de calidad:** "Conjunto de técnicas y actividades de carácter operativo, utilizadas para verificar los requerimientos relativos a la calidad del producto o servicio".
11. **CU:** Casos de Uso.
12. **E-Business:** Cualquier tipo de actividad empresarial realizada a través de las Tecnologías de la Información y las Comunicaciones.
13. **Estrategia de Prueba de Sistema:** es el conjunto de pasos que indican cómo desarrollar de forma organizada las pruebas de sistema a un software o aplicación determinada.
14. **FTP:** son las siglas de File Transfer Protocol, es decir, Protocolo de Transferencia de Archivos. Es un sistema que permite enviar y recibir ficheros entre computadores a través de la red Internet.



15. **Jakarta:** no es un software en sí mismo, sino que más bien se puede ver como un proyecto de proyectos, un repositorio. Está compuesto por varios subproyectos que dan soluciones a problemas en particular.
16. **JDBC:** Es un API de Java, la cual permite conectar los programas escritos en Java con la base de datos.
17. **JNDI ("Java Naming Directory Interface"):** es un API genérico, es una especificación que permite localizar información en distintos directorios distribuidos.
18. **Java 2 Enterprise Edition (J2EE):** es una tecnología compuesta por un conjunto de especificaciones de APIs Java para la construcción de aplicaciones empresariales.
19. **Garantía de calidad:** "Conjunto de acciones planificadas y sistemáticas necesarias para proporcionar la confianza adecuada de que un producto o servicio satisfará los requerimientos dados sobre calidad".
20. **Manual de Procedimiento:** es el conjunto de pasos lógicos ordenados cronológicamente que indican cómo proceder en determinado proceso.
21. **Metalinguaje:** es un lenguaje usado para hacer referencia a otros lenguajes.
22. **Métricas de software:** proporcionan información objetiva que contribuye al mejoramiento de los procesos y productos de software.
23. **Modelado:** es una técnica cognitiva que consiste en crear una representación ideal de un objeto real mediante un conjunto de simplificaciones y abstracciones, cuya validez se pretende constatar. La validación del modelo se lleva a cabo comparando las implicaciones predichas por el mismo con observaciones.
24. **LDAP (Lightweight Directory Access Protocol):** es un protocolo de red.
25. **Lenguaje de marcas:** es una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación.
26. **MySQL:** es un sistema de gestión de base de datos relacional, multihilo y multiusuario.
27. **.NET:** La plataforma de Microsoft orientada a la creación de software para Internet, entorno de ejecución en el que se cargan las aplicaciones desarrolladas en distintos lenguajes.
28. **Oracle:** es una potente herramienta cliente/servidor para la gestión de Bases de Datos.
29. **Paht:** Dirección electrónica.
30. **PHP (acrónimo de PHP: Hypertext Preprocessor):** es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor.

31. **Pruebas de Sistema:** son las que se le realizan al software una vez que se termina la fase de construcción.
32. **RAM (Memoria de Acceso Aleatorio- Random Access Memory):** es la que se encarga de almacenar la información que el computador está utilizando mientras se encuentra encendido.
33. **RUP (Proceso Unificado Racional-Rational Unified Process):** es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.
34. **Sistema de Software:** es el conjunto de programas que administra los recursos de hardware.
35. **SOAP:** es un protocolo elaborado para facilitar la llamada remota de funciones a través de Internet.
36. **Software:** es el conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema.
37. **TI(tecnologías de la información y la comunicación):** son un conjunto de servicios, redes, software, aparatos que tienen como fin el mejoramiento de la calidad de vida de las personas dentro de un entorno, y que se integran a un sistema de información interconectado y complementario.
38. **UML (Lenguaje Unificado de Modelado-Unified Modeling Language):** es un lenguaje de modelado de sistemas de software, es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.
39. **XML (Extensible Markup Language-lenguaje de marcas extensible):** es un metalenguaje extensible de etiquetas.

## Anexos

### Anexo 1 Reporte de Estado de las PC

No. PC	Descripción general del estado de la PC	Por ciento de Cumplimiento
PC 27 lab 207	1PC. 0,99mb de Ram. M Office. Java Virtual Machine 1.3 o superior. JMeter. Navegador IExplorer.	100%

### Anexo 2 Especificación de Recursos

Tarea	Recurso	Cantidad de personas	Fecha límite
Elaboración del plan de pruebas	1PC. M Office.	1	14/05/2008
Realizar la grabación y los test en la herramienta	1PC. 0,99mb de Ram. M Office. Java Virtual Machine 1.3 o superior. JMeter. Navegador IExplorer.	1	15/05/2008
Ejecución de las pruebas	1PC. 0,99mb de Ram. M Office. Java Virtual Machine 1.3 o superior. JMeter. Navegador IExplorer. Se necesita además que se encuentre corriendo el servidor de la aplicación.	1	21/05/2008
Análisis de los resultados	1PC. M Office.	3	23/05/2008

## Anexo 3 Reporte de los resultados de la Prueba de carga para 450 iteraciones del modulo de investigaciones penales.

<b>Cantidad de repeticiones</b>	450								
<b>Características en que fue realizada la prueba: Red LAN a 100Mbps.</b>									
<b>Petición</b>	<b>Muestras</b>	<b>Media</b>	<b>Mediana</b>	<b>Min</b>	<b>Max</b>	<b>Línea 90 %</b>	<b>%Error</b>	<b>Rendimiento</b>	<b>Kb/seg</b>
Iniciar sección	1	109	109	109	109	109	0.0	4,6/seg	13018,3
/calidadinterna/images/escritorioTrabajo/RedoCuerpo.jpg	64	27	16	0	1078	16	0.0	11,6/min	136,3
/calidadinterna/images/escritorioTrabajo/loading.gif	64	9	15	0	31	16	0.0	11,6/min	680,1
/calidadinterna/images/escritorioTrabajo/SIIPOL_Derecha.jpg	64	8	15	0	16	16	0.0	11,6/min	108,8
/calidadinterna/images/escritorioTrabajo/logo_cicpc.jpg	64	11	15	0	31	16	0.0	11,6/min	1900,5
/calidadinterna/images/escritorioTrabajo/SIIPOL_Inferior.jpg	64	9	15	0	31	16	0.0	11,6/min	109,8
/calidadinterna/images/escritorioTrabajo/bg_menu.jpg	64	10	15	0	47	16	0.0	11,6/min	129,8
/calidadinterna/images/escritorioTrabajo/SIIPOL_IzqInf.jpg	64	9	15	0	32	16	0.0	11,6/min	160,6

ANEXOS

/calidadinterna/images/escritorioTrabajo/SIIPOL_Derlnf.jpg	64	11	15	0	110	16	0.0	11,7/seg	150,4
/calidadinterna/escritorioDeTrabajo.jsf?javax.portlet.faces.DirectLink=true	3962	23	0	0	9172	31	0.0	34,7//min	2856,2
/calidadinterna/faces/cicpc_lib/dataScroller/css/datascroller.css	192	16	15	0	328	16	0.0	3,8/seg	145
/calidadinterna/faces/a4j_3_1_4.GAimages/spacer.gif	1280	9	15	0	344	16	0.0	42,2/min	165,4
/calidadinterna/faces/a4j_3_1_4.GAorg.richfaces.renderkit.html	256	9	15	0	172	16	0.0	1,6/min	829,2
/calidadinterna/faces/a4j_3_1_4.GAorg.richfaces.renderkit.images	64	7	15	0	16	16	0.0	1,6/min	10,4
/calidadinterna/images/pest_activa.gif	64	8	16	0	31	16	0.0	1,6/min	10
/calidadinterna/images/escritorioTrabajo/bak.gif	64	8	16	0	46	16	0.0	1,5/min	12,9
/calidadinterna/images/pest_inactiva.gif	64	8	15	0	31	16	0.0	1,6/min	10,2
/calidadinterna/faces/a4j_3_1_4.GAorg.richfaces.renderkit.html	64	9	16	0	16	16	0.0	1,6/min	8,7
/calidadinterna/estilos/style.css	6077	17	16	0	1078	16	0.0	18,0/seg	285479,4
/calidadinterna/faces/cicpc_lib/somJSON/json.js	64	41	15	15	1203	32	0.0	1,5/min	972,9
/calidadinterna/images/mensajadver.jpg	64	14	15	0	344	16	0.0	1,6/min	479
/calidadinterna/analisisinformacion/persona/js/script.js	64	21	15	0	203	31	0.0	1,5/min	987

ANEXOS

/calidadinterna/analisisinformacion/persona/js/validaciones.js	64	22	16	0	297	16	0.0	1,5/min	306,9
/calidadinterna/images/mensajeinfor.jpg	64	25	15	0	593	31	0.0	1,5/min	473,4
/calidadinterna/analisisinformacion/objeto/js/script.js	64	14	16	0	187	16	0.0	1,5/min	610,5
/calidadinterna/analisisinformacion/objeto/js/validaciones.js	64	22	16	0	609	16	0.0	1,5/min	169,1
/calidadinterna/analisisinformacion/arma/js/validaciones.js	64	13	15	0	32	16	0.0	1,5/min	263,5
/calidadinterna/analisisinformacion/arma/js/script.js	64	13	15	0	16	16	0.0	1,5/min	461,8
/calidadinterna/analisisinformacion/vehiculo/js/validaciones.js	64	20	16	0	250	16	0.0	1,5/min	1028,3
/calidadinterna/analisisinformacion/vehiculo/js/script.js	64	13	15	0	31	16	0.0	1,5/min	1180
/calidadinterna/images/escritorioTrabajo/menuDespleg.gif	64	9	16	0	16	16	0.0	1,5/min	20,6
/calidadinterna/js/modalbox/spinner.gif	64	10	15	0	31	16	0.0	1,5/min	1504,4
/calidadinterna/investigacionpenal/subviews/js/validaciones.js	64	14	15	0	31	16	0.0	1,5/min	353,6
/calidadinterna/investigacionpenal/investigacion/js/funcionesComunes.js	64	13	15	0	32	16	0.0	1,5/min	189,6
<b>Total</b>	13624	17	16	0	917	16	0.0	40,2 /seg	301162,6

## Anexo 4. Resultado general de las Pruebas AI Portal Web.

Url	Muestras	Media	Mediana	Línea 90 %	Min	Max	% Error	Rendimiento	Kb/seg
/portalcicpc/	2700	663	94	234	0	51984	0.4	25,8/seg	158841.1
/portalcicpc/mision-vision	50	16960	15906	15906	734	47985	0	20,7/min	5116
/portalcicpc/valores-institucionales	50	15710	13766	35281	734	54484	0	18,9/seg	4499
/portalcicpc/funciones	50	13681	11547	30500	1343	44422	0	18,9/seg	5168
/portalcicpc/objetivos	50	16156	12375	31828	719	45625	0	17,8/seg	6416.5
/portalcicpc/historia	200	3721	31	16329	0	31281	0	1,1/seg	22797.6
/portalcicpc/simbolos	200	3721	47	12954	0	24781	25	1,1/seg	10441.1
/portalcicpc/directiva	300	2740	32	12672	0	37718	0.33	1,5/seg	55450.6
/portalcicpc/bases-juridicas	50	14904	15125	23110	4109	30750	0	1,5/seg	4035.3
/portalcicpc/organigrama	250	3181	46	13110	0	55032	0	16,5/min	18778.6
/portalcicpc/vias-de-ingreso	50	14862	13656	24890	3110	39360	0	1,1/seg	4059.2
/portalcicpc/noticias	150	5168	63	15297	0	35922	0	15,5/min	27087.8
/portalcicpc/noticias/2008/05/20/introduccion-la-nueva-gerencia	150	5542	79	16265	0	36688	0	42,2/min	8628.6
/portalcicpc/event	50	17402	16953	24484	8078	34609	0	48,2/min	4532.6
/portalcicpc/tesis	50	16674	15969	25157	6703	32984	0	40,6/min	3749.3

ANEXOS

/portalcicpc/tesis/1/mostrar	50	15718	15515	19828	1718	31203	0	14,4/min	3769.2		
/portalcicpc/casoscangrejos	50	5179	32	17516	0	33765	0	14,7/min	3708.1		
/portalcicpc/proyectos	150	21791	20344	33813	9797	56296	1	14,8/min	14263.3		
/portalcicpc/preguntasfrecuentes	100	16390	14969	20860	8250	33218	0	40,8/min	7342.1		
/portalcicpc/contactos-institucionales	50	16805	14594	23531	9125	32406	0	24/min	4390		
/portalcicpc/mediateca/archivos	50	16200	15360	23141	9656	26110	0	13,8/min	3134		
/portalcicpc/mediateca/archivos/documentos	50	26113	21890	36062	9875	204065	0	12,7/min	6434.8		
/portalcicpc/mediateca/3/download	100	19861	10453	45469	0	253508	0	12,9/min	2836.5		
/portalcicpc/files/mediateca/documentos/Plan%20de%20la%20nacion%202007-2013.doc	48	59933	24750	19896	8	8750	258236	0	9,2/min	1034	
/portalcicpc/mediateca/archivos/videos	42	29104	15735	73375	6187	115882	0	8/min	980		
/portalcicpc//portalcicpc/sites/all/themes/cicpc/images/menu-over.gif	41	233	172	422	78	828	0	3,5/min	35732		
/portalcicpc/asesoriajuridica	41	87199	29843	24343	1018	5	7	252248	0	4,6/min	54469
/portalcicpc/centro-telefonico-de-atencion-al-ciudadano-ctac	62	145	16	125	0	3593	0	6,9/min	797		
/portalcicpc/user/register	31	86598	54609	21450	1004	243392	0	3,2/min	10.8		



ANEXOS

				0	7				
/portalcicpc//portalcicpc/?q=user%2Fregister	26	70175	120781	25199 9	1325 0	269513	0	12,2/min	842.6
/portalcicpc/articulos/2008/05/21/el-padre-de-la-criminalistica-en-venezuela	55	98319	125	25669 0	0	415516	0	2,4/min	697.2
/portalcicpc//portalcicpc/?q=como-considera-esta-pagina	6	282830	115125	23888 7	62	238887	0	2,1/min	786
/portalcicpc/centro-telefonico-de-atencion-al-ciudadano-ctac	0	0	0	0	0	0	0	0	0
<b>Total</b>	5335	7010	110	18985	0	415516	0.009	4.090/seg	80022,7