

**Universidad de las Ciencias Informáticas**

**Facultad 3**



**Título: Evaluación del rendimiento en la solución Tablas de control.**

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

**Autor:** Lisandra Tamayo Espinosa

**Tutor:** Ing. Raymond Weeden Gamboa

**Junio del 2012**

## **DECLARACIÓN DE AUTORÍA**

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales con carácter exclusivo, permitiéndole hacer uso de la misma en su beneficio.

Para que así conste firmo la presente a los 29 días del mes de julio del año 2012.

\_\_\_\_\_  
Lisandra Tamayo Espinosa

\_\_\_\_\_  
Ing. Raymond Weeden Gamboa



*"La calidad nunca es casual; es siempre el resultado de fuerte determinación, sinceros esfuerzos, dirección inteligente y ejecución sumamente hábil; representa la acertada elección entre varias alternativas."*

*William A. Foster*

**AGRADECIMIENTOS**

*Agradezco con todas las fuerzas de mi corazón a mi familia de la que estoy muy orgullosa, en especial a mis padres por todo el cariño que me han dado, por confiar en mí y apoyarme en todo momento. A mis hermanas Laura y Leticia que han sido dos madres más, por su ternura y por los consejos que siempre me han brindado. A mis sobrinos que tanto adoro.*

*A mis tías Marina e Iraida que han estado a mi lado en los momentos más difíciles. A todas aquellas personas especiales que me cuidaron y me dieron cariño cuando tan lejos estuve de mi país.*

*A mi tutor Raymond por soportarme 3 años en el proyecto. A mis amigos y compañeros de tantos años de estudio, algunos que no veré más pero nunca olvidaré porque son muy importantes en mi vida. A mi novio por ser tan lindo conmigo y estar siempre a mi lado.*  
*A todos ellos, muchas gracias.*

**DEDICATORIA**

*Dedico mi tesis a mis papis queridos que son el tesoro más grande que tengo. A ellos que me dieron la vida, tiempo y esfuerzo para llegar a donde hoy estoy. A esas personitas tan especiales que me hacen sentir la persona más afortunada del mundo.*

*A mis hermanas que me han dado mucho cariño y me tratan como su niña. A mis sobrinos Yony, Claudia y Mariannis.*

## RESUMEN

Al trabajar con aplicaciones web uno de los problemas que frecuentemente encontramos es la respuesta demasiado lenta ante una petición, en ello influyen las características de hardware, pero el mayor peso reside en los valores de rendimiento. Conocer la carga de trabajo que soporta la aplicación y mejorar los tiempos de respuesta de las páginas constituye una necesidad para lograr la satisfacción del cliente. Este trabajo está centrado en la evaluación de la satisfacción del requisito no funcional de rendimiento en la solución Tablas de Control de la Aduana General de la República. Se describen elementos de importancia relacionados con calidad de software, métricas, normas y estándares, tipos de pruebas, herramientas para la automatización de las pruebas y aspectos referentes al trabajo con los requisitos de software. Se propone un procedimiento enfocado en el diseño de los casos de prueba utilizando los requerimientos de software, contando con una distribución clara de los roles a utilizar, las tareas que estos deben desarrollar para cumplir los objetivos propuestos, así como de una descripción lógica y organizada de las actividades agrupadas por fases, con los artefactos de entrada y de salida de éstas.

Se aplica el procedimiento propuesto para realizar las pruebas al módulo Tablas de Control, quedando documentados todos los pasos realizados. Se hace una valoración al comparar los valores reales con los esperados y se evalúa el rendimiento de la aplicación mediante las pruebas de Carga y Estrés. En la fase final se recogen los errores encontrados, lo cual contribuye al mejoramiento de la calidad.

## PALABRAS CLAVES

Calidad de Software, Requerimientos de software, Pruebas, Rendimiento, Procedimiento, Pruebas de Carga y Estrés.

Índice

DECLARACIÓN DE AUTORÍA .....	I
AGRADECIMIENTOS .....	III
DEDICATORIA .....	IV
RESUMEN .....	V
INTRODUCCIÓN .....	1
1.    Fundamentación teórica .....	6
Introducción .....	6
1.1    Calidad de software.....	6
1.1.1    Calidad de software en la actualidad .....	6
1.1.2    Conceptos relacionados con el proceso de calidad .....	8
1.1.3    Normas de calidad de software .....	9
1.1.4    Métricas de calidad.....	10
1.1.5    Estándares y Modelos de Calidad .....	11
1.2    Pruebas de software .....	13
1.2.1    Principios de las pruebas.....	14
1.2.2    Objetivos de las pruebas de software .....	14
1.2.3    Tareas a realizar para probar un software .....	15
1.2.4    Tipos de pruebas de rendimiento .....	15
1.3    Metodología Proceso Unificado de Desarrollo (RUP) .....	17
1.4    Herramientas utilizadas para la administración y ejecución de pruebas de rendimiento ....	17
1.5    Ingeniería de Requisitos.....	19
1.5.1    Requisitos no funcionales.....	19
1.6    Procedimientos para pruebas de rendimiento .....	21
1.6.1    Método para validar procedimientos .....	23
Conclusiones .....	24
2.    Descripción del Proceso de Pruebas.....	25
Introducción .....	25
2.1    Resumen .....	25
2.2    Nombre del procedimiento.....	25
2.3    Alcance.....	25
2.4    Selección de roles.....	27
2.5    Objetivos.....	28

---

2.6	Distribución de las tareas por roles .....	29
2.7	Fases a desarrollar en el proceso de pruebas .....	29
2.7.1	Planificación de las pruebas .....	30
2.7.1.1	Selección del equipo de prueba.....	30
2.7.1.2	Estudio de la documentación reunida del proyecto .....	30
2.7.1.3	Análisis de los recursos materiales.....	31
2.7.2	Diseño de las pruebas .....	31
2.7.2.1	Creación del Modelo de análisis de carga .....	32
2.7.2.2	Configuración del entorno de pruebas .....	33
2.7.2.3	Diseño de los casos de prueba.....	34
2.7.2.4	Elaboración del Plan de Pruebas.....	35
2.7.3	Ejecución de las pruebas.....	36
2.7.3.1	Registro de los resultados .....	37
2.7.3.2	Gestión de las no conformidades.....	37
2.7.4	Evaluación de los resultados .....	38
2.8	Validación del procedimiento.....	38
	Conclusiones .....	44
3.	Pruebas de rendimiento al módulo Tablas de Control .....	45
	Introducción .....	45
3.1	Características fundamentales del módulo .....	45
3.2	Planificación.....	46
3.2.1	Estudio de la documentación.....	46
3.2.2	Análisis de los recursos materiales.....	47
3.3	Diseño de las pruebas .....	48
3.3.1	Modelo de análisis de carga .....	48
3.3.2	Configuración del entorno de prueba.....	50
3.3.3	Diseño de los casos de prueba.....	51
3.4	Ejecución de las pruebas .....	53
3.4.1	Funcionalidad Procesar Fichero .....	53
3.4.2.	Funcionalidad Dar Alta .....	55
3.4.3.	Funcionalidad Dar Baja .....	57
3.4.4.	Funcionalidad Modificar.....	58
3.4.5.	Funcionalidad Buscar Registro .....	60
3.5	Evaluación de las pruebas .....	63
	Conclusiones.....	66
	CONCLUSIONES .....	67

RECOMENDACIONES .....	68
REFERENCIA BIBLOGRÁFICA.....	69
ANEXOS.....	71
Anexo1: Encuesta.....	71
Anexo2: Caso de Prueba 2 .....	72
Anexo 3: Caso de Prueba 2 .....	73
Anexo 4: Caso de Prueba 3 .....	74
Anexo 5: Caso de Prueba 3 .....	75
Anexo 6: Caso de Prueba 4 .....	76
Anexo 7: Caso de Prueba 4 .....	77
Anexo 8: Caso de Prueba 5 .....	78
Anexo 9: Caso de Prueba 5 .....	80
Anexo 10: Resultado de la funcionalidad Modificar Registro para 15 usuarios.....	81
Anexo 11: Criterios de Criticidad .....	81

### Índice de Figuras

Figura 1: Modelo de calidad.....	9
Figura 2: Características y subcaracterísticas de calidad del estándar ISO 9126.....	12
Figura 3: Resultados del análisis a los procedimientos .....	23
Figura4: Descripción del procedimiento. ....	26
Figura 6: Actividades de la fase de planificación .....	30
Figura 7: Actividades de la fase de diseño .....	31
Figura 8: Procedimiento para análisis de la carga.....	33
Figura 9: Actividades fase de ejecución .....	36
Figura 10: Actividades de la fase análisis de las pruebas .....	38
Figura 11: Distribución de la red .....	48
Figura 12: Rendimiento para las Pruebas de Carga.....	63
Figura 13: Tiempo de Respuesta para las Pruebas de Carga.....	64
Figura 14: Rendimiento para las Pruebas de Estrés .....	64
Figura 15: Tiempo Promedio de Respuesta para las Pruebas de Estrés.....	65

## INTRODUCCIÓN

El desarrollo de un software se hace con la expectativa de obtener un resultado flexible, rápido, confiable, seguro, fácil de manejar y sobre todo que cumpla con las necesidades predefinidas. Al profundizar en el desarrollo la posibilidad de cometer errores aumentan, ya sea por mala especificación de los requisitos, uso indebido de las estructuras de datos o problemas al enlazar módulos. Lograr un producto completo es bien complicado, de ahí la necesidad de unir el desarrollo de software con actividades que garanticen la calidad.

Para obtener un software con calidad se requiere de una revisión exquisita y continua que asegure el cumplimiento de las necesidades propuestas inicialmente. Esas necesidades se reconocen como requisitos y tienen como propósito lograr que los integrantes del grupo de prueba convenzan al cliente de que el sistema satisface sus expectativas y se dividen en: Requisitos Funcionales (RF) y Requisitos No Funcionales (RNF). Los RNF centran su atención en las necesidades del cliente, definen las restricciones y propiedades del sistema por ejemplo fiabilidad, usabilidad, mantenimiento, seguridad, eficiencia, portabilidad, confiabilidad, tiempo de respuesta y requisitos de almacenamiento.

Generalmente las pruebas no se realizan con la calidad requerida, no se implementan de forma organizada y sistemática, y mucho menos cuando se trata de evaluar los requisitos no funcionales. Cuba ha avanzado en el desarrollo de software con calidad y el surgimiento de la Universidad de las Ciencias Informáticas (UCI) ha sido un eslabón fundamental. En ella se encuentra el proyecto Gestión Integral de Aduanas (GINA) que pertenece al Centro de Informatización y Gestión de Entidades (CEIGE) de la Facultad 3. En dicho proyecto se desarrolla la solución para la configuración de tablas de control con el fin de gestionar todas las Tablas de Control (TC) de la Aduana General de la República (AGR) de una forma dinámica, así como la interacción mediante servicios con las demás aplicaciones del proyecto.

### **Situación Problemática:**

El módulo TC se especializa en la gestión de los nomencladores (término usado en el proyecto AGR para clasificar y codificar conceptos importantes que forman un sistema rápido y flexible), que se realiza dinámicamente mediante una interfaz única a partir de meta-datos y se almacenan en tablas, las cuales guardan por petición del cliente datos históricos, lo que pudiera afectar a las aplicaciones en cuanto al rendimiento en las búsquedas de datos necesarios o en validaciones

propias de negocio. Existen peculiaridades en la forma de gestión de los datos por ejemplo a la hora de eliminar un registro este no se elimina físicamente de la base de datos sino que se pone en un estado cerrado o inutilizable y en ese mismo estado se deben poner a cada uno de los registros que dependen de él en cada una de las relaciones automáticamente, cuando se va a modificar algún registro se deben actualizar los id (identificador de cada tabla) de todos los que dependen de él en cada una de las tablas. Por ello la aplicación estará interactuando constantemente con una gran cantidad de datos y al mismo tiempo deberá responder a múltiples peticiones de los usuarios, influyendo en su rendimiento y constituyendo un riesgo en el desempeño del sistema. Este módulo es uno de los indispensables para el funcionamiento de cualquier subsistema en el GINA, ya que todas las informaciones se brindan en catálogos de datos.

Esta solución se enfrentó a un proceso de pruebas por parte del cliente, específicamente en el Centro para la Automatización y Digitalización de la Información (CADI), sin embargo, estas no garantizan el cumplimiento de los requisitos no funcionales del sistema en los entornos de explotación y no aseguran que la aplicación trabaje eficientemente al conectarse varios usuarios, al no tener en cuenta la rapidez al procesar las transacciones ni el tiempo de respuesta a los usuarios y eventos al trabajar con datos reales. Para lograr que la aplicación responda a las peticiones de los usuarios de una forma adecuada y rápida, se hace necesario realizar pruebas que determinen realmente el desempeño que pudiera tener la solución.

Por todo lo antes planteado es fundamental reconocer las potencialidades y características para explotarlas al máximo, perfeccionar su funcionamiento, encontrar las debilidades que obstaculicen el buen desempeño de la aplicación y realizar un efectivo proceso de pruebas. Se plantea el siguiente **problema**: ¿Cómo comprobar la satisfacción del requisito de rendimiento en la solución Tablas de Control del Departamento soluciones para la Aduana?

El **objeto de estudio** está enmarcado en el proceso de pruebas a los requisitos no funcionales y el **campo de acción** está dirigido al proceso de investigación, comportamiento, desarrollo y aplicación del procedimiento para las pruebas de rendimiento en la solución TC. Como **objetivo general** se definió, desarrollar un procedimiento para comprobar la satisfacción del requisito no funcional de rendimiento en la solución Tablas de Control del departamento Soluciones para la Aduana.

Para lograr resultados satisfactorios se trazaron los siguientes **objetivos específicos**:

- ✓ Desarrollar un procedimiento de pruebas de carga y estrés acorde a los objetivos propuestos tras un estudio a los procedimientos actuales.
- ✓ Aplicar el procedimiento con la ejecución de pruebas que garanticen la satisfacción del requisito no funcional de rendimiento en la solución Tablas de Control.
- ✓ Comparar si los resultados obtenidos en las pruebas de rendimiento satisfacen los requerimientos propuestos por el cliente.

Se formuló la siguiente **Hipótesis de investigación**:

Si se desarrolla un procedimiento para la realización de las pruebas de carga y estrés en el módulo Tablas de Control, entonces se podrá comprobar el requisito de rendimiento.

Para lograr los objetivos trazados y demostrar la hipótesis establecida se acometieron las siguientes **tareas**:

- ✓ Analizar conceptos relacionados con: normas, estándares, artefactos y métricas de calidad, gestión de la calidad de software, control de la calidad y tipos de pruebas a los requisitos no funcionales.
- ✓ Investigación sobre las diferentes normas, estándares y métricas de calidad.
- ✓ Identificación de los tipos de pruebas más acordes con el análisis de calidad que se pretende.
- ✓ Estudio de métodos de planificación, desarrollo y evaluación de las pruebas.
- ✓ Definir un conjunto de artefactos, tareas, roles y fases a utilizar durante el proceso de pruebas.
- ✓ Elaborar la documentación que evidencie los resultados de las evaluaciones.
- ✓ Comparación de los resultados de las pruebas de carga y estrés.

Para la obtención de resultados positivos en la realización de las tareas de investigación se plantea la necesidad del uso de diferentes métodos científicos, tanto teóricos como empíricos; en busca de una estrategia general que oriente y permita organizar la actividad hacia el conocimiento y la solución del problema.

### **Métodos teóricos:**

El **análisis documental**: facilitó el análisis y sistematización del tema en cuestión, a partir de los documentos de autores nacionales y extranjeros.

El método **hipotético-deductivo** permitió la formulación de la hipótesis central de la investigación.

**La modelación** ayudó a comprender el flujo de actividades a realizar en el procedimiento, siendo una forma más clara y explícita.

El método **sistémico** se empleó para lograr que los elementos que forman parte del procedimiento sean un todo que funcione de manera armónica.

El método **histórico-lógico** y el **dialéctico se emplearon** para el estudio crítico de los procedimientos existentes, para utilizarlos como punto de referencia y compararlos con los resultados alcanzados.

El método **analítico-sintético** ayudo a 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.

#### **Métodos empíricos:**

El método de la **entrevista** se utilizó con el fin de obtener los problemas presentes en las empresas cubanas de desarrollo de software.

**La encuesta** se empleó para obtener información acerca de la percepción que tiene una o un grupo de personas acerca del procedimiento a través de una encuesta elaborada previamente.

Para comprobar la utilidad de los resultados obtenidos a partir del procedimiento definido se prescindió del **método experimento**.

El presente trabajo está estructurado en tres capítulos donde se describe el camino a seguir para lograr una valoración integral del nivel de calidad de la solución del sistema Tablas de Control, además de introducción, conclusiones, recomendaciones, referencias bibliográficas y anexos.

**Capítulo 1:** La Fundamentación Teórica contiene elementos esenciales de la calidad de software, como: evolución histórica, conceptos relacionados con calidad, información sobre normas, estándares, métricas, metodologías y herramientas, haciendo énfasis en las pruebas que se deben realizar para verificar requisitos no funcionales.

**Capítulo 2:** Se describe detalladamente el proceso de prueba, se elabora un procedimiento acorde a las necesidades del proyecto describiendo paso a paso las actividades que se desarrollarán para llevar a cabo las pruebas, así como la especificación de roles, artefactos, tareas, objetivos y fases.

**Capítulo 3:** En este capítulo se ejecutan las pruebas diseñadas y se realiza una evaluación de todos los resultados obtenidos.

# Capítulo I

## 1. Fundamentación teórica

### Introducción

El presente capítulo hace un estudio de los temas generales relacionados con calidad de software, conceptos importantes como: evolución histórica, normas y estándares que hacen de la calidad un proceso menos complicado. Se describen varios tipos de pruebas, enfatizando en las de rendimiento, así como las herramientas que se utilizarán para desarrollarlas. Las pruebas son la base fundamental del presente trabajo y centran su atención en verificar el cumplimiento de los requisitos no funcionales.

### 1.1 Calidad de software

La calidad se define como la ausencia de deficiencias: "Es la totalidad de aspectos y características de un producto o servicio que se refieren a su capacidad para satisfacer necesidades dadas en la adecuación de sus objetivos" (1). También se conoce como: grado en el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario (2). Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo prefijados y con las características implícitas que se esperan de todo software desarrollado profesionalmente (3).

Todos los conceptos estudiados coinciden en que la calidad es una característica o cualidad que muestra que tan útil es el producto. Calidad se relaciona con excelencia, perfección, eficiencia, virtud, corrección, funcionalidad, confiabilidad, usabilidad, portabilidad, flexibilidad, seguridad, integridad, conceptos que deben estar presentes en todo desarrollo de un proyecto.

#### 1.1.1 Calidad de software en la actualidad

Ana Ascasso, responsable de las soluciones de calidad de COMPUWARE, empresa americana dedicada al estudio de la calidad, publicó un artículo con datos interesantes sobre el comportamiento de la calidad de software. El estudio se realizó en 24 países europeos con un total de 184 entrevistas a responsables de los departamentos de Tecnología de la Información (4).

Por ejemplo:

- El 23,9% de las organizaciones admiten que sus equipos de calidad carecen de formación y de experiencia, en el 30,5% de los casos, los equipos sí tienen experiencia pero no están dedicados en la totalidad de su tiempo a asegurar la calidad.
- El 78,3% de las empresas fallan a la hora de aplicar una metodología formal para asegurar la calidad.
- El 44,5% de las empresas declaran que la calidad la gestionan a nivel departamental y que implantan diferentes metodologías en la misma organización.
- El 50% de los proyectos de tecnología informática no consiguen sus objetivos, y en la mayoría de los casos ello se debe a una pobre metodología de calidad desde el principio.

El estudio destacó favorablemente a países como Austria, Reino Unido, Francia, Alemania y España.

Actualmente la calidad tiene un impacto estratégico, la competencia es un tema de interés para toda persona vinculada al desarrollo de software, las necesidades son cada vez mayores y para satisfacerlas surgen métodos que hacen más fácil el trabajo y obtienen un mejor resultado.

Citmatel es la empresa cubana de Tecnologías de la Información y Servicios Telemáticos Avanzados, tiene gran distinción y prestigio entre las entidades cubanas por su excelencia y creciente proyección hacia el mercado externo e interno con una amplia diversidad de productos y servicios integrales. Se encuentra entre las más importantes editoriales cubanas, con títulos reconocidos por su calidad, muchos de ellos premiados en ferias y exposiciones nacionales e internacionales. En esta empresa no llevan a cabo un diseño de casos de pruebas, sino que realizan un cuestionario para evaluar distintas características de calidad que debe tener el software tales como: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad, portabilidad y reusabilidad. Dentro de cada característica se evalúan también las sub-características que la integran.

La UCI está integrada por facultades que se dividen en varios centros, en cada uno de ellos se encuentra un grupo de calidad especializado en las actividades de Pruebas de Funcionalidad, afianzándose en la técnica de Caja Negra. Para realizar el resto de las pruebas la universidad cuenta con Calisoft (Centro para la excelencia en el desarrollo de proyectos tecnológicos) que se dedica a certificar la calidad de todos los productos de software que se producen en la institución

y facilita a los proyectos las mejores prácticas. El centro asegura que cada artefacto trabaje de forma adecuada al terminar todas las pruebas y para ello cuenta con varios laboratorios industriales para pruebas de software (LIPS). El trabajo en los laboratorios se divide en grupos y cada uno se especializa en un tipo de prueba a los RNF, utilizando para ello una estrategia para ganar en tiempo y en recursos.

### 1.1.2 Conceptos relacionados con el proceso de calidad

El proceso de calidad se reconoce como una obra con un alto grado de complejidad, entender cómo funciona y aplicarlo en un proyecto lleva trabajo y dedicación. Para facilitar el estudio se decidió profundizar en conceptos que hacen menos complejo el proceso y algunos de ellos a continuación se exponen:

**Sistema de gestión de la calidad:** Conjunto de responsabilidades, procedimientos, procesos y recursos que se establecen para llevar a término la gestión de la calidad (5).

**Gestión de la Calidad de Software:** Aspecto de la función de gestión que determina y aplica la política de la calidad, los objetivos y las responsabilidades, que realiza con medios tales como la planificación, control, garantía y mejora de la calidad (5). Tiene como objetivo entender las expectativas del cliente y aplicar un conjunto de actividades para hacerlas cumplir. Se basa en las políticas de la empresa y llega hasta la gestión de cada proyecto.

**Control de la Calidad de Software:** Conjunto de técnicas y actividades de carácter operativo, utilizadas para verificar los requisitos relativos a la calidad, centrados en mantener bajo control el proceso de desarrollo. Debe comprobar que el producto final funciona correctamente de acuerdo a las especificaciones y en colaboración con otros sistemas de software y de base de datos (6). Es una observación constante, una vigilancia permanente a todo el proceso de desarrollo y ciclo de vida del software.

**Garantía de calidad:** Conjunto de acciones planificadas y sistemáticas necesarias para proporcionar la confianza adecuada de que un producto o servicio satisface los requerimientos dados sobre calidad (5).

**Administración de pruebas:** Conjunto de pasos encaminados a lograr un manejo integral del proceso de pruebas, garantiza que se detecte una mayor cantidad de errores sin que se pierda

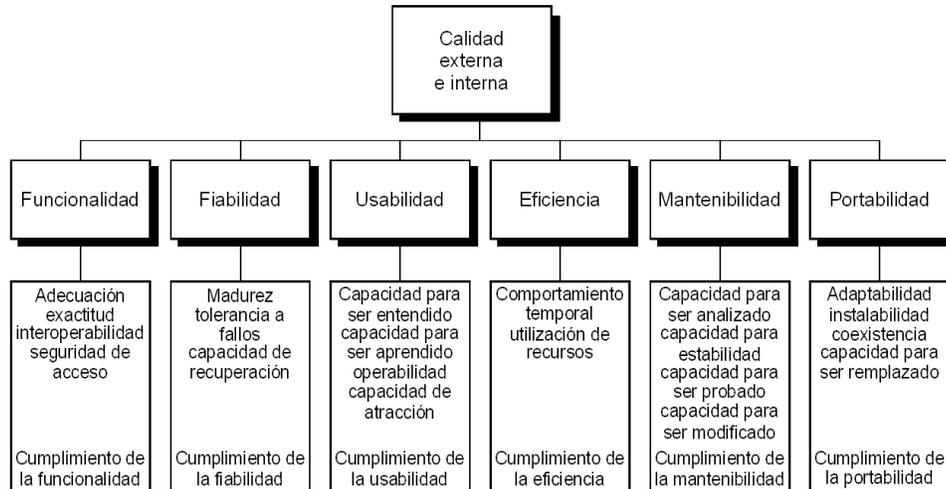
tiempo y recursos. Brinda facilidades en el control o chequeo de cada una de las tareas a realizar.

**Arquitectura:** Toma en consideración elementos de calidad del sistema tales como: rendimiento, reutilización y capacidad de evolución, por lo que debe ser flexible durante todo el proceso de desarrollo. La arquitectura se ve influenciada por la plataforma de software, sistema operativo, gestor de bases de datos, protocolos y consideraciones de desarrollo.

**Artefacto:** Producto tangible del proyecto que se produce y se modifica. Puede ser un modelo o sus elementos, documentos, gráficos, código fuente y ejecutables.

### 1.1.3 Normas de calidad de software

Existen normas que aumentan la productividad disminuyendo los costos de producción, simplificando el trabajo y facilitando la capacitación del personal, una de éstas es la norma 5420, dedicada a la Calidad del producto de software y plantea lo siguiente:



**Figura 1:** Modelo de calidad

La Organización Internacional de Normalización (ISO) es una norma de gran popularidad en el proceso de desarrollo. La ISO 9000 tiene como objetivo desarrollar un código con prácticas de administración para garantizar el aseguramiento y administración de la calidad, dice que hacer para responder a los requerimientos del mercado, por ejemplo (7):

- Escribir un manual de calidad, describiendo el Sistema de Calidad en alto nivel.

- Escribir documentos en forma de procedimientos que describan cómo deben hacer el trabajo en la organización.
- Crear un sistema para controlar la distribución y reedición de documentos.
- Diseño e implantación de un sistema de acciones preventivas y correctivas.
- Identificar las necesidades en cuanto a entrenamiento en la organización.
- Determinar las medidas y equipos para realizar las pruebas.
- Capacitar al personal de la organización en la operación del Sistema de Calidad.
- Planificar y llevar a cabo auditorías internas de calidad.
- Tener en cuenta los requerimientos del estándar con los que no cumple la organización.

La ISO 9000 se divide en cuatro guías o modelos:

- ISO 9001. Modelo de Sistemas de Calidad para el Aseguramiento de la Calidad en el Diseño, Desarrollo, Producción, Instalación y Mantenimiento.
- ISO 9002. Sistemas de Calidad. Modelo de Aseguramiento de la Calidad para la producción, instalación y servicio.
- ISO 9003. Directrices para la aplicación de la norma ISO 9001 para el desarrollo, suministro y mantenimiento de software.
- ISO 9004. Elementos para la Administración y el Sistema de Calidad. Guía para el Sistema de Aseguramiento de la Calidad.
- ISO 9042. Gestión de la Calidad y Elementos de Sistemas de Calidad.

### **1.1.4 Métricas de calidad**

En el glosario de términos y estándares de la IEEE (Instituto de Ingenieros Eléctricos y Electrónicos) se encuentra como definición de métrica: medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado (8).

Las métricas también se conocen como: Una serie de medidas o pasos que ayudan a definir con mayor exactitud el desarrollo y calidad de un producto (9). Son el medio básico para medir, monitorizar, entender, evaluar, controlar y predecir el desarrollo de un software. Identifican eventos, tendencias, comportamiento para la toma de decisiones estratégicas ante los desvíos e incidentes o problemas que surgen durante la ejecución. Medida de alguna propiedad entregable.

Son escalas de unidades para medir un atributo, dígase atributo a las características del producto o del proceso que proporciona información, en el caso de un software se refiere a la disciplina de analizar y recoger datos basándonos en mediciones reales. Herramienta para verificar o medir el estado de un proyecto y para hacer más efectivo el trabajo se decidió agruparlas en: métricas orientadas al software, proyecto, proceso y personas. A continuación se enuncian algunas de las clasificaciones para las métricas de software (10):

**Métricas técnicas:** Centran especial atención en la complejidad lógica y el grado de modularidad del software, miden la estructura del sistema con el objetivo de saber cómo está hecho y no en cómo se obtiene.

**Métricas de calidad:** Proporcionan una indicación de cómo se ajusta el software a los requisitos implícitos y explícitos del cliente.

**Métricas del producto:** Miden el rendimiento del proceso de ingeniería de software, demostrando que tan productivo va a ser.

Las métricas se utilizan para el control de los proyectos en general y no se deben considerar ni estándares ni universales pues son seleccionadas según las necesidades de cada proyecto.

### 1.1.5 Estándares y Modelos de Calidad

La estandarización es toda actividad documentada que norma el comportamiento de un grupo de personas. Los estándares nos dan los medios para que todos los procesos se realicen siempre de la misma forma, mientras nos surjan ideas para mejorarlos. Son una guía para la productividad y la calidad (7). La Ingeniería de Software ha introducido y popularizado una serie de estándares y modelos para medir y certificar la Calidad del Software.

ISO 9126 es un estándar internacional que contiene un modelo de medición y un modelo calidad que permiten llevar a cabo la evaluación de la calidad de productos software. Actualmente el estándar está conformado por 4 partes que dirigen: un modelo de calidad, métricas externas, métricas internas y calidad en las métricas de uso. Agrupa en 6 características, 27 subcaracterísticas de calidad. El estándar provee un buen modelo de referencia para medir la calidad del software (11):

<p><b>a. Funcionalidad.</b> Capacidad del producto software para proporcionar funciones que satisfagan las necesidades especificadas e implícitas.</p> <ul style="list-style-type: none"> <li>➤ Adecuación.</li> <li>➤ Exactitud.</li> <li>➤ Interoperabilidad.</li> <li>➤ Seguridad.</li> <li>➤ Conformidad.</li> </ul>	<p><b>d. Eficiencia.</b> La capacidad del producto software para proporcionar el rendimiento apropiado, relativo a la cantidad de recursos utilizados.</p> <ul style="list-style-type: none"> <li>➤ Tiempo de Respuesta.</li> <li>➤ Utilización de recursos.</li> <li>➤ Conformidad.</li> </ul>
<p><b>b. Fiabilidad.</b> Capacidad del producto software para mantener un nivel especificado de rendimiento.</p> <ul style="list-style-type: none"> <li>➤ Madurez.</li> <li>➤ Tolerancia a fallos.</li> <li>➤ Recuperabilidad.</li> <li>➤ Conformidad.</li> </ul>	<p><b>e. Mantenibilidad.</b> La capacidad del producto software para ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación del software a cambios en el entorno, en los requisitos o en las especificaciones funcionales.</p> <ul style="list-style-type: none"> <li>➤ Analizabilidad.</li> <li>➤ Cambiabilidad.</li> <li>➤ Estabilidad.</li> <li>➤ Facilidad de prueba.</li> <li>➤ Conformidad</li> </ul>
<p><b>c. Usabilidad.</b> La capacidad del producto software de ser entendido, aprendido, utilizado y atractivo al usuario.</p> <ul style="list-style-type: none"> <li>➤ Comprensibilidad.</li> <li>➤ Facilidad de aprendizaje.</li> <li>➤ Operabilidad.</li> <li>➤ Atracción.</li> <li>➤ Conformidad.</li> </ul>	<p><b>f. Portabilidad.</b> La capacidad del producto software de ser transferido de un entorno a otro.</p> <ul style="list-style-type: none"> <li>➤ Adaptabilidad.</li> <li>➤ Facilidad de instalación.</li> <li>➤ Coexistencia.</li> <li>➤ Reemplazabilidad.</li> <li>➤ Conformidad.</li> </ul>

**Figura 2:** Características y subcaracterísticas de calidad del estándar ISO 9126

**Integración de Modelos de Madurez de Capacidades (CMMI):** es un modelo para la mejora de procesos que proporciona a las organizaciones de diferentes sectores los elementos esenciales al ser un modelo integrado y mejorado, es una evolución de CMM. Las mejores prácticas se publican en los documentos llamados modelos y en la actualidad hay dos áreas de interés cubiertas por los modelos de CMMI: Desarrollo (DEV-CMMI) y Adquisición (ACQ-CMMI), posee 22 áreas y está estructurado en cinco niveles de madurez (12):

1. **Inicial.** Es el primer nivel, no es necesario hacer ningún esfuerzo para llegar aquí, las organizaciones en este nivel no disponen de un ambiente adecuado para el desarrollo de software. Aunque se utilicen técnicas correctas de ingeniería, los esfuerzos se ven minados por falta de planificación. Los procesos varían según los individuos, el éxito de los proyectos se basa la mayoría de las veces en el esfuerzo personal, aunque a menudo se producen fracasos y casi siempre retrasos. El resultado de los proyectos es impredecible y está pobremente controlado.
2. **Repetible.** En este segundo nivel se encuentran las empresas en las que existe planificación y seguimiento de proyectos y está implementada la gestión de los mismos. No obstante, aún existe un riesgo significativo de no cumplir las metas.
3. **Definido.** Existe un conjunto establecido de procesos estándar globales bien definidos (estableciendo sus objetivos) dentro de la organización. Existe un sistema de gestión de los proyectos. Una diferencia crítica entre los niveles 3 y 2 de madurez es el alcance de los

estándares, descripciones de los procesos y procedimientos. En el nivel 2 pueden variar entre las distintas instancias de los procesos (entre diferentes proyectos); a nivel 3 son globales dentro de la organización e igual en todas las instancias de cada proceso.

4. **Gestionado.** Se caracteriza por disponer las organizaciones de un conjunto de métricas significativas de calidad y productividad, que se usan de modo sistemático para la toma de decisiones y la gestión de riesgos. El software resultante es de alta calidad.
5. **Optimizado.** La organización completa está volcada en la mejora continua de los procesos. Se hace uso intensivo de las métricas y se gestiona el proceso de innovación: inicial, repetible, definido, administrativo y optimización, cada uno sirve de base para que los demás sean más eficientes y establecer una medida del progreso o avance.

Cada nivel excepto el primero cuenta con un número de Áreas Claves de Proceso que pueden clasificarse en tres tipos: gestión, organizacional e ingeniería.

### 1.2 Pruebas de software

“Las pruebas de software son un elemento crítico para la garantía de la calidad y representan una revisión final de las especificaciones, del diseño y la codificación. La prueba no puede asegurar la ausencia de defectos; solo puede demostrar que existen defectos en el software” (3).

De acuerdo a la IEEE el concepto de prueba se define como: “Actividad en la cual un sistema o componente es ejecutado bajo condiciones específicas, se observan o almacenan los resultados y se realiza una evaluación de algún aspecto del sistema o componente” (13).

La prueba permite obtener la mayor cantidad de errores que traerían un comportamiento inapropiado en el momento de la ejecución. Es el flujo de trabajo más complicado y costoso cuyo propósito es revisar el resultado y validar el software. Sus resultados se observan, se evalúan, se revisan y se registran. Las pruebas son la revisión meticulosa del software, ya sea por partes o de forma general. En ocasiones no se les presta la atención necesaria creyendo que el trabajo ha quedado perfecto o simplemente por falta de tiempo. Por tal motivo se debe planificar y llevar estrictamente su ejecución, en busca de cualquier problema para darle una solución a tiempo.

### 1.2.1 Principios de las pruebas

Los principios muestran el grado de importancia del proceso de prueba en la búsqueda exhaustiva de todo lo que puede convertirse en un defecto en el producto final. Por tal motivo estos constituyen el eslabón fundamental en el aseguramiento de la calidad, a continuación se mencionan algunos:

- Deberían planificarse mucho antes de que comiencen. La planificación de las pruebas debe realizarse tan pronto como esté completo el modelo de requisitos y se tenga consolidado el modelo de diseño (14).
- Para ser más eficaces, deberían ser realizadas por un equipo independiente. El ingeniero de software que creó el sistema no es el más indicado para realizar las pruebas debido a que consciente o inconscientemente puede omitir casos de prueba importantes que conlleven a descubrir nuevos errores. Es recomendable organizar un grupo de trabajo independiente para las mismas que suministre una visión más objetiva del software (9).
- Una buena prueba no puede ser redundante ya que el tiempo y los recursos son limitados (15).
- Los ingenieros de pruebas deben prestarle más atención a las pruebas y crear casos de pruebas con una alta garantía de encontrar errores (15).
- El principio de Pareto es aplicable a la prueba de software. Este principio implica que al 80% de todos los errores encontrados se les puede dar seguimiento hasta un 20% de todos los módulos del programa, el problema sería separar los módulos sospechosos y probarlos concienzudamente (15).

### 1.2.2 Objetivos de las pruebas de software

Los objetivos que deben perseguir todo diseño y ejecución de una Prueba de Software son (14):

- Probar si el software hace lo que debe hacer.
- Probar si el software hace lo que no debe, es decir: si provoca efectos secundarios o adversos.
- Descubrir un error que aún no ha sido descubierto.
- No debe ser redundante, debe encontrar el mayor número de errores con la menor cantidad de tiempo y esfuerzo posible.

- Mostrar hasta qué punto las funciones del software operan de acuerdo a las especificaciones y requisitos del cliente.

### 1.2.3 Tareas a realizar para probar un software

Para ejecutar pruebas de software es necesario (16):

1. **Diseño de las pruebas:** Comprende la identificación de la técnica o técnicas de pruebas que se utilizarán para probar el software, distintas técnicas de prueba ejercitan diferentes criterios como guía para realizar las pruebas.
2. **Generación de los casos de prueba:** Consiste en la confección de los distintos casos de prueba según la técnica o técnicas identificadas previamente. La generación de cada caso de prueba debe ir acompañada del resultado que ha de producir el software al ejecutar dicho caso para detectar un posible fallo en el programa. Los casos de prueba determinan un conjunto de entradas, condiciones de ejecución y resultados esperados para un objetivo particular. Cada técnica de pruebas proporciona unos criterios distintos para generar estos casos o datos de prueba.
3. **Definición de los procedimientos de la prueba:** Conlleva a una especificación de cómo se va a llevar a cabo el proceso, quien lo va a realizar y cuándo.
4. **Ejecución de la prueba:** Es el momento de aplicar los casos de prueba generados e identificar los posibles fallos producidos al comparar los resultados esperados con los resultados obtenidos.

### 1.2.4 Tipos de pruebas de rendimiento

Se realizó un estudio minucioso acerca de las pruebas de software existentes, haciendo énfasis en las que se aplican con el objetivo de medir el rendimiento de las aplicaciones por ser el tema en cuestión.

**Prueba de Rendimiento:** Las pruebas de rendimiento son las pruebas que se realizan desde una perspectiva para determinar lo rápido que realiza una tarea un sistema en condiciones particulares de trabajo (17).

Estas pruebas tienen por objetivo verificar si el rendimiento del sistema está en línea con las expectativas de diseño, sometiéndolo a una carga determinada, en un entorno informático

determinado. Para la ejecución de éstas, se pueden configurar varios tipos de cargas de trabajo para evaluar el sistema bajo diversas condiciones y de esta forma identificar los posibles cuellos de botella que impiden que el sistema pueda satisfacer a los usuarios con el nivel de calidad que se definió en los acuerdos iniciales. Además, las pruebas de rendimiento ayudan a determinar el tiempo requerido para que el software realice una tarea y proporcionan una idea de cuán estable es el sistema (18).

Pueden compararse dos sistemas de acuerdo a los criterios de rendimiento y decir cuál de ellos funciona mejor. Mide el software para encontrar que partes del sistema o de carga influye en el funcionamiento general. Es fundamental alcanzar un alto nivel de rendimiento y para ello se recomienda que las pruebas de rendimiento comiencen en los inicios del proyecto, entre más tarde se encuentre un defecto de rendimiento más difícil es de eliminarlo y el costo es mayor. Pruebas de rendimiento:

- **Benchmark:** Compara el rendimiento de un elemento nuevo o desconocido a uno de carga de trabajo conocido, al realizar la prueba se debe verificar la utilización de recursos y la conformidad.
- **Contención:** Valida que el elemento que se prueba maneje adecuadamente las demandas cuando muchos actores solicitan el mismo recurso (19).
- **Carga:** Es la encargada de probar el funcionamiento del software conectando varios usuarios de forma concurrente buscando los valores límites. Valora y valida la aceptabilidad de un sistema bajo cargas variables, mientras este se mantiene constante.
- **Performance Profile:** Monitorea el tiempo tanto en la ejecución como el acceso a los datos, direccionando los cuellos de botellas y los procesos ineficientes.
- **Spike Testing:** Observa la conducta del sistema al variar el número de usuarios, con cambios drásticos de aumento y disminución de carga.

**Pruebas de requerimientos:** Los requerimientos deben tener una explicación detallada, clara, precisa, coherente y completa de los aspectos que deben estar presentes en el software pues de ello depende la generación de los casos de uso y el análisis de los casos de prueba. La tarea fundamental es lograr la corrección, coherencia y exactitud de los requisitos y para ello se lleva a cabo el proceso de verificación, comenzando con un análisis y una inspección, para luego evaluar la consistencia, completitud y factibilidad de los mismos.

### **1.3 Metodología Proceso Unificado de Desarrollo (RUP)**

Actualmente no existe una metodología universal para hacer frente a un proyecto de desarrollo de forma totalmente exitosa, sino que las existentes se usan para asegurar la calidad de los productos y lograr un mejoramiento continuo de todos los procesos.

Para el presente trabajo es objetivo estudiar el Proceso Unificado de Desarrollo (RUP). Esta metodología unida al Lenguaje Unificado de Modelado (UML) es la más utilizada en el análisis, la implementación y la documentación de sistemas orientados a objetos. Define claramente quién, cómo, cuándo y qué debe hacerse. Su enfoque está basado en modelos y utiliza un lenguaje bien definido para tal fin. El proceso de software propuesto por RUP tiene tres características esenciales: dirigido por Casos de Uso, centrado en la arquitectura y es iterativo e incremental. Todo ello permite establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo (20). De forma organizada designa las tareas y roles en una empresa de desarrollo, se considera el proceso de desarrollo más general de los existentes pues se basa en las mejores prácticas intentadas y probadas. Se divide en cuatro fases: Inicio (define el alcance del proyecto), Elaboración (definición, análisis, diseño), Construcción (Implementación) y Transición (fin). Define una serie de roles para la realización de las actividades concernientes al flujo de trabajo de Prueba: Especialistas en Pruebas (Probador), Analista de Pruebas, Diseñador de Pruebas, Administrador de Pruebas, Ingeniero de componentes, Ingeniero de pruebas de sistema y Probador.

En el proyecto Aduana actualmente no se usa esta metodología sino que se trabaja con el Modelo de Desarrollo, pero este modelo no cuenta con una descripción detallada del flujo de prueba, solo trabaja con el rol de Especialista que es el encargado de coordinar las pruebas a nivel de centro y con Calisoft.

### **1.4 Herramientas utilizadas para la administración y ejecución de pruebas de rendimiento**

Es indispensable el uso de herramientas para mejorar la calidad y eficiencia de las pruebas, encontrar las partes del software que influyen en el mal rendimiento y poder establecer niveles que mantengan un tiempo aceptable de respuesta. Muchas son las herramientas para automatizar las pruebas de rendimiento, a continuación se describen algunas:

**Application Center Test (ACT):** Está diseñada para realizar pruebas de carga y estrés, brindando información sobre problemas de escalabilidad y rendimiento. Es una herramienta de Microsoft incluida en el Visual Studio .NET. Permite gestionar cookies, grabar scripts de prueba de sesiones de Internet Explorer y soporta diferentes esquemas de autenticación. Su funcionamiento consiste en simular gran cantidad de usuarios conectándose a un servidor y realizando múltiples peticiones HTTP.

**Open Load Tester:** Impulsada por IBM Web Sphere, de fácil uso, solución rápida para pruebas de rendimiento. Minimiza el tiempo de trabajo y simplifica los procesos.

**Web Application Stress:** Activa el entorno de prueba mediante la simulación realista de varios exploradores que solicitan páginas desde una aplicación web, y permite grabar una secuencia de comandos mediante el acceso, desde el explorador, a las páginas que se desea incluir en la prueba. La secuencia se puede guardar y ejecutar desde cualquier equipo cliente Windows NT o Windows 2000 que disponga de la aplicación. No es necesario que se disponga del mismo número de equipos clientes que los que tendrá la aplicación de producción, puesto que la herramienta es capaz de simular varios clientes desde cada estación de trabajo (21).

**OpenSTA:** Herramienta de open-source que permite el testeo de estrés de aplicaciones web, está desarrollada bajo estándares de software libre. Es más potente que ACT en cuanto a control y distribución de usuarios.

**JMeter:** Aplicación de escritorio desarrollada en Java para realizar pruebas de rendimiento y dentro de éstas, pruebas de carga. Desarrollada bajo estándares de software libre, es extensible y el usuario puede desarrollar en Java un controlador. Muestra los resultados de las pruebas en una amplia gama de informes y gráficas. Se destaca por ser versátil y estable, por su uso gratuito y la obtención rápida de los cuellos de botella. Puede simular una gran carga en el servidor, multitarea y con grandes facilidades para extender su funcionalidad mediante plugins. Permite simular la conexión simultánea de varios usuarios a una aplicación utilizando múltiples ordenadores. Define un plan de prueba con una estructura de árbol que puede ser utilizada en otras pruebas.

Las tres primeras son herramientas propietarias y su uso es limitado, OpenSTA y JMeter se destacan por estar desarrolladas bajo estándares de software libre, pero se escogió esta última debido a la experiencia alcanzada en cuanto a su uso en la universidad, por ser extensible y

permitir la programación de un controlador en el lenguaje Java, por ser de fácil uso y mostrar los resultados de diferentes formas.

### 1.5 Ingeniería de Requisitos

La ingeniería de requisitos juega el papel principal en el proceso de producción de software. Su objetivo es especificar correctamente el comportamiento del sistema. Es el análisis sistemático de los requisitos determinando la necesidad y condiciones de éstos. Según el glosario de la IEEE requisitos se define como: Condición o capacidad del sistema o componente que satisface un contrato, estándar, especificación u otro documento formal (8).

Los requerimientos se dividen en requerimientos funcionales: son una declaración de una función o característica que debe ser implementada en un sistema, y requerimientos no funcionales: definidos como una restricción o comportamiento que se espera de un sistema, limitación o propiedad emergente del software en desarrollo (22).

Los requerimientos funcionales definen todas las funciones que realizará el sistema. Los requerimientos no funcionales son características que limitan al sistema como: rendimiento, interfaces de usuario, fiabilidad, mantenimiento, seguridad, portabilidad y estándares.

#### 1.5.1 Requisitos no funcionales

Los Requisitos No Funcionales (RNF) aparecen con las necesidades del usuario, definen las restricciones que luego pasarán a ser propiedades del sistema e irán encaminadas a las pruebas. Son propiedades emergentes que incluyen condiciones o cualidades del sistema, por ejemplo: tiempo de respuesta, almacenamiento, confiabilidad, fiabilidad, capacidad de los dispositivos de entrada y salida.

Los RNF son los más difíciles de encontrar, de expresar y de recoger en un documento, su verificación y testeado se hace una actividad complicada y por ello poco cuantificable, llevando a ser evaluados de forma subjetiva. A continuación se resumen algunos requisitos no funcionales que se destacaron por su grado de importancia dentro de la investigación.

**Requisito de Rendimiento y Escalabilidad:** Es fundamental para que el usuario especifique cuáles son sus expectativas en cuanto al tiempo de respuesta.

**Requisito de Software:** Software del que se debe disponer después de implementado el sistema, por ejemplo: Sistema Operativo Windows 95; Máquina Virtual de Java versión 1.3. Depende de la infraestructura instalada y de la plataforma.

**Requisito de Hardware:** Elementos de hardware necesarios para que la aplicación cumpla con todas sus funcionalidades.

**Requisitos de Seguridad:** Es quizás el tipo de requerimiento más difícil, provoca los mayores riesgos pues no solo se refiere a la seguridad del sistema sino la del ambiente en el que se desarrolla, incluye la seguridad física y la administrativa. La seguridad puede ser tratada en tres aspectos diferentes:

- **Confidencialidad:** Se protege la información de acceso no autorizado y de divulgación.
- **Integridad:** La información se protege contra la corrupción y estados inconsistentes mediante mecanismos de chequeo de integridad y realización de auditorías.
- **Disponibilidad:** Le brinda a los usuarios autorizados el acceso a la información deseada sin ser afectados por los mecanismos de seguridad.

**Requisitos de Usabilidad:** Describen los niveles apropiados de usabilidad, se derivan de una combinación de lo que el cliente está tratando de lograr con el producto y lo que los usuarios finales esperan del mismo.

### Requisitos del GINA

El proyecto Gestión Integral de las Aduanas cuenta con un documento de Especificación de Requisitos de Software en su versión 1.0, en él establecen los requisitos funcionales y no funcionales del sistema. Los requisitos de usabilidad y rendimiento se presentan seguidamente (23):

**Usabilidad:** Los sistemas desarrollados deben prestar facilidades de usabilidad que satisfagan las necesidades de todos los usuarios. Estos deben contar con un menú que les permita a los usuarios acceder a las principales funciones que sean de su interés, así como brindarles la posibilidad de interactuar con los diferentes productos sin tener previa preparación, solo con los conocimientos necesarios del negocio.

### Eficiencia

- Tiempo de respuesta por transacción 1.09 segundos como máximo.
- Rendimiento: 1000 transacciones por segundos, cantidad de datos que pueden ser transferidos.
- Capacidad: 80 clientes conectados.
- Utilización de recursos: 18 % memoria y 20 % en disco duro.

### 1.6 Procedimientos para pruebas de rendimiento

Calisoft cuenta con un procedimiento que rige el desarrollo de todas las pruebas a nivel de departamento, dentro de las que se encuentran las pruebas de carga y estrés; este cuenta con actividades como solicitud de pruebas y pruebas exploratorias iniciales, que en el procedimiento que se requiere no son necesarias, además carece de otras como configuración del entorno de las pruebas y comparación de los resultados obtenidos con los requisitos de rendimiento, que si son indispensables en el procedimiento a desarrollar. El departamento cuenta con grupos de prueba que se especializan en un tipo específico pero estos trabajan con una estrategia muy sencilla y adaptada a sus intereses, no con un procedimiento, que necesitaría de gran cantidad de recursos y tiempo, haciendo el proceso de pruebas demasiado complicado para sus objetivos.

Se hizo un estudio y se llegó a la conclusión de que ningún procedimiento contiene todos los criterios que a continuación se describen, los cuales son imprescindibles para aplicar las pruebas de rendimiento al módulo Tablas de Control (TC).

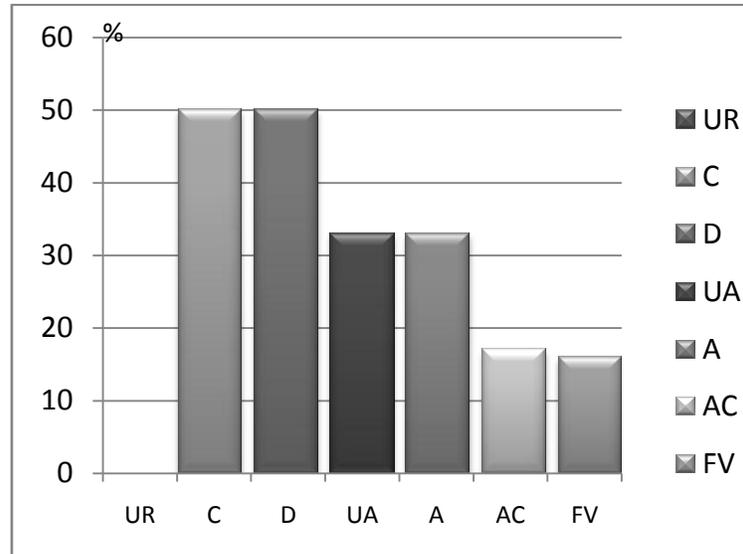
Criterios	Descripción
Uso de requisitos (UR)	Los casos de prueba se deben crear a partir de la descripción de los requisitos de software y no de los casos de uso.
Comparación (C)	Se debe hacer una comparación entre los resultados obtenidos en las pruebas y los requisitos de rendimiento.
Dependencia (D)	El procedimiento descrito no debe estar unido al uso de otra herramienta que no sea JMeter.
Uso de artefactos (UA)	Se deben describir los artefactos que entran y salen de cada actividad propuesta.
Actividades (A)	Las actividades deben organizarse de forma lógica en cada una de las fases.
Análisis de la carga	Para lograr una correcta configuración del entorno de las pruebas es

(AC)	necesario analizar la carga a simular.
Forma de validación (FV)	Debe estar validado por alguno de los métodos existentes y no con la aplicación de este.

Los resultados arribados fueron los siguientes:

Procedimientos	Criterios						
	UR	C	D	UA	A	AC	FV
Procedimiento para diseñar e implementar pruebas no funcionales en los proyectos de la UCI (24).		X					
Prueba Automática de Carga y Estrés en el Proyecto CICPC (25).		X	X	X			
Procedimiento para pruebas de rendimiento de Carga y Estrés al Sistema Único de Aduanas (26).					X	X	
Procedimiento para pruebas de software con herramientas automatizadas en el Departamento de Pruebas de Software (27).				X			
Automatización de pruebas de carga y estrés del Sistema de Información de Perforación de Pozos Petroleros (SIPP) (28).		X	X				X
Pruebas de rendimiento y optimización de Aplicaciones J2EE basadas en la Web (29).			X		X		

X: significa que cumple el criterio.



**Figura 3:** Resultados del análisis a los procedimientos

En el documento Evaluación de Generación de Carga en Entornos Virtualizados para Pruebas de Rendimiento de Software (Evaluating Load Generation in Virtualized Environments for Software Performance Testing), que se encuentra en una de las bases de datos referenciales de la biblioteca se propone un procedimiento simple, cuenta con cinco secciones: configurar la prueba y activar el generador, aplicar la prueba de carga, utilizar herramienta de automatización y vigilar los recursos, recopilar los resultados y analizar los resultados; todas estas actividades las realiza el ingeniero de prueba. Este procedimiento plantea simular con la herramienta Load Runner que permite ejecutar cuatro pasos fundamentales: creación de los script de prueba, definición del escenario o configuración del entorno de la herramienta (teniendo en cuenta los parámetros de entrada), ejecución del escenario y evaluación de la prueba. En la última actividad se describen los requisitos de software y de hardware, se comparan los resultados obtenidos al utilizar una máquina física y los obtenidos en una virtual (30). Este procedimiento no se recomienda ya que no cumple los criterios analizados anteriormente.

### 1.6.1 Método para validar procedimientos

Para la validación de procedimientos el método Delphi se considera confiable, eficaz y sistemático y por ello se escogió. Este consiste en recopilar la opinión de un grupo de expertos (personas con bastas experiencias en un tema específico) utilizando encuestas que recojan su

experiencia intuitiva y lógica, sus grados de competencia y sus valoraciones individuales, luego los resultados se analizan y se tienen presentes al evaluar el procedimiento. El método cuenta con dos características fundamentales:

**Anonimato:** Ningún experto conoce la identidad de los demás miembros del grupo, evitando la influencia en los resultados de las encuestas.

**Respuestas estadísticas:** Se basa en la opinión común de los expertos y los resultados utilizando métodos estadísticos, se convierte la información obtenida en un arma valiosa.

### Conclusiones

Luego de analizar los diferentes estándares, métricas y normas, así como estudiar varios tipos de pruebas con sus objetivos y principios, se decidió para solucionar el problema señalado aplicar pruebas de carga y estrés que verificarán el comportamiento del requisito no funcional de rendimiento en el módulo Tablas de Control. Después de profundizar en las metodologías utilizadas actualmente para guiar el desarrollo de software se escogió RUP por ser la de mayor experiencia y contar con las mejores prácticas intentadas y probadas, por organizar y plantear un grupo de roles, artefactos, fases y actividades para hacer más fácil el desarrollo de las pruebas.

Se estudiaron varias herramientas existiendo competencia entre ACT, OpenSTA y JMeter, decidiendo que la última es la más óptima para realizar las pruebas por ser una aplicación de escritorio para pruebas de rendimiento bajo estándares de software libre, por ser extensible y mostrar los resultados en una amplia gama de gráficos e informes.

Muy pocas empresas en el mundo le dan el grado de importancia que lleva el proceso de pruebas de software y no realizan un procedimiento para probar los requisitos de rendimiento. Para implementar dichas pruebas se necesita de una definición detallada de las tareas y los objetivos a cumplir, con una selección de trabajadores capacitados que desarrollen actividades de forma organizada y suministren a las pruebas un alto grado de calidad. Se hizo un estudio de varios procedimientos, concluyendo que ninguno se adapta a las necesidades del módulo y por ello en el próximo capítulo se desarrollará un procedimiento que cumpla con todos los criterios analizados anteriormente.

# Capítulo 2

## 2. Descripción del Proceso de Pruebas

### Introducción

En este capítulo se desarrolla un procedimiento para garantizar la organización del proceso de aplicación de las pruebas adaptado a las necesidades y características del módulo Tablas de Control. Se describen detalladamente todas las actividades a realizar, facilitando y organizando el trabajo, se especifica el alcance, los objetivos y las tareas a realizar por roles.

### 2.1 Resumen

Para comprobar si el producto final cumple con los requisitos de rendimiento planteados inicialmente, es necesario desarrollar una guía que trace el camino a seguir para el desempeño correcto de todas las actividades de prueba o una secuencia de pasos para realizar todos los casos de prueba de forma automática y cumplir los objetivos propuestos distribuyendo todas las responsabilidades.

### 2.2 Nombre del procedimiento

Procedimiento para comprobar el requisito no funcional de rendimiento en el módulo Tablas de Control con las pruebas de Carga y Estrés.

### 2.3 Alcance

Comprobación del requisito no funcional de rendimiento en la solución Tablas de Control de la AGR.

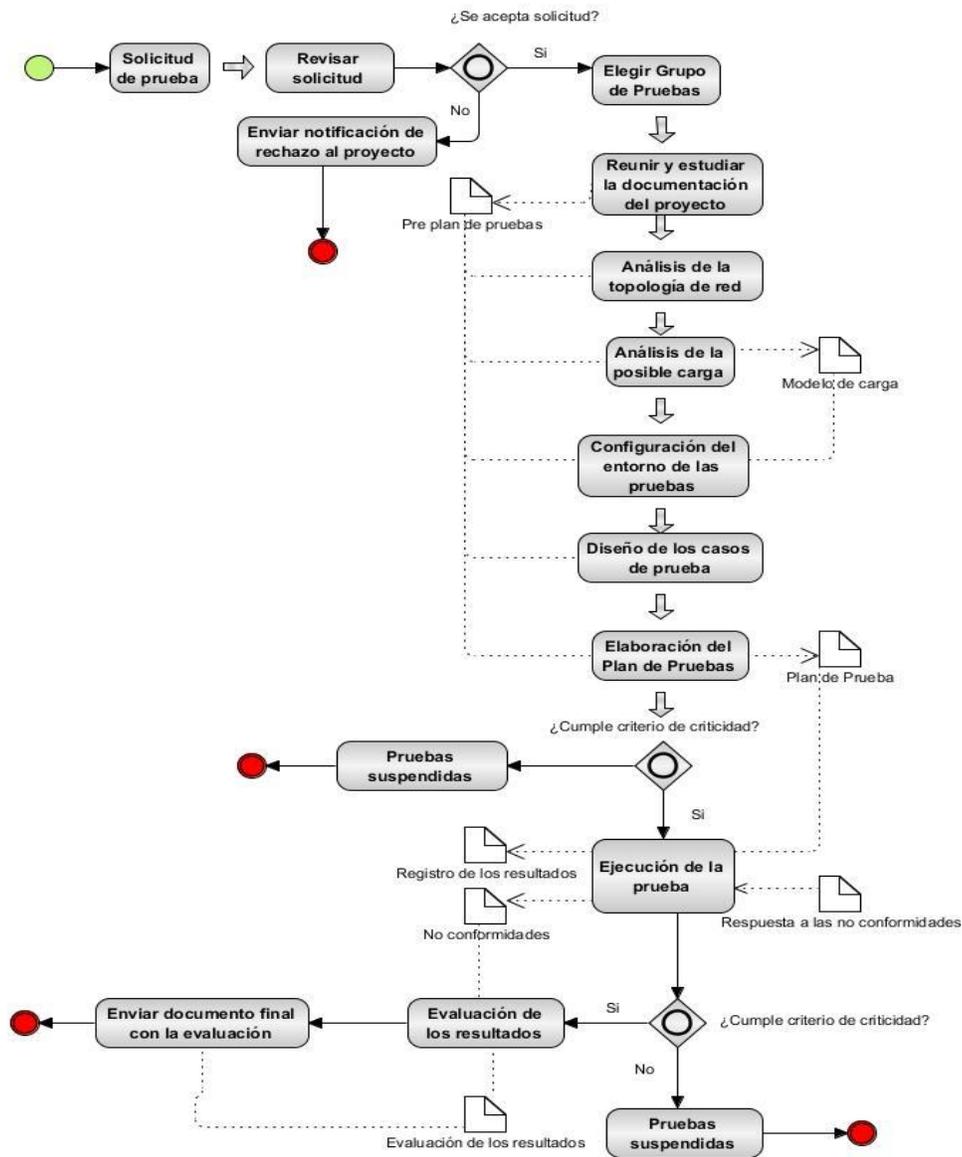
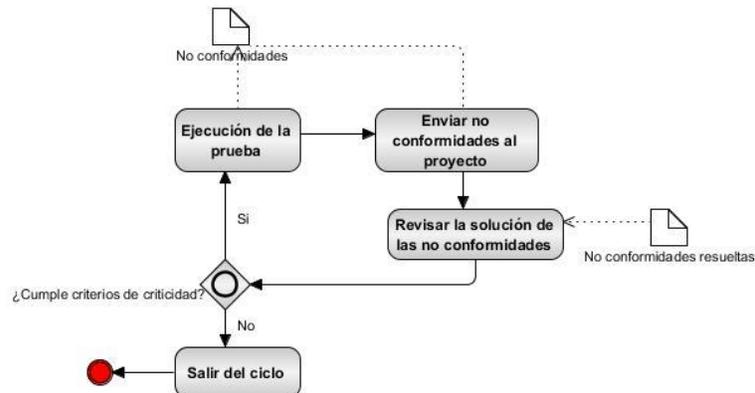


Figura4: Descripción del procedimiento.



### 2.4 Selección de roles

Las pruebas serán realizadas por un equipo seleccionado por el jefe de proyecto, teniendo en cuenta: personal con conocimiento de la aplicación y de todos los aspectos de la ingeniería de software, habilidades de planificación, análisis y gestión, así como experiencia anterior en la programación. Cada miembro puede desempeñar uno o varios roles y viceversa, el rol es el encargado de organizar las responsabilidades al realizar las tareas. Se decidió usar en el desarrollo de las pruebas de rendimiento al módulo Tablas de Control cuatro roles fundamentales:

**Jefe de prueba o Administrador:** Es el máximo responsable del éxito de las pruebas, planifica y dirige los recursos, encuentra solución a todos los problemas que aparecen en el transcurso de las pruebas. Desarrolla:

- **Plan de Prueba:** Este artefacto guarda los objetivos y enfoques, describe los recursos necesarios y la misión del proceso, define la configuración del entorno, le da estructura a la implementación, identifica ideas y motivaciones. Puede ser modificado por el Analista y el Diseñador de las pruebas.
- **Resumen de evaluación de Prueba:** Es la recopilación organizada de los principales resultados de las pruebas, que pueden ser revisadas y evaluadas por el analista, sirve de experiencia para procesos de pruebas futuros.

**Analista de prueba:** Es el encargado de identificar las técnicas y herramientas, supervisa el proceso y los resultados de cada ciclo, es la representación del cliente que no tiene participación directa. Es el responsable de definir:

- **Casos de prueba:** Evalúan aspectos esenciales para el destino de las pruebas que constituyen entradas, condiciones a tener en cuenta en la ejecución y resultados esperados, incluyen un subconjunto de requisitos o características de rendimiento.
- **Modelo de análisis de carga:** Modelo que define uno o más perfiles del área de trabajo, estudiando las condiciones del entorno de prueba para crear una carga de trabajo realista que permita precisar los riesgos de rendimiento existentes.
- **Resultados de las pruebas:** Descripción resumida del análisis de los registros de las pruebas y valoración detallada de la calidad. La herramienta JMeter simplifica este documento brindando informes con los resultados.

**Diseñador de pruebas:** Tiene la responsabilidad de orientar el enfoque de las pruebas para garantizar la implementación satisfactoria de éstas, incluyendo la aplicación de las técnicas y herramientas identificadas. Se responsabiliza de la construcción de los siguientes artefactos:

- Diseño de prueba: Describe todos los elementos estructurales y casos de prueba, así como las actividades de implementación.
- Estrategia de pruebas: Especifica cómo se desarrollará la prueba, la estimación del tiempo a emplear, las personas involucradas y lo que hará cada una.
- Especificación del Entorno de Pruebas: Configuración general del entorno de pruebas con la descripción de los requisitos de Software y de Hardware.

**Probador:** Es el responsable de desarrollar y registrar las pruebas, se conoce también como verificador, implementa y ejecuta las aplicaciones de prueba. Realiza los siguientes artefactos:

- Script de pruebas: En el caso de las pruebas automatizadas representa instrucciones descritas paso a paso que son leídas por la herramienta y que habilita la ejecución de las pruebas.
- Registro de pruebas: Es la descripción detallada de la ejecución del conjunto de aplicaciones que sirve de constancia de que se hizo la prueba y se obtuvieron resultados, constituye un medio auditable y muestra los tiempos empleados que posibilitan llegar a concepciones futuras del éxito o no de las mismas.

### 2.5 Objetivos

Los roles mencionados anteriormente tienen como actividad fundamental realizar el ciclo completo de pruebas de rendimiento para concluir si el software cumple con los patrones propuestos, para ello utilizarán el procedimiento con los objetivos de:

- Verificar si el sistema cumple con los requisitos no funcionales de rendimiento.
- Verificar que la aplicación soporta la carga de trabajo necesaria.
- Buscar los límites en el funcionamiento de la aplicación ejecutándose con un número superior de usuarios al esperado.
- Determinar el número máximo de usuarios concurrentes, el tiempo medio de respuesta a una petición y el número de transacciones por segundo.

### 2.6 Distribución de las tareas por roles

#### **Administrador de prueba:**

- Monitorización del proceso de prueba.
- Estudio de la documentación del proyecto.
- Planificación y dirección de los recursos.
- Validación de los objetivos y el enfoque de la prueba.
- Evaluación del resultado de las pruebas.
- Gestión y validación de las no conformidades.

#### **Analista de pruebas:**

- Estudio de la documentación existente del proyecto.
- Definición de necesidades, detalles, destinos e ideas.
- Determinación y análisis de los resultados.

#### **Diseñador de las pruebas**

- Estudio de la documentación existente del proyecto.
- Especificación de las configuraciones del entorno de prueba.
- Descripción de las técnicas y los casos de prueba.

#### **Probador**

- Estudio de la documentación existente del proyecto.
- Ejecución e implementación de las aplicaciones de prueba.
- Detecta las no conformidades y analiza las anomalías.
- Creación del documento final de las pruebas con los resultados registrados.

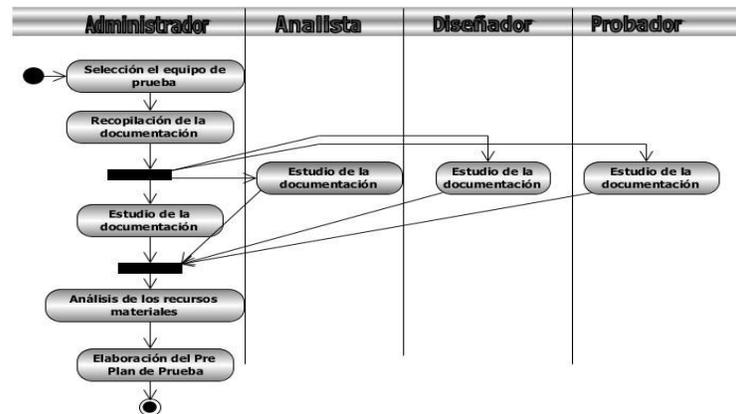
### 2.7 Fases a desarrollar en el proceso de pruebas

Primeramente se seleccionaron los roles, luego los objetivos para enfocar el trabajo de estos cuando efectúen las tareas señaladas y se organizaron las actividades en cuatro fases:

- Planificación de las pruebas.
- Diseño de las pruebas.

- Ejecución de las pruebas.
- Evaluación de los resultados.

### 2.7.1 Planificación de las pruebas



**Figura 5:** Actividades de la fase de planificación

Constituye la base de todo el proceso y asegura el éxito de las pruebas, para ello es indispensable que todo el equipo este en función de una buena planificación. En esta fase cae el mayor trabajo sobre el administrador de prueba aunque debe participar todo el equipo de calidad junto a los analistas y desarrolladores del proyecto. Se organiza y se prepara el proceso de pruebas de rendimiento, se hace un estudio de la aplicación y se crea una estrategia de trabajo. Estas son solo algunas de las actividades, otras de gran importancia se describen seguidamente:

#### 2.7.1.1 Selección del equipo de prueba

El jefe de proyecto es el encargado de elegir el administrador de pruebas, luego este selecciona su equipo, que debe contar con conocimiento de la aplicación y de todos los aspectos de la ingeniería de software, habilidades de planificación, análisis y gestión, así como experiencia anterior en la programación, teniendo presente las habilidades de cada uno se procedería a distribuir los roles.

#### 2.7.1.2 Estudio de la documentación reunida del proyecto

Luego de organizado el equipo se recoge el expediente del software y entre todos se hace un estudio minucioso de la documentación del módulo que incluye como *artefactos de entrada*:

- ✓ Manual de Usuario.

- ✓ Plan de Aseguramiento de la Calidad.
- ✓ Documento de Especificación de los Requisitos de Software.

En este paso se debe profundizar en el análisis de los requisitos de rendimiento, para ello se planifica un encuentro con los funcionarios de la AGR para explicar las principales funcionalidades de la aplicación. Luego de aclarar todos los requerimientos se comienza a trabajar en el desarrollo de los casos de pruebas y en la configuración del Pre Plan de Pruebas. En este documento se deben dejar plasmadas todas las especificaciones de las pruebas, incluyendo la validación de los requerimientos y las funcionalidades más importantes, se establecen los mecanismos para recoger y analizar los resultados, controla todo el proceso de prueba, organizando las actividades y documentándolas. El jefe de pruebas es el máximo responsable de su confección pero se recomienda la participación de todo el equipo. En las etapas siguientes se le harán transformaciones y pasará a ser el Plan de Prueba. Esta actividad tiene como artefacto de salida el Pre Plan de Pruebas.

### 2.7.1.3 Análisis de los recursos materiales

Es fundamental analizar la red del lugar donde se aplicarán las pruebas ya que las características de la topología influyen directamente en los resultados de las mismas. Se sugiere ver también las ventajas y desventajas de cada componente de la red. En el desarrollo de unas pruebas se trabajará con una computadora pero en otras se usarán varias, simulando así la conexión de todos los módulos del proyecto Aduana al módulo TC.

### 2.7.2 Diseño de las pruebas

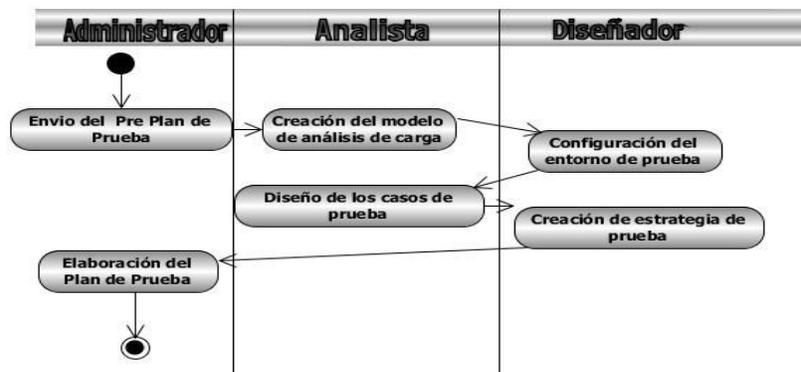


Figura 6: Actividades de la fase de diseño

El rol de mayor fuerza en esta fase es el de diseñador de pruebas que se encarga de configurar el entorno y debe contar con el apoyo de todo el equipo de calidad, por ejemplo: el jefe de prueba debe comenzar a circular el documento de las no conformidades, debe reunirse con todo el equipo para revisar el procedimiento, los objetivos y demás elementos importantes vistos en la planificación, además de señalar los pasos para obtener los casos de prueba y los scripts.

### **2.7.2.1 Creación del Modelo de análisis de carga**

El modelo que seguidamente se presenta constituye un artefacto de salida en el diseño de las pruebas, es sumamente necesario pues identifica varios perfiles de carga que influyen en la definición del entorno operativo donde se evaluará el software. Los perfiles o pasos señalados, surgen a partir de las características iniciales y las estadísticas del usuario.

*Estudio inicial:* Para llegar a definir una carga de trabajo realista que valore con exactitud el riesgo de rendimiento y no a conclusiones erróneas, es necesario hacer un estudio previo de los valores de rendimiento que el sistema debe cumplir. Las necesidades de rendimiento que se establecen tienen como objetivo la determinación de los valores reales con los cuales el sistema debe trabajar sin problemas.

*Selección de las variables del sistema:* La interacción con el sistema y el rendimiento de este se ven afectados por factores como el sistema operativo, navegadores, velocidad de acceso y ancho de banda.

*Selección de los atributos del actor:* Los actores del sistema pueden ser usuarios normales o usuarios simulados por una herramienta y se toman un tiempo interactuando con el sistema. Este tiempo se tiene en cuenta al analizar la carga ya que no causa el mismo efecto un usuario que conoce el sistema y trabaja cargando constantemente que uno que lo hace casualmente.

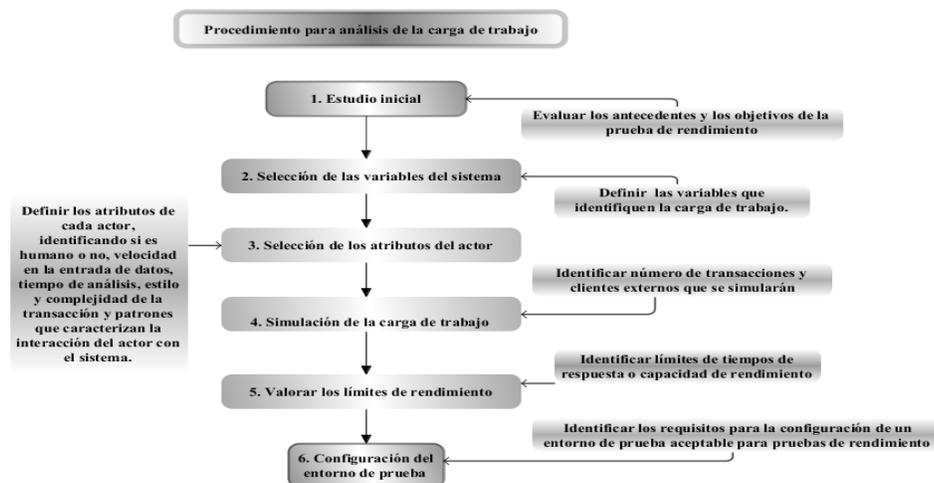
La forma en que los usuarios entran al sistema también influye en los resultados de la carga pues no provoca el mismo impacto una cantidad de usuarios conectándose escalonadamente que todos al mismo tiempo.

Cuando se conectan usuarios concurrentes es muy difícil que coincidan en un determinado punto o enlace, ya que la actividad de estos es generalmente asíncrona, no obstante es necesario valorar esa posibilidad cuando se modelen los usuarios virtuales en la herramienta.

*Simulación de la carga de trabajo:* En este paso se debe definir el número de transacciones y de clientes externos que se simularán, así como los valores reales con los que trabajará el sistema para cumplir las necesidades de rendimiento.

*Valorar límites de rendimiento:* Permite obtener la capacidad de rendimiento.

*Configuración del entorno de prueba:* Como actividad final, luego de estudiar y definir varios perfiles, se pasará a configurar un entorno de pruebas que de soporte al esfuerzo de éstas, especificando disposiciones de software, hardware y de configuración. Esta actividad posibilita construir un entorno controlado donde se realizarán las actividades y se obtendrán resultados válidos.



**Figura 7:** Procedimiento para análisis de la carga

### 2.7.2.2 Configuración del entorno de pruebas

*Artefacto de entrada:* Modelo de análisis de la carga de trabajo.

Para la aplicación de las pruebas se configura un entorno lo más parecido posible al lugar donde se explotará la aplicación para obtener resultados realistas. Un entorno de prueba bien controlado es un aspecto importante para la resolución de anomalías y errores, en su diseño debemos tener presente especificaciones de hardware y de software:

Para las pruebas de rendimiento es fundamental contar con los requerimientos de las PC clientes, el principal elemento a vigilar es la memoria física y se debe trabajar con los resultados del análisis de la carga pues entre más usuarios virtuales se generen, mayor carga de procesos

existirá y mayor memoria RAM será consumida, además de que el sistema operativo y JMeter ocuparán como mínimo 64 Megas.

JMeter como las demás herramientas de prueba recogen una amplia gama de informes que son guardados en el disco duro, de ahí la necesidad de vigilar constantemente el espacio que se va ocupando mientras se realizan las pruebas. Los generadores de carga deben disponer de más de 10 GB de espacio libre en el disco. Otro elemento relevante es el uso del CPU, recomendando mantenerse siempre por debajo del 75 %.

En las regulaciones de software se señala velar por el Sistema Operativo, gestor de Base de Datos, una copia de la aplicación y demás herramientas necesarias para las pruebas.

### **2.7.2.3 Diseño de los casos de prueba**

*Artefactos de entrada:* Documento de Especificación de los Requisitos de Software.

Los casos de prueba son el punto esencial para la ejecución de las pruebas, describen un conjunto de entradas y resultados esperados que constituyen prioridades de la aplicación, identifican los scripts de prueba y proporcionan un esquema de implementación de éstos. Son condiciones que especifican una idea de prueba, utilizan un subconjunto de requisitos de rendimiento y riesgos que pueden afectar al producto. Durante las pruebas en la herramienta estos se recogerán en un Plan de Pruebas que la aplicación creará, especificando la dirección IP del servidor, el puerto por el que se va conectar, el protocolo, el método para la petición HTTP y las direcciones URL de las páginas.

Se van a efectuar pruebas complejas pues se introducirán datos al sistema y serán enviados a la base de datos, siendo indispensable que los casos de pruebas ayuden en la comprensión fácil de éstas. Se sugiere diseñar una plantilla acorde a las necesidades del módulo donde se recoja un resumen general de lo que se quiere con el caso de prueba, condiciones de ejecución, secciones a probar y registro de deficiencias encontradas. Para el diseño de los casos de prueba se utilizará la plantilla definida en el Procedimiento para la realización de pruebas de rendimiento de Carga y Estrés al Sistema Único de Aduanas del año 2009, confeccionada para la herramienta JMeter.

Se hace una breve descripción de los elementos que conforman esta planilla:

Fecha	Versión	Descripción	Autor
dd/mm/aaaa	V 1.0	Descripción de los casos de prueba	

*Funcionalidades a probar:*

Se escribirán las funcionalidades del sistema que serán probadas (ventanas que serán visitadas dentro de la aplicación). En la parte Comentario se pondrán detalles de lo que realiza la funcionalidad.

Funcionalidad(es)	Comentario

*Variables:*

Número de usuarios concurrentes (Número de Hilos en la herramienta): Variable que establece la cantidad de usuarios conectados simultáneamente.

Tiempo entre conexión y conexión (Período de subida en segundos): Variable que representa el tiempo que ocurrirá entre las conexiones de los usuarios definidos en Número de Hilos.

Número de iteraciones (Contador del Bucle): Esta variable es la encargada de recoger la cantidad de veces que el usuario desea que se conecte el grupo de hilos definido.

Variables	Valores
<b>Número de usuarios concurrentes</b>	
<b>Tiempo entre conexión y conexión</b>	
<b>Número de iteraciones</b>	

*Artefacto de salida:* Diseño de los Casos de prueba.

#### 2.7.2.4 Elaboración del Plan de Pruebas

*Artefacto de entrada:* Documento de Especificación de los Requisitos y Pre Plan de Pruebas.

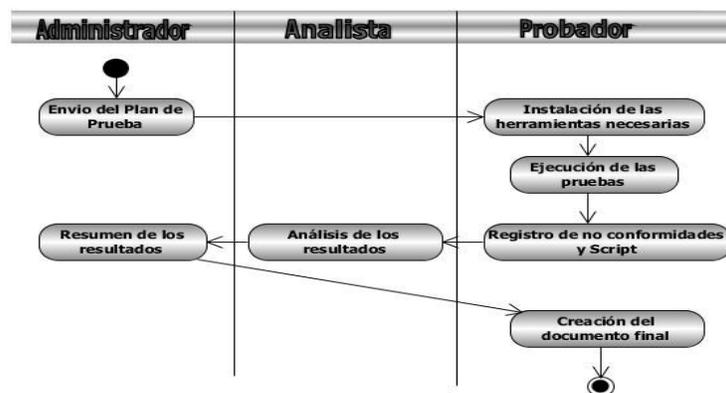
Constituye el documento resumen de esta fase, describiendo: quién, cómo, cuándo y dónde se realizarán las actividades de ejecución, haciendo alusión a la estrategia de pruebas. Este debe incluir el alcance, calendario, procedimientos a seguir para el trabajo con las no conformidades, análisis de los riesgos y aspectos que el equipo de prueba estime necesario para fortalecer el proceso. Se considera el medio de comunicación entre el equipo de prueba y el cliente, por lo que deben evitarse detalles poco relevantes y de difícil comprensión.

El Plan de Pruebas debe contar con:

- Objetivo.
- Alcance.
- Roles y responsabilidades
- Recursos del sistema
- Artefactos
- Evaluación de las pruebas
- Criterios de Criticidad.

*Artefacto de salida:* Plan de Prueba.

### 2.7.3 Ejecución de las pruebas



**Figura 8:** Actividades fase de ejecución

*Artefacto de entrada:* Estrategia de Prueba y Plan de Prueba.

En la etapa de planificación se hizo un estudio detallado de la aplicación abriendo camino al diseño de los casos de prueba. Luego se configuró el entorno, el próximo paso sería prepararlo, instalando el software y cumpliendo con las restricciones de hardware señaladas. Seguidamente

se ejecutan las pruebas con la ayuda del JMeter, se registran los resultados y las no conformidades.

### 2.7.3.1 Registro de los resultados

Para el registro de los resultados se utilizarán las plantillas diseñadas en el Procedimiento para pruebas de rendimiento de Carga y Estrés al Sistema Único de Aduanas:

Resultados generales:

Petición	Muestras	Media	Mediana	Min	Max	Línea 90 %	%Error	Rendimiento	Kb/s

Lista de links con errores:

Lista de links con errores (En caso de existir):	Comentario

### 2.7.3.2 Gestión de las no conformidades

*Artefactos de entrada:* Producto de Software y Plan de Pruebas.

Al ejecutar las pruebas aparecen errores que constituyen no conformidades; al reportarlas se debe considerar el caso de prueba que las originó y la repercusión que tiene en el funcionamiento de la aplicación. Se pueden clasificar en significativas o no significativas y al gestionarlas deben quedar registradas en un documento final con el siguiente formato:

No. de la NC	Id del Caso de Prueba	Script	Descripción	Clasificación	Prioridad	Responsable

- No. de la NC: identificador de la no conformidad (NC).
- Id del Caso de Prueba: identificador del caso de prueba donde se encontró la no conformidad.
- Script: instrucción para reproducir la NC.
- Descripción: breve descripción de la NC.
- Clasificación: significativa o no significativa.
- Prioridad: baja, media, alta o crítica.

- Responsable: nombre de la persona que reporta la NC.

Artefactos de salida: Script de pruebas y Registro de prueba.

#### 2.7.4 Evaluación de los resultados



**Figura 9:** Actividades de la fase análisis de las pruebas

*Artefactos de entrada:* Plan de Pruebas, Script y Registro de prueba.

Constituye la fase final del procedimiento descrito; dónde se analizarán los resultados, se hará un resumen de los mismos y se creará el informe final. Al evaluar el software se debe hacer una comparación entre los RNF de rendimiento y los resultados obtenidos en la pruebas, para concluir si las pruebas fueron satisfactorias y conocer si el grado de calidad del sistema es aceptable. Se debe identificar el origen de cada problema y el grado de afectación de estos para trazar un grupo de acciones y resolverlos inmediatamente. En esta etapa se decide si el Plan de Pruebas sufrirá cambios, contemplando nuevos casos de prueba. La evaluación del sistema debe contar con el tiempo que demora cada transacción y la posibilidad de mejorarlos. *Artefactos de salida:* Documento final con la evaluación de los resultados.

#### 2.8 Validación del procedimiento

Se decidió validar el procedimiento descrito utilizando el método Delphi. Siguiendo las tres etapas propuestas por él, primeramente se comenzó con la selección del grupo de expertos teniendo en cuenta los siguientes criterios: graduado de ingeniería informática, más de dos años de experiencia laboral, conocimientos de aplicaciones web, vinculado al desarrollo de proyectos, conocimiento del tema de calidad y vinculado al proceso de pruebas de software. Luego se elaboraron y se aplicaron los cuestionarios para medir los coeficientes de conocimiento y de

valoración, escogiéndose para la validación del procedimiento cinco expertos. Se hizo necesario calcular el coeficiente de competencia de los expertos para ello se utilizó la fórmula:

$K = \frac{1}{2} (k_c + k_a)$  Donde  $K_c$ : es el coeficiente de conocimiento y  $K_a$ : es el coeficiente de argumentación. La interpretación de los coeficientes de competencias es la siguiente:

- Si  $0,8 < k < 1,0$  Coeficiente de competencia alto.
- Si  $0,5 < k < 0,8$  Coeficiente de competencia medio.
- Si  $k < 0,5$  Coeficiente de competencia bajo.

El coeficiente de conocimiento se obtuvo de la primera pregunta de la encuesta aplicada, recoge una autoevaluación del experto donde marca con una x la casilla que refleje su conocimiento acerca del tema evaluado en una escala del 1 al 10. Luego el valor de cada casilla se multiplicará por 0.1 para ajustarlo a la teoría de las probabilidades

1	2	3	4	5	6	7	8	9	10

El coeficiente de argumentación se obtuvo de la segunda pregunta del cuestionario que recoge las fuentes que le sirven para argumentar su conocimiento acerca del tema tratado.

	Fuentes de argumentación	Grado de influencia		
		Bajo(B)	Medio(M)	Alto(A)
P1	Análisis teóricos realizados por Ud.			
P2	Su experiencia obtenida.			
P3	Trabajos de autores nacionales.			
P4	Trabajos de autores internacionales.			
P5	Su propio conocimiento del estado del problema.			
P6	Su intuición.			

La casilla seleccionada en la tabla anterior se lleva a puntos según la escala siguiente:

Fuentes de argumentación	Grado de influencia		
	Bajo	Medio	Alto

Análisis teóricos realizados por Ud.	0.1	0.2	0.3
Su experiencia obtenida.	0.05	0.05	0.05
Trabajos de autores nacionales.	0.1	0.2	0.25
Trabajos de autores internacionales.	0.1	0.2	0.3
Su propio conocimiento del estado del problema.	0.05	0.05	0.05
Su intuición.	0.05	0.05	0.05

Obteniendo los resultados que a continuación se presentan:

Expertos	P1	P2	P3	P4	P5	P6	Ka	Kc	K	Competencia
E1	0.2	0.05	0.25	0.2	0.05	0.05	0.8	0.8	0.8	Alto
E2	0.2	0.05	0.25	0.3	0.05	0.05	0.9	0.8	0.85	Alto
E3	0.2	0.05	0.25	0.3	0.05	0.05	0.9	0.8	0.85	Alto
E4	0.2	0.05	0.25	0.3	0.05	0.05	0.9	0.8	0.85	Alto
E5	0.2	0.05	0.2	0.2	0.05	0.05	0.75	0.7	0.73	Medio

La encuesta aplicada se puede ver en el Anexo 1. Es necesario analizar la concordancia entre los expertos y la fórmula planteada por Kendall para determinar el Coeficiente de concordancia, el cual es:

$$W = \frac{12s}{k^2(N^3 - N)} ; \quad S = \sum_{j=1}^n (S_j - \bar{S})^2 \text{ donde } \bar{S} = \frac{\sum_{j=1}^n S_j}{N}$$

S: Suma de los cuadrados de las desviaciones observadas de la media de los rangos.

N: Número de entidades, cantidad de preguntas realizadas a los expertos.

$\bar{S}$ : Suma de los rangos dividido entre la cantidad de preguntas realizadas.

K: Cantidad de expertos.

W: Concordancia entre los expertos.

A las categorías se le asignaron valores: Categoría máxima (5), Categoría media (3) y Categoría Baja (1), mostrándose los valores a continuación:

Preguntas	Expertos					
	E1	E2	E3	E4	E5	Sj

P1	5	3	5	3	3	19
P2	5	3	3	3	3	17
P3	5	3	5	3	3	19
P4	5	3	5	3	3	19
P5	5	3	5	5	3	21
P6	5	3	5	3	3	19
P7	3	3	3	3	3	15
P8	5	3	5	3	3	19
P9	5	3	5	5	3	21

Calculando W

$$\bar{S} = \frac{\sum_{j=1}^n S_j}{N} \quad \text{Donde } N=9, \bar{S} = 169/9 = 18.8.$$

$$S = \sum_{j=1}^n (S_j - \bar{S})^2 = 0.04 + 3.24 + 0.04 + 0.04 + 4.84 + 0.04 + 14.44 + 0.04 + 4.84 = 27.56.$$

Luego  $K=5$  y  $N=9$ ,  $W = \frac{12s}{k^2(N^3 - N)}$   $W = 330.72 / 18000 = 0.01837$  este valor expresa el grado de concordancia entre los 5 expertos al emitir la evaluación de las preguntas, siempre debe estar entre 0 y 1.

Ahora se debe calcular Chi Cuadrado Real utilizando  $X^2 = K(N-1)W = 0.7348$ . Si  $X^2_{\text{real}} < X^2_{(\infty, N-1)}$  entonces existe concordancia entre los expertos, donde  $\infty=0.05$  para un nivel de confianza del 95% entonces  $0.7348 < X^2(0.05, 8)$ ,  $0.7348 < 15.5073$ . Concluyendo que existe concordancia entre los expertos.

### Exploración de los resultados

Los resultados de los cuestionarios se recogen en las siguientes tablas:

Tabla de frecuencias absolutas				
Preguntas	MA	A	NA	Total
1	2	3	0	5

2	1	4	0	5
3	2	3	0	5
4	2	3	0	5
5	4	1	0	5
6	2	3	0	5
7	0	5	0	5
8	2	3	0	5
9	3	2	0	5

**Primer Paso:** Se construye la tabla de frecuencias acumuladas, sumando escalonadamente los valores de la tabla por filas, se elimina la columna total.

Tabla de frecuencias absolutas acumuladas			
Preguntas	MA	A	NA
1	2	5	5
2	1	5	5
3	2	5	5
4	2	5	5
5	4	5	5
6	2	5	5
7	0	5	5
8	2	5	5
9	3	5	5

**Segundo Paso:** Se copia la tabla anterior y se borran los resultados, se vuelve a llenar la tabla con las frecuencias relativas acumuladas, las cuales se obtienen dividiendo los valores de frecuencia absoluta acumulada entre el número total de expertos.

Tabla de frecuencias absolutas acumuladas			
Preg.	MA	A	NA
1	0.4	1	1
2	0.5	1	1
3	0.4	1	1
4	0.4	1	1
5	0.8	1	1

6	0.4	1	1
7	0	1	1
8	0.4	1	1
9	0.6	1	1

**Tercer Paso:**

Se copia la tabla anterior y se elimina la columna NA.

						<b>N =</b>	<b>0.71</b>
<b>Puntos de corte</b>							
<b>Preguntas</b>	<b>MA</b>	<b>A</b>	<b>Suma</b>	<b>P</b>	<b>N - P</b>	<b>Categoría</b>	
1	0.4	1	1.4	0.7	0.01	Muy adecuado	
2	0.5	1	1.5	0.75	0.05	Muy adecuado	
3	0.4	1	1.4	0.7	0.01	Muy adecuado	
4	0.4	1	1.4	0.7	0.01	Muy adecuado	
5	0.8	1	1.8	0.9	- 0.19	Muy adecuado	
6	0.4	1	1.4	0.7	0.01	Muy adecuado	
7	0	1	1	0.5	0.21	Muy adecuado	
8	0.4	1	1.4	0.7	0.01	Muy adecuado	
9	0.6	1	1.6	0.8	- 0.09	Muy adecuado	
<b>Suma</b>	3.9	9	12.9				
<b>Puntos de corte</b>	0.43	1					

Luego se le añade:

**Suma:** suma de columnas y suma de filas.

**P:** Promedio.

**N:** División de la suma de las sumas entre el resultado de multiplicar el número de indicadores por el número de preguntas.

**N-P:** Valor promedio que otorgan los expertos para cada indicador propuesto.

Rangos obtenidos a partir de los puntos de corte:

Muy adecuado	Adecuado	No adecuado
Menor que 0.43	0.43 ; 1	Mayor que 1

Todos los resultados son de Muy adecuado por tanto se da por concluida la validación satisfactoriamente. En caso de obtener criterios con valores de Adecuado o No adecuado se deben reelaborar por resultar poco adecuado según las valoraciones de los expertos.

### **Conclusiones**

Se concluyó con la creación del procedimiento para efectuar las pruebas de rendimiento al módulo Tablas de Control. Se escogieron los roles, los artefactos y algunas tareas según lo planteado por RUP y las necesidades del módulo, se organizaron las actividades por fases. Las tareas ayudarán a cumplir los objetivos de cada rol en el proceso, los artefactos serán el resultado tangible de las actividades y quedarán documentados para luego utilizarse como entrada y salida en las fases. Estas últimas agruparán y organizarán las actividades que cumplirán los objetivos trazados para lograr el éxito en la aplicación de las pruebas, cuyos resultados se mostrarán en el próximo capítulo.

# Capítulo 3

## 3. Pruebas de rendimiento al módulo Tablas de Control

### Introducción

En este capítulo se realizarán las pruebas de rendimiento al módulo Tablas de Control utilizando el procedimiento descrito anteriormente y se documentarán los resultados. Se hará una comparación con los requisitos planteados en el Documento de Especificación para darle una evaluación justa a la aplicación y contribuir al mejoramiento de ésta. Se aplicará el proceso usando la herramienta de automatización JMeter para las pruebas de Carga y Estrés, se usará la técnica de Caja Negra trabajando con la entrada de datos y la validación de éstos mediante la comparación con los resultados esperados.

### 3.1 Características fundamentales del módulo

El módulo Tablas de Control (TC) es considerado la columna vertebral de todos los subsistemas que posee la Gestión Integral de las Aduanas (GINA). Las actividades fundamentales que se llevan a cabo en el proyecto se basan en la entrada, almacenamiento, procesamiento y distribución de los datos. TC permite hacer todo lo expresado anteriormente de forma organizada y flexible.

La aplicación cuenta con una pantalla principal que permite la navegación por todo el subsistema y la interacción con las funcionalidades: Dar Alta, Dar Baja, Modificar, Buscar Registro y Procesar Fichero. La interfaz Dar Alta tiene como objetivo registrar un artículo nuevo o ponerlo en vigencia o futuro en caso de que estuviera cerrado, así como sus datos en tablas hijas. La Pantalla Dar Baja permite pasar a estado cerrado un registro en una tabla señalada. La interfaz Modificar nos brinda la posibilidad de modificar un registro. Cuando señalamos la opción Procesar Fichero aparece una pantalla donde se adicionará un fichero, siempre que este tenga la extensión XML. La búsqueda de un fichero se puede realizar por el buscador de la página inicial o mediante la interfaz Búsqueda Avanzada, ambos caminos permiten hacer búsquedas dentro de las tablas especificando parámetros.

Las Tablas de Control facilitan la flexibilidad del sistema, dejan organizada la información y aseguran la consistencia de los datos. La estructura y contenido de estos últimos estará en

correspondencia con las necesidades de análisis, agrupamiento y reagrupamiento de la información, considerándose una aplicación compleja.

La aplicación está implementada sobre la arquitectura cliente-servidor, donde un usuario o cliente se conecta a la aplicación mediante un navegador web haciendo uso de las interfaces JavaScript, lenguaje dinámico y eficiente basado en objetos y guiado por eventos, diseñado específicamente para la arquitectura señalada, vale confirmar que el lenguaje debe ser soportado por el navegador. Se usan las librerías de Ext JS que proporcionan pestañas, menús, barras de herramientas, de diálogos y composiciones automáticas de páginas. El servidor de aplicaciones utiliza Symfony: framework bastante completo que separa la capa de presentación de la de aplicación de la lógica del negocio y la lógica del servidor, facilitando herramientas que reducen el tiempo de desarrollo. Para la gestión de las bases de datos en entornos cliente-servidor es utilizado Oracle, considerado el mayor y más usado Sistema Manejador de Base de Datos Relacional (RDBMS) en el mundo.

### 3.2 Planificación

#### 3.2.1 Estudio de la documentación

Se estudiaron los documentos reunidos: Manual de Usuario, Plan de Aseguramiento de la Calidad y el Documento de Especificación de los Requisitos de Software, logrando un dominio completo de la aplicación y de todas sus funcionalidades. Además se realizó el estudio de tutoriales para el trabajo con la herramienta. Al analizar el documento de Especificación de los Requisitos surgió la necesidad de aclarar algunos de ellos, se planificó entonces una reunión con los especialistas de la AGR y los jefes de cada subsistema quedando como resultado la siguiente propuesta de requisitos no funcionales de rendimiento:

*1- Para el trabajo directo a la aplicación TC*

- Cantidad de usuarios concurrentes que necesita soportar la Aplicación: 50.
- Tiempo de respuesta en situaciones de carga: 6 s.
- Tiempo de recuperación del sistema: 2 s.

*2- Para el trabajo de los módulos con TC*

#### **LCF (Lucha Contra el Fraude)**

- Cantidad de usuarios concurrentes que necesita soportar la Aplicación internamente 20 usuarios, 5 transacciones cada 2 segundos.
- Tiempo de respuesta en situaciones de carga: 200 milisegundos.
- Tiempo de recuperación del sistema: 2 minutos.

### **MTI (Medio de Transporte Internacional)**

- Cantidad de usuarios concurrentes que necesita soportar la Aplicación internamente: 25.
- Tiempo de respuesta en situaciones de carga: 5 a 6 segundos.
- Tiempo de recuperación del sistema: 2 segundos.

### **Administración**

- Cantidad de usuarios concurrentes que necesita soportar la Aplicación internamente: 60.
- Tiempo de respuesta en situaciones de carga: 5 a 6 segundos.
- Tiempo de recuperación del sistema: hasta 10 segundos.

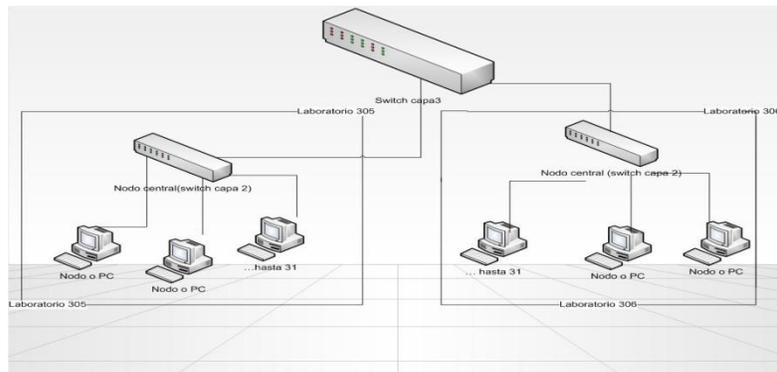
### **Registro Central**

- Cantidad de usuarios concurrentes que necesita soportar la Aplicación internamente: 3, una transacción cada 2 milisegundos.
- Tiempo de respuesta en situaciones de carga: 4 segundos.
- Tiempo de recuperación del sistema: 1 día.

### **3.2.2 Análisis de los recursos materiales**

La simulación de las pruebas se hará en los laboratorios pertenecientes al proyecto AGR, son dos laboratorios que cuentan con 31 PC cada uno para un total de 62 PC.

La red está organizada con una topología en forma de árbol que consiste en un conjunto de subredes conectadas a un enlace troncal. El nodo central o enlace está ocupado por un switch capa 2 conectando todas las máquinas en un mismo canal de comunicación y controlando el tráfico de la información. Las computadoras están conectadas por un cableado UTP a los puntos de red que las unen al switch ubicado en cada laboratorio, este a su vez se conecta a otro switch capa 3. Esta topología tiene como ventaja amplificar la potencia en la transmisión de señales viajando éstas a mayores distancias.



**Figura 10:** Distribución de la red

Características del switch capa 2:

- Aísla el tráfico entre las subredes.
- Segmenta la red en dominios de colisión y disminuye el tamaño de éstos.

Características del switch capa 3:

- Maneja eficientemente el tráfico.
- Maneja sus propios recursos sin consumir ancho de banda, ni generar tráfico innecesario.
- Tiene todos los niveles de seguridad con los que debe contar un enrutador.

### 3.3 Diseño de las pruebas

#### 3.3.1 Modelo de análisis de carga

**Estudio inicial:** Se analizaron los resultados obtenidos en las pruebas realizadas a otros módulos del proyecto para ganar en experiencia y valorar el comportamiento de algunos parámetros en condiciones similares. Concluyendo que:

- Las páginas implementadas en Java Script consumen un tiempo de respuesta demasiado grande retardando la repuesta de la aplicación.
- Al utilizar la herramienta JMeter se crean niveles de estrés muy altos y consume demasiada memoria RAM.
- Al realizar pruebas con un número de concurrencia alto la base de datos deja de responder provocando la caída de la aplicación.

- Se encuentra un porcentaje de errores mayor al utilizar el navegador Firefox que al usar el Internet Explorer.
- Al simular una carga de 100 usuarios conectados concurrentemente el servidor no soportó y no fue satisfactoria la prueba.

### **Selección de las variables del sistema:**

*Sistema Operativo:* Windows 7 Profesional en las máquinas clientes y Ubuntu 11.4 en el servidor.

- Máquina Virtual: Java Virtual Machine 1.3 o superior.
- Herramienta de prueba: JMeter Versión 2.4.
- Antivirus: Kaspersky Workstation 6.0.
- Gestor de Base de Datos: Oracle Standard Edition ONE Versión 11.
- Ancho de banda: 10/100 Mb/s.

### **Selección de los atributos del actor:**

- Tiempo medio entre conexiones.
- Cantidad de usuarios que pueden acceder a la aplicación en horario normal de forma concurrente.
- Cantidad de usuarios que pueden acceder a la aplicación en horario pico de forma concurrente.
- Muestras: Cantidad de páginas (Hilos) que simulan la cantidad de usuarios que están interactuando con el sistema desde la misma URL.
- Media: Media de páginas que se cargaron de manera satisfactoria.
- Mediana: Tiempo promedio que han tardado en cargarse las páginas.
- 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 de las páginas que se cargaron de manera satisfactoria.
- % Error: Por ciento de error de las páginas que no se llegaron a cargar de manera satisfactoria, debe estar por debajo del 1%.
- Kb/s: Velocidad de carga de las páginas.

**Simulación de la carga de trabajo:** Se simulará una carga de trabajo que variará entre 50 y 150 usuarios conectados concurrentemente a cada una de las interfaces en las pruebas de carga,

valorando una transacción cada 2 milisegundos. En las pruebas de Estrés serán de 60 a 120 usuarios en iteraciones de 10 a 5 para un total de 600 peticiones.

**Valorar límites de rendimiento:** Anteriormente se realizaron pruebas de rendimiento al módulo Control de Persona, se tuvo en cuenta la conexión de 50 usuarios para ver el mínimo de rendimiento y 100 usuarios para valorar el máximo en una misma interfaz, se obtuvieron los siguientes resultados:

Para 50 usuarios conectados simultáneamente el rendimiento varió entre 2.5 /s y 23.8 /s. El sistema no pudo responder para 100 usuarios, el servidor no fue capaz de atender la sobrecarga de peticiones. El rendimiento estuvo por debajo de lo esperado.

Probando otra interfaz, al simular 50 usuarios y generar 2800 muestras el rendimiento alcanzó valores de 1042.4 /s y de 16253.6 /s. Para 100 usuarios conectados se generaron 5600 muestras obteniéndose un rendimiento de 1000.2 /s y de 15594.6 /s. Se consideró un rendimiento satisfactorio.

### 3.3.2 Configuración del entorno de prueba

Teniendo en cuenta la cantidad de usuarios a simular se trabajará en las pruebas con máquinas clientes de 1GB de memoria RAM y un disco duro de 160 GB, con un microprocesador Intel(R) Core 2 Duo a 2.02 GHz.

Se cuenta con un servidor de Aplicación, el cual a su vez, trabaja con un microprocesador Intel(R) Core 2 Duo a 2.02 GHz, 2 GB de memoria RAM y 160 GB de capacidad en el disco duro, que permitirá la transferencia rápida de información. También se usará un servidor de Base de Datos con un sistema gestor de datos cliente-servidor que responderá a las peticiones de forma confidencial, este se caracteriza por poseer un disco de 80 GB y 2 GB de memoria RAM, es una Intel(R) Pentium(R) D con un CPU de 3.00GHz.

En las regulaciones de software se señala contar con una copia estable de la aplicación para hacer las pruebas, instalar la máquina virtual de Java versión 1.3 o superior en un servidor, poseer el gestor de Base de Datos Oracle versión 11.0, instalar el paquete Office 2010 y otras herramientas básicas.

**3.3.3 Diseño de los casos de prueba**

Para el diseño de los casos de prueba se utilizó la planilla escogida anteriormente. A continuación se presenta la descripción del Caso de Prueba 1 para las pruebas de carga y estrés, el resto de los casos diseñados se encuentran en los Anexos.

**Caso de Prueba 1:**

**Nombre del Proyecto:** Gestión Integral de Aduanas.

**Nombre Módulo:** Tablas de Control.

**Versión:** 1.0

**Tipo de prueba:** Prueba de Carga.

Funcionalidad(es)	Comentario
Procesar Fichero	El especialista de la Aduana selecciona la opción Procesar Fichero dándole la posibilidad de eliminar o resetear ficheros así como añadir o subir un archivo con extensión XML.

*Variables:*

Para la Variable Tiempo entre conexión y conexión se propone valor cero pues simularán usuarios conectados al mismo tiempo y el número de iteraciones tendrá valor 1 porque solo se conectarán una vez.

Para 50 usuarios:

Variables	Valores
<b>Número de usuarios concurrentes</b>	50
<b>Tiempo entre conexión y conexión</b>	0
<b>Número de iteraciones</b>	1

Para 100 usuarios:

Variables	Valores
Número de usuarios concurrentes	100
Tiempo entre conexión y conexión	0
Número de iteraciones	1

Para 150 usuarios:

Variables	Valores
Número de usuarios concurrentes	150
Tiempo entre conexión y conexión	0
Número de iteraciones	1

**Caso de Prueba 1:**

**Nombre del Proyecto:** Gestión Integral de Aduanas.

**Nombre Módulo:** Tablas de Control.

**Versión:** 1.0

**Tipo de prueba:** Prueba de Estrés.

Funcionalidad(es)	Comentario
Procesar Fichero	El especialista de la Aduana selecciona la opción Procesar Fichero dándole la posibilidad de eliminar o resetear ficheros señalados, así como añadir o subir un archivo con extensión XML.

*Variables:*

En las pruebas de estrés el Tiempo de conexión definirá la distancia entre una conexión y otra de usuarios conectándose escalonadamente y el número de iteraciones será la cantidad de veces que se conectará ese hilo de usuario.

Para 600 usuarios

Variables	Valores
<b>Número de usuarios</b>	60
<b>Tiempo entre conexión y conexión</b>	4
<b>Número de iteraciones</b>	10

Para 630 usuarios:

Variables	Valores
<b>Número de usuarios</b>	90
<b>Tiempo entre conexión y conexión</b>	3
<b>Número de iteraciones</b>	7

Para 600 usuarios:

Variables	Valores
<b>Número de usuarios</b>	120
<b>Tiempo entre conexión y conexión</b>	2
<b>Número de iteraciones</b>	5

### 3.4 Ejecución de las pruebas

#### 3.4.1 Funcionalidad Procesar Fichero

*Resultados de las pruebas de Carga:*

Al efectuar el caso de prueba # 1 para la funcionalidad Procesar Fichero se obtuvieron los siguientes resultados:

Para 50 usuarios conectados concurrentemente el CPU estuvo por debajo del 71 % y la memoria del 83 %. El 33.3 % de error está dado por el vínculo:

<http://10.52.17.4/aduana/tc.php/upload/ValidarXml>

## PRUEBAS DE RENDIMIENTO AL MÓDULO TABLAS DE CONTROL

Petición	Muestras	Media	Mediana	Min	Max	Línea 90 %	%Error	Rendimiento	Kb/s
Procesar Fichero	150	1923	900	0	4907	2069	33.33	20.3 /s	95.7

Para 100 usuarios el CPU mantuvo valores por debajo del 85% y la memoria de un 81 a 82 %.

Petición	Muestras	Media	Mediana	Min	Max	Línea 90 %	%Error	Rendimiento	Kb/s
Procesar Fichero	300	2293	1069	0	10048	8072	33.33	27.1 /s	56.7

Para 150 usuarios conectados concurrentemente la aplicación no soportó la carga, quedando varias peticiones sin responder debido al error encontrado anteriormente.

Los valores de rendimiento no cumplieron las 1000 solicitudes/s, el porcentaje de error es alto pero el tiempo promedio de respuesta es bastante aceptable puesto que en ambas pruebas está por debajo de 1.09.

### *Resultados de las pruebas de Estrés:*

En las pruebas de estrés a la funcionalidad Procesar Fichero se obtuvieron los resultados siguientes:

Para 60 usuarios el CPU estuvo por debajo del 56 % y la memoria entre 79 y 81 % simulándose todas las peticiones, se encontraron errores en:

<http://10.52.17.4/aduana/tc.php/upload/ValidaXml>

Petición	Muestras	Media	Mediana	Min	Max	Línea 90 %	%Error	Rendimiento	Kb/s
Procesar Fichero	1800	1789	2313	0	6450	3302	33.33	30.7 /s	64.2

Para 90 usuarios el CPU estuvo por debajo del 77 % y la memoria entre un 82 y un 85 %, se respondieron a todas las peticiones y se encontraron los mismos errores.

Petición	Muestras	Media	Mediana	Min	Max	Línea 90 %	%Error	Rendimiento	Kb/s
Procesar Fichero	1890	1455	1770	0	6501	2752	33.33	54.3 /s	113.8

Para 120 usuarios la aplicación no soportó.

### 3.4.2. Funcionalidad Dar Alta

*Resultados de las pruebas de Carga:*

En el Caso de Prueba # 2 para la funcionalidad Dar Alta se obtuvieron los siguientes resultados:

Para 50 usuarios conectados concurrentemente se obtuvieron 500 muestras sin errores para un rendimiento de 16 /s y una velocidad de 44.4 kb/s. En esta prueba el CPU se mantuvo en el 95% y la memoria física de 90 a 95 % estando en ambos casos por encima de los valores establecidos.

Petición	Muestras	Media	Mediana	Min	Max	Línea 90 %	%Error	Rendimiento	Kb/s
Dar Alta	500	2956	2610	260	12040	5269	0.00	16.5 /s	44.4

Con 100 usuarios se lograron 1000 muestras con un rendimiento de 25.4/s y una tasa de error de 10.10 % que representa que 100 peticiones no pudieron ser atendidas. El CPU se mantuvo por debajo del 85 % y la Memoria física de un 80% a un 86%. Los errores encontrados están en el vínculo:

[http://10.52.17.4/aduana/tc.php/tc/datosTc?\\_dc=1337633758114](http://10.52.17.4/aduana/tc.php/tc/datosTc?_dc=1337633758114)

Petición	Muestras	Media	Mediana	Min	Max	Línea 90 %	%Error	Rendimiento	Kb/s
Dar Alta	1000	3284	1041	59	21332	7507	10.10	25.4/s	68.7

Al conectar 150 usuarios el sistema no pudo responder llegando solo a simular 1975 muestras con un 12.66 % de error. El error se encontró en:

[http://safebrowsing.clients.google.com/safebrowsing/downloads?client=navclient-auto-ffox&appver=12.0&pver=2.2&wrkey=AKEgNis71fU6PvUdTXNWiojZ7l\\_jl4CoZgTNN3S0EXcz4LBeV1hUMzPvJb96ANil-RjYwpLb4hzYvrOr89hOAdGlcPnzckIGLg](http://safebrowsing.clients.google.com/safebrowsing/downloads?client=navclient-auto-ffox&appver=12.0&pver=2.2&wrkey=AKEgNis71fU6PvUdTXNWiojZ7l_jl4CoZgTNN3S0EXcz4LBeV1hUMzPvJb96ANil-RjYwpLb4hzYvrOr89hOAdGlcPnzckIGLg)

Podemos concluir que para esta funcionalidad el sistema responde insatisfactoriamente pues los valores de rendimiento son bastante inestables para 50 y 100 usuarios y están por debajo de las 1000 transacciones por segundo.

*Resultados de las pruebas de Estrés:*

Al efectuarse las pruebas de Estrés para el Caso de Prueba # 2 se obtuvieron los resultados que a continuación se presentan:

Para 60 usuarios conectados en 10 iteraciones el CPU se mantuvo por debajo del 71 % y la memoria entre un 79 y un 81 % simulándose todas las peticiones sin encontrar errores.

Petición	Muestras	Media	Mediana	Min	Max	Línea 90 %	%Error	Rendimiento	Kb/s
Dar Alta	5400	1527	1480	110	6556	1886	0.00	37.6 /s	107.0

Para 90 usuarios conectados en 7 iteraciones el CPU alcanzó hasta el 81 % pasando los límites establecidos y la memoria se mantuvo entre un 85 y un 89 % debido a que se simularon más usuarios en menos tiempo.

Petición	Muestras	Media	Mediana	Min	Max	Línea 90 %	%Error	Rendimiento	Kb/s
Dar Alta	5670	2193	2192	103	8519	2597	0.00	39.0 /s	110.9

Para 120 usuarios se generó una carga que la aplicación no soportó y dejaron de responderse 8 peticiones por el error:

[http://safebrowsing.clients.google.com/safebrowsing/downloads?client=navclient-auto-ffox&appver=12.0&pver=2.2&wrkey=AKEgNis71fU6PvUdTXNWiojZ7l\\_jl4CoZgTNN3S0EXcz4LBeV1hUMzPvJb96ANil-RjYwpLb4hzYvrOr89hOAdGlcPnzckIGLg](http://safebrowsing.clients.google.com/safebrowsing/downloads?client=navclient-auto-ffox&appver=12.0&pver=2.2&wrkey=AKEgNis71fU6PvUdTXNWiojZ7l_jl4CoZgTNN3S0EXcz4LBeV1hUMzPvJb96ANil-RjYwpLb4hzYvrOr89hOAdGlcPnzckIGLg)

El rendimiento en esta funcionalidad no llegó a cumplir los límites establecidos, los resultados de los demás parámetros fueron satisfactorios pues los valores obtenidos en las pruebas para 60 y 90 usuarios se comportaron muy similares, la velocidad de carga de las páginas en ambos casos solo se diferenció en 3.9 Kb/s y no se encontraron errores.

### 3.4.3. Funcionalidad Dar Baja

*Resultados de las pruebas de Carga:*

Al desarrollar el caso de prueba # 3 simulando 50 usuarios, el CPU estuvo por debajo del 63 % y la memoria de un 82 %, no se encontraron errores y los resultados se exponen a continuación:

Petición	Muestras	Media	Mediana	Min	Max	Línea 90 %	%Error	Rendimiento	Kb/s
Dar Baja	500	1705	1671	10	8678	2688	0.00	23.9 /s	60.1

Para 100 usuarios el CPU estuvo por debajo del 42 % y la memoria de un 89 % sin encontrarse errores.

Petición	Muestras	Media	Mediana	Min	Max	Línea 90 %	%Error	Rendimiento	Kb/s
Dar Baja	1600	3422	2893	150	13393	5639	0.00	18.0 /s	51.3

Para 150 usuarios, como en la funcionalidad Dar Alta, la aplicación no soportó la carga.

La funcionalidad Dar Baja no cumplió el rendimiento señalado pero mantuvo valores estables al probar con 50 y 100 usuarios.

*Resultados de las pruebas de Estrés:*

Al efectuar el Caso de Prueba # 3 se obtuvieron los resultados siguientes:

Para 60 usuarios conectados en 10 iteraciones el CPU se mantuvo por debajo del 55 % y la memoria entre 77 y 81 %, simulándose todas las peticiones sin encontrar errores.

Petición	Muestras	Media	Mediana	Min	Max	Línea 90 %	%Error	Rendimiento	Kb/s

## PRUEBAS DE RENDIMIENTO AL MÓDULO TABLAS DE CONTROL

Dar Baja	4800	2898	2911	129	7913	3533	0.00	20.0 /s	57.0
----------	------	------	------	-----	------	------	------	---------	------

Para 90 usuarios el CPU conectados en 7 iteraciones estuvo por debajo del 54 % y la memoria entre un 72 y un 78 %, se respondieron a todas las peticiones sin encontrar errores.

Petición	Muestras	Media	Mediana	Min	Max	Línea 90 %	%Error	Rendimiento	Kb/s
Dar Baja	5040	2376	2419	134	6617	2787	0.00	36.7 /s	104.4

Para 120 usuarios se generó una carga que llevó al CPU y a la memoria a sus máximos valores y la aplicación no soportó, dejaron de responder 9 peticiones.

La funcionalidad Dar Baja en la prueba de estrés no cumplió con los valores de rendimiento, no obstante los resultados en las iteraciones 1 y 2 fueron muy estables en cuanto a la velocidad de carga de las páginas.

### 3.4.4. Funcionalidad Modificar

*Resultados de las pruebas de Carga:*

Al efectuar el caso de prueba # 4 para la funcionalidad modificar registro se obtuvieron los resultados que a continuación se exponen:

Para 50 usuarios conectados concurrentemente el CPU estuvo por debajo del 51 % y la memoria de un 81 a un 82 %.

Petición	Muestras	Media	Mediana	Min	Max	Línea 90 %	%Error	Rendimiento	Kb/s
Modificar Registro	1400	1100	890	111	9926	1556	0.00	10.5 /s	29.8

Para 100 usuarios el CPU mantuvo valores por debajo del 69 % y la memoria de 87 %.

Petición	Muestras	Media	Mediana	Min	Max	Línea 90 %	%Error	Rendimiento	Kb/s
Modificar Registro	1400	2414	2237	161	10383	2991			

## PRUEBAS DE RENDIMIENTO AL MÓDULO TABLAS DE CONTROL

							0.00	28.0 /s	79.5
--	--	--	--	--	--	--	------	---------	------

Para 150 usuarios conectados concurrentemente la aplicación no soportó la carga, quedando varias peticiones sin responder debido al siguiente error:

[http://safebrowsing.clients.google.com/safebrowsing/downloads?client=navclient-auto-ffox&appver=12.0&pver=2.2&wrkey=AKEgNis71fU6PvUdTXNWiojZ7I\\_jl4CoZgTNN3S0EXcz4LBeV1hUMzPvJb96ANil-RjYwpLb4hzYvrOr89hOAdGlcPnzckKIGLg](http://safebrowsing.clients.google.com/safebrowsing/downloads?client=navclient-auto-ffox&appver=12.0&pver=2.2&wrkey=AKEgNis71fU6PvUdTXNWiojZ7I_jl4CoZgTNN3S0EXcz4LBeV1hUMzPvJb96ANil-RjYwpLb4hzYvrOr89hOAdGlcPnzckKIGLg)

Se concluyó que el número de muestras en ambos casos fue la misma a pesar de que los hilos simulados tenían el doble de diferencia, los valores de tiempo promedio de carga de las páginas y la velocidad fueron muy inestables, no se cumplieron los valores de rendimiento.

### *Resultados de las pruebas de Estrés:*

En las pruebas de estrés hechas a la funcionalidad Modificar Registro se obtuvieron los resultados siguientes:

Para 60 usuarios el CPU estuvo por debajo del 75 % llegando a los límites y la memoria entre 80 y 88 % simulándose todas las peticiones sin encontrar errores.

Petición	Muestras	Media	Mediana	Min	Max	Línea 90 %	%Error	Rendimiento	Kb/s
Modificar Registro	4200	2964	3007	198	7659	3583	0.00	19.6 /s	55.7

Para 90 usuarios el CPU estuvo por debajo del 54 % y la memoria entre un 72 y un 78 %, se respondieron a todas las peticiones sin errores.

Petición	Muestras	Media	Mediana	Min	Max	Línea 90 %	%Error	Rendimiento	Kb/s
Modificar Registro	4410	2466	2347	153	16926	2830	0.00	34.6 /s	98.4

Para 120 usuarios se generó una carga que llevó al CPU y a la memoria a sus máximos valores y la aplicación no soportó, dejaron de responder 13 peticiones, encontrando el error en el vínculo:

[http://10.52.17.4/aduana/tc.php/tc/datosTc?\\_dc=1337714619384](http://10.52.17.4/aduana/tc.php/tc/datosTc?_dc=1337714619384)

Las pruebas de estrés mostraron un rendimiento y un tiempo promedio de carga de las páginas inaceptable.

### **3.4.5. Funcionalidad Buscar Registro**

La funcionalidad buscar registro del caso de prueba # 5 cuenta con dos secciones; en la primera el camino comienza seleccionando la tabla y luego introduciendo el identificador en el buscador, se conoce como Buscar Registro; la segunda es por la opción Búsqueda Avanzada.

*Resultados de las pruebas de Carga:*

#### **Buscar Registro**

Para 50 usuarios conectados concurrentemente el CPU se observó por debajo del 50 % y la memoria entre el 81 y 82 %. Obteniéndose:

Petición	Muestras	Media	Mediana	Min	Max	Línea 90 %	%Error	Rendimiento	Kb/s
Buscar Registro	300	2234	1711	324	9160	5894	0.00	17.2 /s	48.8

Para 100 usuarios el CPU alcanzó hasta el 74 % y la memoria hasta el 83 %.

Petición	Muestras	Media	Mediana	Min	Max	Línea 90 %	%Error	Rendimiento	Kb/s
Buscar Registro	700	2314	1728	191	10908	4732	0.00	32.6 /s	92.8

Para 150 la aplicación no soportó la carga de usuarios, el CPU llegó al 78 % y la memoria al 88%, quedaron 21 peticiones sin responder.

#### **Búsqueda Avanzada**

## PRUEBAS DE RENDIMIENTO AL MÓDULO TABLAS DE CONTROL

Para 50 usuarios el CPU llegó al 54 % y a un 81 % la memoria. Los valores obtenidos fueron:

Petición	Muestras	Media	Mediana	Min	Max	Línea 90 %	%Error	Rendimiento	Kb/s
Búsqueda Avanzada	300	2128	1647	271	7376	6194	0.00	18.7/s	53.3

Para 100 usuarios el CPU alcanzó hasta el 65 % y la memoria se mantuvo entre 83 y 85 %.

Petición	Muestras	Media	Mediana	Min	Max	Línea 90 %	%Error	Rendimiento	Kb/s
Búsqueda Avanzada	600	2308	1759	131	10587	6557	0.00	32.3 /s	91.8

Para 150 la aplicación no soportó la carga de usuarios, quedaron 21 peticiones sin responder.

En la prueba de carga para la funcionalidad Buscar Registro los valores de rendimiento no llegaron a los establecidos. El tiempo promedio de carga se observó estable en todas las iteraciones al igual que la velocidad de carga, no se encontraron errores, el porcentaje de páginas cargadas fue satisfactorio.

*Resultados de las pruebas de Estrés:*

### Buscar Registro

Para 60 usuarios el CPU se observó por debajo del 56 % y la memoria de 83 a 85 %. Los resultados de esta prueba se muestran a continuación:

Petición	Muestras	Media	Mediana	Min	Max	Línea 90 %	%Error	Rendimiento	Kb/s
Buscar Registro	3 600	2727	2786	118	7837	3192	0.00	21.1 /s	60.1

Para 90 usuarios concurrentemente conectados el CPU alcanzó hasta el 52 % y la memoria se mantuvo entre el 87 y el 93 %.

## PRUEBAS DE RENDIMIENTO AL MÓDULO TABLAS DE CONTROL

Petición	Muestras	Media	Mediana	Min	Max	Línea 90 %	%Error	Rendimiento	Kb/s
Buscar Registro	4410	4361	4388	159	24878	7537	0.00	19.9 /s	56.5

Para 120 la aplicación no soportó la carga de usuarios, el CPU llegó al 95 % y la memoria al 90%, quedaron 12 peticiones sin responder y se encontraron varios errores en:

[http://10.52.17.4/aduana/tc.php/tc/datosTc?\\_dc=1337715343974](http://10.52.17.4/aduana/tc.php/tc/datosTc?_dc=1337715343974)

### Búsqueda Avanzada

Para 60 usuarios el CPU alcanzó el 58 % y un 90 % la memoria sin observarse errores. Los valores obtenidos en las pruebas fueron:

Petición	Muestras	Media	Mediana	Min	Max	Línea 90 %	%Error	Rendimiento	Kb/s
Búsqueda Avanzada	3600	2778	2726	142	17520	6194	0.00	19.9/s	56.5

Para 90 usuarios el CPU alcanzó hasta el 56 % y la memoria se mantuvo entre 89 y 93 %

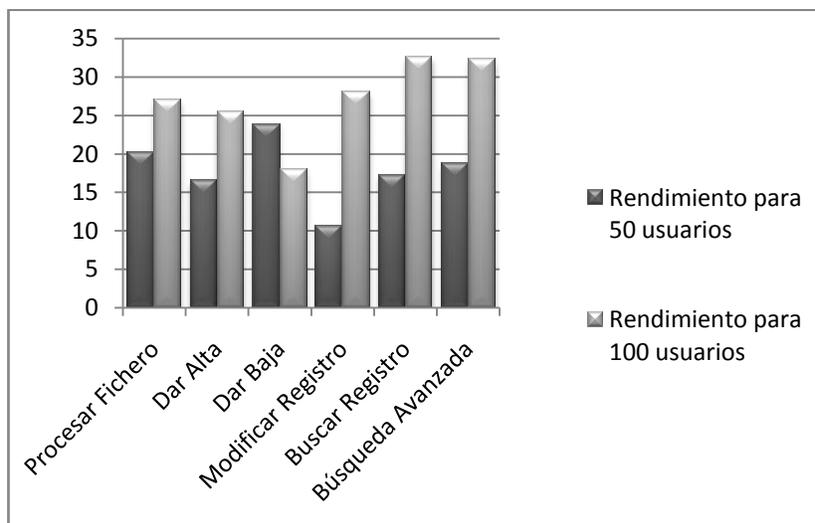
Petición	Muestras	Media	Mediana	Min	Max	Línea 90 %	%Error	Rendimiento	Kb/s
Búsqueda Avanzada	3780	4187	4223	171	12371	4946	0.00	20.6 /s	58.7

Para 120 la aplicación no soportó la carga de usuarios, quedaron 11 peticiones sin responder, el CPU llegó al 97 % y la memoria al 89 %.

### 3.5 Evaluación de las pruebas

A continuación se observan los gráficos obtenidos para los valores de Rendimiento de las Pruebas de Carga y Estrés.

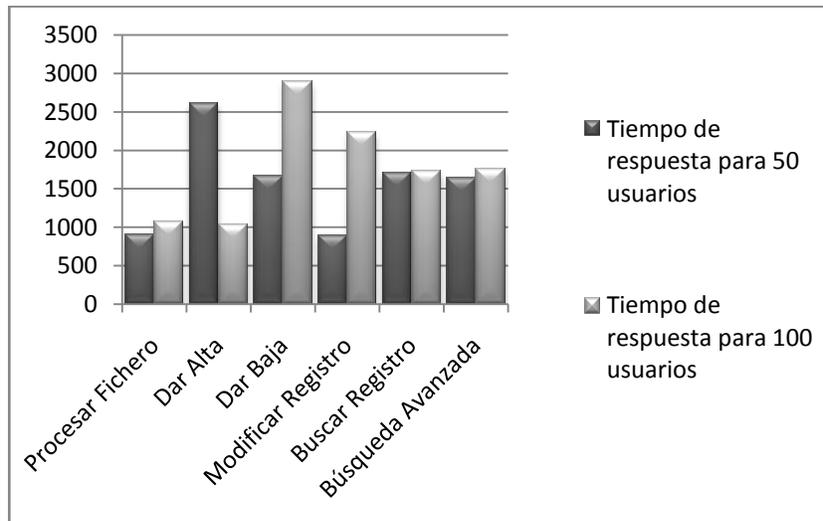
Para 50 y 100 usuarios conectados concurrentemente en las pruebas de Carga, teniendo en cuenta que para 150 peticiones la aplicación no soportó, el rendimiento fue insatisfactorio en las 6 funcionalidades quedando por debajo de las 1000 solicitudes/s, valor establecido en el documento de Especificación de los Requisitos de Software.



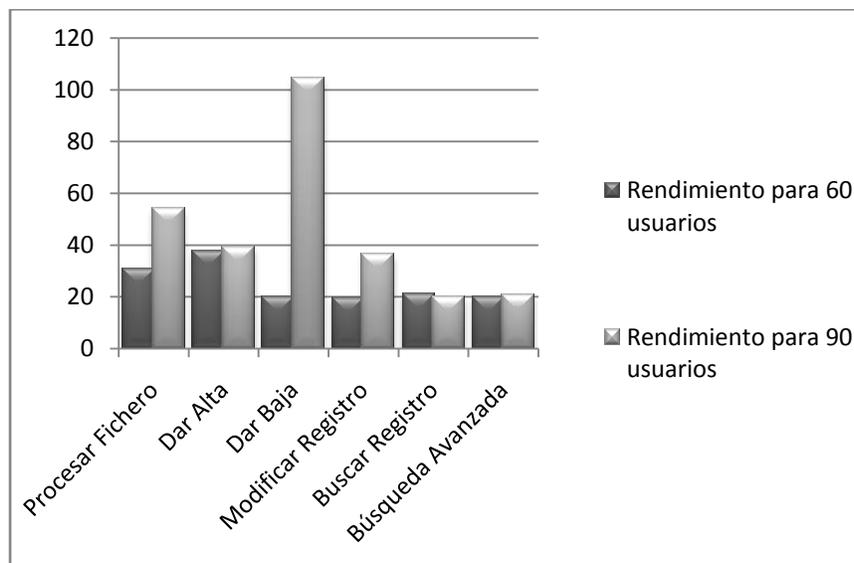
**Figura 11:** Rendimiento para las Pruebas de Carga

Los tiempos promedio de respuesta fueron aceptables observándose todos por debajo de los 3 s, las seis funcionalidades mostraron tiempos inferiores a los 1.7 s cuando se conectaron 50 usuarios, al conectar 100 subieron los valores, siendo la funcionalidad Buscar Registro la de mayor tiempo. Estos resultados compensan el mal rendimiento de la aplicación, puesto que el tiempo de respuesta satisface las expectativas del cliente obteniendo una buena opinión de la aplicación al no tener que esperar demasiado por las respuestas del sistema. Ver Figura 10.

En las pruebas de Estrés al conectar 60 y 90 usuarios para 10 y 7 iteraciones respectivamente con un tiempo entre conexión y conexión menor a 4 s para un total de 600 y 630 usuarios, los valores de rendimiento no cumplen las 1000 solicitudes/s aunque se comportaron mejor que en las pruebas de Carga. Ver Figura 11.

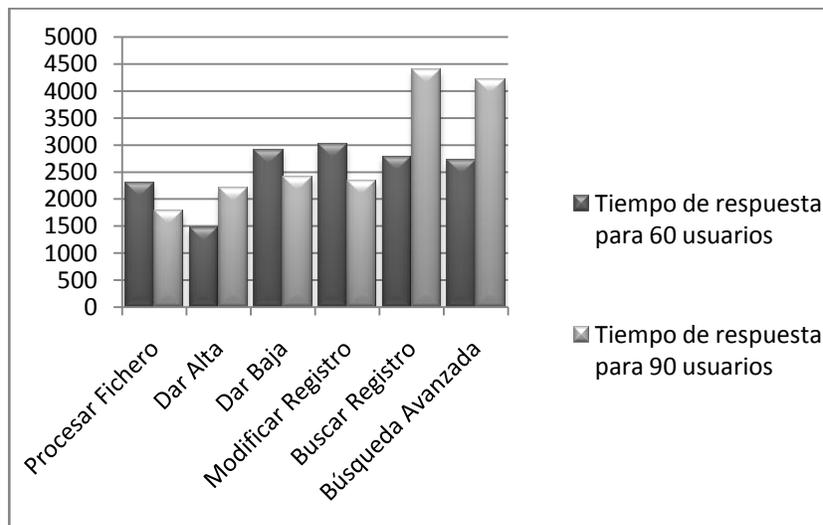


**Figura 12:** Tiempo de Respuesta para las Pruebas de Carga



**Figura 13:** Rendimiento para las Pruebas de Estrés

El tiempo de respuesta en las pruebas de estrés también fue aceptable manteniéndose por debajo de los 4.5 s, todas las funcionalidades al conectar 60 usuarios mostraron tiempos inferiores a 3 s. Al conectar 90 usuarios, en cuatro funcionalidades se observaron valores inferiores a 2.5 s mientras las funcionalidades Buscar Registro y Búsqueda Avanzada se mostraron por encima de los 4 s.



**Figura 14:** Tiempo Promedio de Respuesta para las Pruebas de Estrés

Se observó que la utilización del CPU en las pruebas de Carga siempre fue menor que en las pruebas de Estrés pero en ambos casos solo pasó del 70 % la funcionalidad Dar Alta, mostrando valores por encima del 75 % que no son aceptados. Se pudo ver como el uso del CPU en las pruebas donde la aplicación no soportó la carga llegó hasta el 100 % de utilización y la memoria al 93%. El uso de la memoria física en todas las pruebas pasó del 18 %, incluso por encima del 70%.

Se encontraron varios errores en el transcurso de las pruebas, estos fueron en los vínculos:

<http://10.52.17.4/aduana/tc.php/upload/ValidarXml>

[http://10.52.17.4/aduana/tc.php/tc/datosTc?\\_dc=1337633758114](http://10.52.17.4/aduana/tc.php/tc/datosTc?_dc=1337633758114)

[http://10.52.17.4/aduana/tc.php/tc/datosTc?\\_dc=1337714619384](http://10.52.17.4/aduana/tc.php/tc/datosTc?_dc=1337714619384)

[http://10.52.17.4/aduana/tc.php/tc/datosTc?\\_dc=1337715343974](http://10.52.17.4/aduana/tc.php/tc/datosTc?_dc=1337715343974)

[http://safebrowsing.clients.google.com/safebrowsing/downloads?client=navclient-auto-ffox&appver=12.0&pver=2.2&wrkey=AKEgNis71fU6PvUdTXNWiojZ7I\\_jl4CoZgTNN3S0EXcz4LBeV1hUMzPvJb96ANil-RjYwpLb4hzYvrOr89hOAdGlcPnzckIGLg](http://safebrowsing.clients.google.com/safebrowsing/downloads?client=navclient-auto-ffox&appver=12.0&pver=2.2&wrkey=AKEgNis71fU6PvUdTXNWiojZ7I_jl4CoZgTNN3S0EXcz4LBeV1hUMzPvJb96ANil-RjYwpLb4hzYvrOr89hOAdGlcPnzckIGLg)

En la entrevista con los especialistas de la aduana se acordó que en los entornos reales de explotación existirían de 6 a 10 usuarios conectados concurrentemente a la aplicación realizando diferentes tareas. Valorar los tiempos para esta carga se hizo una prioridad y se escogió para la

prueba la funcionalidad Modificar Registro ya que fue la que menos rendimiento obtuvo en las pruebas de carga. Se simularon 15 usuarios dejando un margen para 5 conexiones más, los valores arrojados están en la tabla del Anexo # 9.

Para 765 muestras el tiempo promedio de respuesta fue de 54 milisegundos, estando muy por debajo de lo planteado en los requisitos y cumpliendo las expectativas del cliente. El rendimiento no alcanzó las 1000 solicitudes/s pero superó en 70 solicitudes/s al mejor resultado obtenido en las pruebas de carga. En estas condiciones no se encontraron errores y la velocidad de carga fue de 124.7 superando también a las obtenidas anteriormente. Los valores de consumo del CPU y la memoria fueron de 85 y 89%, superando en ambos casos lo requerido pero se debe aclarar que las características de hardware de las computadoras utilizadas para las pruebas son inferiores a las que se poseerá en el entorno real de explotación.

En general se recomienda mejorar los valores de rendimiento, revisar las funcionalidades ineficientes, recordando que esta aplicación trabaja con un alto volumen de datos y las operaciones que realiza son de gran complejidad.

### **Conclusiones**

En el capítulo que aquí concluye se desarrollaron las pruebas al módulo TC utilizando el procedimiento descrito en el capítulo 2. Primeramente se hizo un estudio de las principales características de la aplicación utilizando los documentos del expediente de proyecto, luego se hizo un análisis de la red y de la posible carga a simular para configurar el entorno de prueba. Se diseñaron los casos de prueba y se produjo una ejecución satisfactoria de estos en la herramienta JMeter. Se analizaron los resultados obtenidos en las pruebas de carga y estrés, llegando a la conclusión de que la aplicación soporta 100 usuarios conectados al mismo tiempo y hasta 630 en 7 iteraciones para un tiempo de conexión de 3 segundos, la funcionalidad con un rendimiento más crítico fue la de Modificar Registro y el tiempo promedio de respuesta más alto fue de 4.5 s en la funcionalidad Buscar Registro.

Se puede afirmar que se cumplieron todos los objetivos propuestos, logrando efectuar pruebas que arribaron a importantes resultados que servirán para mejorar el rendimiento de la aplicación y lograr que trabaje eficientemente en los entornos de explotación. Se encontraron errores que demuestran la profundidad del proceso y sirven de retroalimentación para que las próximas versiones sean más eficientes.

**CONCLUSIONES**

Se puede concluir que el presente trabajo cumplió los objetivos trazados. Los resultados de las pruebas de Carga y Estrés validaron el requisito no funcional de rendimiento y la eficiencia del procedimiento descrito. El procedimiento propuesto garantizó la organización e integridad del proceso con un alto grado de seguridad gracias a la definición de artefactos y actividades, recomendándose su uso en otras aplicaciones dentro del proyecto.

El modelo de análisis de carga permitió una configuración del entorno de pruebas efectivo que logró soportar la simulación de las pruebas y llevar la aplicación a sus límites. El diseño de los casos de prueba usando los requisitos de software ayudó a desarrollar una ejecución clara de las mismas probando todas las funcionalidades que poseen interfaz.

Se comprobó que el trabajo con la herramienta JMeter minimiza la inversión de tiempo y recursos, obteniendo una medición más exacta que al realizar las pruebas de forma manual. Se realizó una comparación de los resultados obtenidos con los valores planteados en el documento Especificación de los Requisitos de Software arribando a importantes conclusiones sobre la calidad de la aplicación.

Los errores encontrados y las funcionalidades deficientes quedaron registrados para próximas valoraciones que ayudarán a mejorar la calidad del producto y servirán de experiencia para el desarrollo de otras soluciones.

### RECOMENDACIONES

- Revisar los errores encontrados en las pruebas realizadas y corregirlos.
- Optimizar el funcionamiento de la aplicación para mejorar los valores de rendimiento.
- Revisar las funcionalidades con un tiempo de respuesta crítico.
- Realizar otras iteraciones de prueba para verificar la rectificación de los errores.
- Profundizar en el estudio de otros tipos de pruebas de rendimiento para evaluaciones futuras de la aplicación.

## REFERENCIA BIBLIOGRÁFICA

1. **Auditoría, Departamento de Control de Calidad y.** *Control de Calidad en los Sistemas.* 2000.
2. **SALANOVA, P.E.** *Modelos de Calidad Web, Universidad Nacional de Educación a Distancia:308.* [En línea] 2006. [Citado el: 10 de 1 de 2011.]
3. **PRESSMAN, R.S.** *Ingeniería del Software.Un enfoque práctico.ed.McGraw Hill.* s.l. : vol.I,500p, 2005.
4. **Borrero, Lázara Y. Perez Piñero y Yurina López.** *Propuesta de Pruebas para el Sistema de Manejo de Ensayos Clínicos.* La Habana : s.n., 2007.
5. **8402, ISO.** *Quality-Vocabulary.* 1994.
6. **Gutierrez, J.J y Escalona, M.J.** *Estudio comparativo de propuestas para la generacion de casos de prueba a partir de requisitos funcionales.* [En línea] 1 de 2005. [Citado el: 6 de 1 de 2010.] <http://www.lsi.us.es/docs/informes/LSI-2005-01.pdf>.
7. **Napal, Irina.** *Las métricas de software en la actualidad. Su necesidad de aplicación.* III Conferencia Científica de la Universidad de las Ciencias Infomáticas, s.l. : 2007.
8. **Aguero, Ing.Dennis Neuland.** *Áreas del aseguramiento de la calidad.* [En línea] [Citado el: 17 de 2 de 2011.] [http://calisoft.uci.cu/tmp/documentos/articulo\\_sga.pdf](http://calisoft.uci.cu/tmp/documentos/articulo_sga.pdf).
9. **Pressman, R, S.** *Ingeniería del software un enfoque práctico.* 1997.
10. **IEEE Standard Glossary of Software Engineering Terminology.** [En línea] [Citado el: 24 de 2 de 2011.] <http://ieeexplore.ieee.org/xpl>.
11. **Pardo, César Jesús.** *Análisis comparativo del estándar ISO 9000-3 con las subcaracterísticas de calidad de la ISO 9126.* s.l. : Escuela Superior de Informática Ciudad Real, 2009.
12. **Bravo, Ing. alexander Ore.** *CalidadSoftware.com.* [En línea] 13 de Abril de 2008. [Citado el: 17 de Octubre de 2011.] <http://www.calidadsoftware.com>.
13. **ESCALONE, L.F.** *ESTUDIO COMPARATIVO DE LOS MODELOS Y ESTANDARES DE CALIDAD DEL SOFTWARE.* UNIVERSIDAD TECNOLOGICA NACIONAL,FACULTAD REGIONAL BUENOS AIRES : s.n., 2006.
14. **IEEE.** *Computer Dictionary.* Computer Society. 1990.
15. **Piattini, Mario G.** *Analisis y diseño de aplicaciones infomaticas de gestión, una perspectiva de ingeniería del software.* 2004.
16. **PÚBLICAS.** *Implantación y Aceptación del Sistema.* [En línea] [Citado el: 10 de 2 de 2011.] <http://www.csi.map.es/csi/metrica3/iasproc.pdf>.
17. **Lores Sánchez, Lianet and Monne Roque, Diana.** *Aplicación de las pruebas de liberación al Sistema Informático De Genética Médica.* Universidad de las Ciencias Informáticas,La Habana : Trabajo de Diploma, 2009.
18. *Evaluating Load Generation in Virtualized Environments for Software Performance Testing.* **Marco A., Netto,Suzane Menon, Hugo V.** Porto Alegre, Brazil : s.n., 2011.
19. **Soriano, Amelia.** *Tipos de Prueba.* [En línea] [Citado el: 28 de 2 de 2011.] [http://carolina.terna.net/ingsw3/datos/Tipos\\_Prueba.pdf](http://carolina.terna.net/ingsw3/datos/Tipos_Prueba.pdf).
20. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El proceso Unificado de Desarrollo de Software.* [aut. libro] Félix Varela. La Habana : s.n., 2004.págs. 1-12.Vol. 1.
21. **Ramirez, Yilen Pons.** *Procedimiento para pruebas de eficiencia bajo carga intensiva en aplicaciones web de salud.* La Habana : s.n., 2007.
22. **Sommerville, IAN.** *Software Engineering 8va Edition.*
23. **Barreda, Ing. Liannis Soria.** *Especificación de Requisitos de Software.* s.l. : Versión 1.0, 2010.

24. **Juan Gualberto Pérez López, Dairon Quiñones Lazo.** *Procedimiento para diseñar e implementar pruebas no funcionales en los proyectos de la UCI.* 2010.
25. **Almenares, Liudmila Sánchez.** *Prueba Automática de Carga y Estrés en el Proyecto CICPC.* La Habana : s.n., 2008.
26. **Anay Caridad Barban Frómeta, Paula Zenaida Hernández Figueredo.** *Procedimiento para pruebas de rendimiento de Carga y Estrés al Sistema Único de Aduanas.* La Habana : s.n., 2009.
27. **Ariannis Marzán Matos, Juan Carlos Villar Bravo.** *Procedimiento para pruebas de software con herramientas.* 2010.
28. **Dayana Vítores Carmona, Yanisleydis Pelier Escalona.** *Automatización de pruebas de carga y estrés del Sistema de Perforación de Pozos Petroleros.* 2010.
29. **Qinglin Wu, Department of Computer Sciences Yunyang Teachers.** *Performance Testing and Optimization of J2EE-based Web Applications.* China : s.n., 2010.
30. **Marco A. Netto, Suzane Menon, Hugo V. Vieira.** *Evaluating Load Generation in Virtualized Environments for Software Performance Testing.* Porto Alegre, Brazil : Faculty of Informatics Dell Inc., 2011. IEEE International Parallel & Distributed Processing Symposium.
31. *Soluciones Informáticas Globales.* **Ladreda, Fernandez.** 10127, 2007.
32. **Pérez, P.G.** Principios básico del desarrollo seguro. [En línea] 2005. [Citado el: 23 de 1 de 5.] [http://germininus.com/sala\\_prensa/articulos/ppos%20desarrollo%20seguro.pdf](http://germininus.com/sala_prensa/articulos/ppos%20desarrollo%20seguro.pdf).
33. **MAñAS, J.A.** Pruebas de Programas. [En línea] 1994. [Citado el: 23 de 1 de 2011.] <http://www.it.uc3m.es/tsps/testing.htm#s1>.
34. **IEEE.** "Metrics", *IEEE.* 1991. Std1995.
35. **Espinoza Bárbara, Quinta Vanessa, Vegas Alexandra.** *Pruebas de Rendimiento.* 2005.
36. **(AVL). Vega Jara, y otros.** *De la prueba de carga.* 2004.
37. **Cárdenas, Lida Rodríguez.** *Estrategia de pruebas a la hipermedia sistemas de inteligencia social(SIS).* 2007.
38. **Pfleeger, S &Fenton N.** *PWS Publishing Company.* 1997.
39. **Carralero Mulet, Raúl. Cedeño Gonzáles, Fabian Felipe.** *Propuesta de un proceso de pruebas basadas en Rup para Proyectos de Gestión.* 2007.
40. **Varela, Félix.** *El Proceso Unificado de Desarrollo de Software.* La Habana : págs. 283-288. Vol. 1, 2004.
41. **J.A.** Pruebas de Programas. [En línea] 1994. [Citado el: 2 de 2 de 2011.] <http://www.it.uc3m.es/tsps/testing.htm#s1>.
42. **IBM.** [En línea] [Citado el: 2 de 2 de 2011.] <http://www-306.ibm.com/software/rational/>.
43. **Moreno G, Juan C.** *Análisis de Requisitos.* [En línea] 2008. [Citado el: 3 de 2 de 2011.] <http://www.slideshare.net/jcgmoreno/tema-1-ingenieria-de-requisitos>.
44. **SQS.** Software Quality Systems S.A. . [En línea] 2007. [Citado el: 7 de 3 de 2011.] <http://www.sqs.es/es/>.
45. **CES.** Centros de Ensayos de Software. [En línea] 2004. [Citado el: 7 de 3 de 2011.] <http://www.ces.com.uy>.
46. **Boehm.** *Ingeniería de Requerimientos-Ingeniería de Software.* 1979.
47. **Almaguer, Yaima Llaugert Mitjans y Yusdelky Meriño.** *Propuesta de instrumentos y herramientas para la evaluación del aprendizaje y su aplicación e integración en la plataforma para la gestión del aprendizaje ZERA.* 2010.

---

**ANEXOS****Anexo1: Encuesta**

1. ¿Considera necesario el desarrollo de un procedimiento para evaluar la satisfacción del requisito no funcional de rendimiento?

\_\_\_\_\_Muy necesario \_\_\_\_\_ Necesario \_\_\_\_\_ No es necesario

2. ¿Considera adecuada la estructura del procedimiento?

\_\_\_\_\_Muy adecuada \_\_\_\_\_ Adecuada \_\_\_\_\_ No se adecua

3. ¿El procedimiento es de fácil comprensión y aplicación?

\_\_\_\_\_Muy fácil \_\_\_\_\_ Fácil \_\_\_\_\_ Difícil

4. ¿Qué grado de importancia le confiere a las actividades que se definieron en el procedimiento?

\_\_\_\_\_Muy importante \_\_\_\_\_ Importante \_\_\_\_\_ No es importante

5. ¿Las actividades definidas se ajustan a los objetivos de un procedimiento para pruebas de rendimiento?

Se ajustan en un:

\_\_\_\_\_100%\_\_\_\_\_ 50%\_\_\_\_\_ 0%

6. ¿Considera bien distribuidas las actividades según las necesidades del procedimiento?

\_\_\_\_\_Muy bien \_\_\_\_\_ Bien \_\_\_\_\_ Mal

7. ¿Cree que los roles elegidos sean los necesarios?

\_\_\_\_\_Si\_\_\_\_\_ No \_\_\_\_\_ Otros que recomienda

8. ¿Se describen correctamente los artefactos de entrada y salida en cada una de las fases?

\_\_\_\_\_Muy bien \_\_\_ Bien \_\_\_\_\_ Mal

9. ¿Con la utilización del procedimiento se logrará desarrollar un proceso de pruebas con alto grado de calidad?

Será eficiente en un:

\_\_\_x\_\_\_ 100% \_\_\_\_\_ 50% \_\_\_\_\_ 0%

Sugerencias o recomendaciones:

### **Anexo2: Caso de Prueba 2**

**Nombre del Proyecto:** Gestión Integral de Aduanas.

**Nombre Módulo:** Tablas de Control.

**Versión:** 1.0

**Tipo de prueba:** Prueba de Carga.

Funcionalidad(es)	Comentario
Adicionar o Dar Alta a un Registro	Se selecciona la tabla en la que se va a Dar Alta, luego se adiciona el registro en caso de que su código no aparezca, sino solo sería poner el registro vigente. No se aceptan dos registros con el mismo Id.

**Variables:**

Variables	Valores
<b>Número de usuarios concurrentes</b>	50
<b>Tiempo entre conexión y conexión</b>	0
<b>Número de iteraciones</b>	1

Variables	Valores
Número de usuarios concurrentes	100
Tiempo entre conexión y conexión	0
Número de iteraciones	1

Variables	Valores
Número de usuarios concurrentes	150
Tiempo entre conexión y conexión	0
Número de iteraciones	1

### Anexo 3: Caso de Prueba 2

**Nombre del Proyecto:** Gestión Integral de Aduanas.

**Nombre Módulo:** Tablas de Control.

**Versión:** 1.0

**Tipo de prueba:** Prueba de Estrés.

Funcionalidad(es)	Comentario
Adicionar o Dar Alta a un Registro	Se selecciona la tabla en la que se va a Dar Alta, luego se adiciona el registro en caso de que su código no aparezca, sino solo sería poner el registro vigente. No se aceptan dos registros con el mismo Id.

**Variables:**

Variables	Valores
-----------	---------

<b>Número de usuarios</b>	60
<b>Tiempo entre conexión y conexión</b>	4
<b>Número de iteraciones</b>	10

<b>Variables</b>	<b>Valores</b>
<b>Número de usuarios</b>	90
<b>Tiempo entre conexión y conexión</b>	3
<b>Número de iteraciones</b>	7

<b>Variables</b>	<b>Valores</b>
<b>Número de usuarios</b>	120
<b>Tiempo entre conexión y conexión</b>	2
<b>Número de iteraciones</b>	5

#### **Anexo 4: Caso de Prueba 3**

**Nombre del Proyecto:** Gestión Integral de Aduanas.

**Nombre Módulo:** Tablas de Control.

**Versión:** 1.0

**Tipo de prueba:** Prueba de Carga.

<b>Funcionalidad(es)</b>	<b>Comentario</b>
Eliminar o Dar Baja a un Registro	Se selecciona la tabla donde se desea Dar Baja o dejar un registro sin vigencia, se selecciona el registro y se le pone fecha fin, debe ser mayor o igual que la de inicio. El registro no desaparece de la tabla.

**Variables:**

Variables	Valores
Número de usuarios concurrentes	50
Tiempo entre conexión y conexión	0
Número de iteraciones	1

Variables	Valores
Número de usuarios concurrentes	100
Tiempo entre conexión y conexión	0
Número de iteraciones	1

Variables	Valores
Número de usuarios concurrentes	150
Tiempo entre conexión y conexión	0
Número de iteraciones	1

### Anexo 5: Caso de Prueba 3

**Nombre del Proyecto:** Gestión Integral de Aduanas.

**Nombre Módulo:** Tablas de Control.

**Versión:** 1.0

**Tipo de prueba:** Prueba de Estrés.

Funcionalidad(es)	Comentario
Eliminar o Dar Baja a un Registro	Se selecciona la tabla donde se desea Dar Baja o dejar un registro sin vigencia, se selecciona el registro y se le pone fecha fin, debe ser mayor o igual que la de inicio. El registro

	no desaparece de la tabla.
--	----------------------------

**Variables:**

Variables	Valores
Número de usuarios	60
Tiempo entre conexión y conexión	4
Número de iteraciones	10

Variables	Valores
Número de usuarios	90
Tiempo entre conexión y conexión	3
Número de iteraciones	7

Variables	Valores
Número de usuarios	120
Tiempo entre conexión y conexión	2
Número de iteraciones	5

**Anexo 6: Caso de Prueba 4**

**Nombre del Proyecto:** Gestión Integral de Aduanas.

**Nombre Módulo:** Tablas de Control.

**Versión:** 1.0

**Tipo de prueba:** Prueba de Carga.

Funcionalidad(es)	Comentario
-------------------	------------

Modificar Registro	Se hace necesario señalar una tabla en el Árbol Tablas de Control y el registro o los registros que se quieren modificar, en caso de que sean varios registros estos se modificarían por campos.
--------------------	--

**Variables:**

Variables	Valores
<b>Número de usuarios concurrentes</b>	50
<b>Tiempo entre conexión y conexión</b>	0
<b>Número de iteraciones</b>	1

Variables	Valores
<b>Número de usuarios concurrentes</b>	100
<b>Tiempo entre conexión y conexión</b>	0
<b>Número de iteraciones</b>	1

Variables	Valores
<b>Número de usuarios concurrentes</b>	150
<b>Tiempo entre conexión y conexión</b>	0
<b>Número de iteraciones</b>	1

**Anexo 7: Caso de Prueba 4**

**Nombre del Proyecto:** Gestión Integral de Aduanas.

**Nombre Módulo:** Tablas de Control.

**Versión:** 1.0

**Tipo de prueba:** Prueba de Estrés.

Funcionalidad(es)	Comentario
Modificar Registro	Se hace necesario señalar una tabla en el Árbol Tablas de Control y el registro o los registros que se quieren modificar, en caso de que sean varios registros estos se modificarían por campos.

**Variables:**

Variables	Valores
Número de usuarios	60
Tiempo entre conexión y conexión	4
Número de iteraciones	10

Variables	Valores
Número de usuarios	90
Tiempo entre conexión y conexión	3
Número de iteraciones	7

Variables	Valores
Número de usuarios	120
Tiempo entre conexión y conexión	2
Número de iteraciones	5

**Anexo 8: Caso de Prueba 5**

**Nombre del Proyecto:** Gestión Integral de Aduanas.

**Nombre Módulo:** Tablas de Control.

**Versión:** 1.0

**Tipo de prueba:** Prueba de Carga.

Funcionalidad(es)	Comentario
Buscar Registro	La pantalla Búsqueda permite realizar búsquedas dentro de las tablas especificando datos, nos muestra todos los registros que coinciden con los elementos especificados.

**Variables:**

Variables	Valores
Número de usuarios concurrentes	50
Tiempo entre conexión y conexión	0
Número de iteraciones	1

Variables	Valores
Número de usuarios concurrentes	100
Tiempo entre conexión y conexión	0
Número de iteraciones	1

Variables	Valores
Número de usuarios concurrentes	150
Tiempo entre conexión y conexión	0
Número de iteraciones	1

### Anexo 9: Caso de Prueba 5

**Tipo de prueba:** Prueba de Estrés.

Funcionalidad(es)	Comentario
Buscar Registro	La pantalla Búsqueda permite realizar búsquedas dentro de las tablas especificando datos, nos muestra todos los registros que coinciden con los elementos especificados.

**Variables:**

Variables	Valores
Número de usuarios	60
Tiempo entre conexión y conexión	4
Número de iteraciones	10

Variables	Valores
Número de usuarios	90
Tiempo entre conexión y conexión	3
Número de iteraciones	7

Variables	Valores
Número de usuarios	120
Tiempo entre conexión y conexión	2
Número de iteraciones	5

**Anexo 10: Resultado de la funcionalidad Modificar Registro para 15 usuarios**

Petición	Muestras	Media	Mediana	Min	Max	Línea 90 %	%Error	Rendimiento	Kb/s
Modificar Registro	765	117	54	9	2912	277	0.00	105.0 /s	124.7

**Anexo 11: Criterios de Criticidad**

Los Criterios de Criticidad establecen los parámetros para declarar un producto del desarrollo de software en estado crítico de terminación.

Son aplicables en 3 momentos fundamentales:

- Cuando se revisa la solicitud de pruebas.
- Antes de comenzar las iteraciones de pruebas.
- Antes de Evaluar las pruebas.

**Prueba Suspendida (PS):** Se detiene la prueba y se reinicia en la misma actividad que se detuvo. Como máximo una prueba puede estar detenida 10 días, depende de los motivos que hicieron que se parara y debe cumplirse por parte del equipo de desarrollo. En caso de que no se cumpla el tiempo, se declara Prueba Abortada.

Criterios de Criticidad:

- Se presentan todos los artefactos necesarios.
- Está lista la última versión del artefacto a probar.
- Se resolvieron las NC detectadas.
- Están presentes todos los componentes del sistema (hardware, software).
- Están claros los requisitos de rendimiento.
- No excede el producto de una iteración.
- Se cumple con lo pactado en el Plan de Pruebas.
- No excede el tiempo total de la prueba según lo planificado en el cronograma.