

Universidad de las Ciencias Informáticas
Facultad 3



**Título: Análisis de la eficiencia para la solución Control
de Personas de la Aduana General de la República
de Cuba**

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Autor: Yasmín Álvarez Rodríguez

Tutores: Ing. Yisel Rodríguez Pérez

Ing. Alain Rodríguez Arias

Ciudad de La Habana, Junio 2011

“Año 53 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Yasmin Alvarez Rodríguez

Yisel Rodríguez Pérez

Alain Rodríguez Arias

Firma del Autor

Firma del Tutor

Firma del Tutor

AGRADECIMIENTOS

A mis padres Ana y Ramón por apoyarme siempre, por darme ánimo en los momentos que más lo necesité, por ser mis guías y ejemplo, por confiar en mí y quererme tanto. Por cuidar a mi pequeño bebé desde que nació para que yo pudiera terminar mis estudios, y hacerlo como si fuera otro hijo más, los amo con todo mi corazón y realmente no encuentro las palabras que describan lo agradecida que estoy a la vida por tenerlos.

A mis hermanos Ernesto y Maité por ser como son, por quererme tanto y ayudar a mis padres en la ardua tarea que les tocó llevar, los quiero con todo mi corazón.

A mi pequeño Marlon que lo amo con locura y es mi fuerza para seguir luchando hasta el último momento, te amo bebé.

A mis abuelitos Lule, Ramón y Leinides, por ser los mejores abuelos del mundo, los quiero mucho.

A mi bisabuelita Valentina que ya no está conmigo pero vive en mi recuerdo.

A mis tías y a mis primos por siempre estar ahí cuando más los necesité.

A mi familia querida, sobre todo a los que han estado siempre ahí para mí y me han demostrado su cariño, los quiero.

A mis queridos vecinos por considerarme su hija, por ser tan lindos y quererme tanto, los adoro (Yaya, María, Rafe, Gloria, en fin a todos).

A mis otras hermanitas Lala e Idalmita por luchar con mi tito.

A mi novio por ser lo mejor y más lindo que me ha pasado, por respetarme y apoyarme siempre, por estar a mi lado en los momentos buenos y malos, por su preocupación y ayuda, te quiero mucho.

A todos mis amigos y compañeros de la universidad, con los que he podido compartir momentos inolvidables, los quiero y siempre estarán presentes aún en la distancia.

En especial a Lila, Yíya, Anne, Leo, Say, Leydis y Made, en fin a todos los quiero muchísimo y gracias por todo, gracias por ser mis amigos.

A Yisel, mi tutora por su ayuda y por todos los consejos brindados, gracias.

DEDICATORIA

Dedico esta tesis a mis padres Ana y Ramón con todo el amor del mundo, si he llegado hasta aquí hoy es gracias a su esfuerzo, amor y confianza. Muchas gracias por estar siempre a mi lado y no dejarme en los momentos más difíciles. Son los mejores padres del mundo, gracias por darme la vida y hacerme sentir la persona más feliz del mundo por tenerlos.

A mis hermanos, mi bebé y a todos los que de una forma u otra me han ayudado en algún momento de mi vida.

RESUMEN

En la actualidad existe un gran avance en cuanto a las ciencias informáticas y las comunicaciones donde cada día adquiere más relevancia en la vida de las personas y en las empresas. Por tal motivo la Aduana General de la República de Cuba (AGR) en conjunto con la Universidad de las Ciencias Informáticas (UCI), se encuentran desarrollando el sistema informático Gestión Integral de Aduanas (GINA) donde han obtenido varios resultados. Específicamente este trabajo está estrechamente relacionado con uno de ellos, el Control de Personas (CP), realizado en el 2010. El cual tiene como objetivo fundamental gestionar y monitorizar a las personas que entran y salen del país, para así contribuir a la seguridad nacional.

En el presente trabajo se dejó documentado todo lo concerniente al proceso de pruebas para evaluar la eficiencia del módulo Control de Personas. Entre los tópicos que se trataron se encuentra una breve argumentación de los niveles, técnicas y métodos de prueba existentes, así como algunas herramientas para automatizar las pruebas con vista a determinar la más conveniente a utilizar en el proceso de pruebas a desarrollar, justificando su selección. Se utilizó para el desarrollo de las pruebas el procedimiento diseñado para el Departamento Soluciones para la Aduana específicamente para pruebas de carga y estrés, donde se diseñaron los casos de pruebas para todas las funcionalidades del módulo Control de Personas. También se especificaron los requisitos no funcionales de la aplicación donde fueron posteriormente validados por el cliente.

Finalmente se expusieron los resultados arrojados de las pruebas de carga y estrés. Realizándose un análisis evaluativo del proceso ejecutado para determinar el rendimiento del módulo Control de Personas, cuyo propósito fundamental fue evaluar la calidad del mismo una vez realizadas las pruebas, las cuales arrojaron resultados puntuales para dar una valoración al respecto y lograr que la dirección del Centro de Automatización para la Dirección y la Información de la Aduana General de la República (CADI) tome las medidas más convenientes.

PALABRAS CLAVE

Calidad, Pruebas de Software, Rendimiento, Requisito

TABLA DE CONTENIDOS

AGRADECIMIENTOS	II
DEDICATORIA	IV
RESUMEN.....	V
INTRODUCCIÓN	4
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	8
1.1 Introducción del Capítulo.....	8
1.2 Ingeniería de Requisitos.....	8
1.2.1 Requisitos	9
1.2.2 Etapas de la Ingeniería de Requisitos	9
1.2.3 Técnicas para la captura de requisitos	10
1.2.4 Técnicas de validación de requisitos	11
1.3 Calidad de Software	12
1.4 Pruebas de Software.....	13
1.4.1 Objetivo de las Pruebas de Software.....	14
1.4.2 Principios de las pruebas	14
1.5 Niveles de pruebas	15
1.5.1 Tipos de pruebas.....	17
1.6 Herramientas automatizadas para la realización de pruebas	20
1.7 Procedimiento estudiado para llevar a cabo el proceso de pruebas	22
1.8 Conclusiones parciales	22
CAPÍTULO 2: DISEÑO DE LA PROPUESTA	24
2.1 Introducción	24
2.2 Descripción del módulo Control de Personas.....	24
2.3 Captura y especificación de los requisitos no funcionales.	25
2.3.1 Requisitos especificados	25

2.3.2 Selección de la técnica de validación de los requisitos.....	28
2.4 Selección del Nivel de Prueba.....	28
2.4.1 Tipo de Prueba a realizar	28
2.5 Entorno de Prueba	29
2.5.1 Estudio de la topología de red	30
2.5.2 Selección de la herramienta	30
2.5.3 Elección de las variables	31
2.5.4 Criterio de aceptación de los resultados	32
2.6 Diseño y configuración de las pruebas en la herramienta	32
Figura 4: Configuración Final.	35
2.7 Funcionalidades de la solución Control de Personas	36
2.9 Conclusiones parciales	41
CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS	43
3.1 Introducción	43
3.2 Resultados de las pruebas de Cargas.....	43
3.2.1 Funcionalidad Buscar Persona API	43
3.2.2 Funcionalidad Concretar Propuesta	44
3.2.3 Funcionalidad Definir Propuesta.....	44
3.2.4 Funcionalidad Definir Repetidor	45
3.2.5 Funcionalidad Realizar Estudio API.....	45
3.2.6 Funcionalidad Gestionar PIA por Jefe LCF AGR.....	45
3.2.7 Funcionalidad Monitorizar Viajeros.....	46
3.2.8 Funcionalidad Mostrar Estudio API.....	46
3.2.9 Funcionalidad Mostrar Infracciones por Fecha	47
3.2.10 Funcionalidad Mostrar Infracciones por Persona.....	47
3.2.11 Funcionalidad Mostrar PIA	48
3.2.12 Funcionalidad Mostrar Posibles Redes de Vínculos	48
3.2.13 Funcionalidad Mostrar Propuestas	48
3.2.14 Funcionalidad Mostrar Requerimiento Puntual.	49
3.2.15 Funcionalidad Registrar Incidencia.....	49

3.2.16 Funcionalidad Insertar Requerimiento Puntual	49
3.2.17 Funcionalidad Ver Historial.....	50
3.2.18 Resultados generales de las pruebas de carga	50
3.3 Resultados de las pruebas de Estrés	55
3.3.1 Funcionalidad Buscar Persona API	55
3.3.2 Funcionalidad Concretar Propuesta	55
3.3.3 Funcionalidad Definir Propuesta.....	56
3.3.4 Funcionalidad Definir Repetidor	56
3.3.5 Funcionalidad Realizar Estudio API.....	56
3.3.6 Funcionalidad Gestionar PIA por Jefe LCF AGR	56
3.3.7 Funcionalidad Monitorizar Viajeros.....	56
3.3.8 Funcionalidad Mostrar Estudio API.....	57
3.3.9 Funcionalidad Mostrar Infracciones por Fecha	57
3.3.10 Funcionalidad Mostrar Infracciones por Personas	57
3.3.11 Funcionalidad Mostrar PIA	57
3.3.12 Funcionalidad Mostrar Posibles Redes de Vínculos	57
3.3.13 Funcionalidad Mostrar Propuesta	58
3.3.14 Funcionalidad Mostrar Requerimiento Puntual	58
3.3.15 Funcionalidad Mostrar Registrar Incidencia.....	58
3.3.16 Funcionalidad Insertar Requerimiento Puntual	58
3.3.17 Funcionalidad Ver Historial.....	58
3.3.18 Resultados generales de las pruebas de estrés	59
3.4 Características del servidor central de la AGR	61
3.5 Aportes obtenidos	61
3.6 Conclusiones parciales	62
CONCLUSIONES	64
RECOMENDACIONES.....	65
BIBLIOGRAFÍA.....	66
GLOSARIO	68

INTRODUCCIÓN

El desarrollo de software ha alcanzado un crecimiento vertiginoso en el mundo donde todas las empresas quieren producir aplicaciones informáticas con alta calidad, en el menor tiempo posible y a costos mínimos. Un software es de calidad cuando satisface las necesidades y expectativas del cliente o la empresa que lo adquiere.

Producir un sistema informático de calidad a un costo razonable trae beneficios para las partes involucradas, por lo tanto la calidad pasa a ser un atributo de mucha importancia para cualquier sistema informático.

Cuba no está ajena al desarrollo de software por lo que está llevando a cabo la informatización del país. Muchas empresas cubanas están solicitando productos informáticos para agilizar su trabajo y ganar en eficacia. Un ejemplo de ello se tiene en la Aduana General de la República de Cuba (AGR), la cual es una organización creada con el objetivo fundamental de garantizar la seguridad nacional. Con dicho propósito se crea el área de Lucha Contra el Fraude (LCF) encargada de enfrentar las acciones terroristas, de narcotráfico y las que ponen en riesgo el patrimonio cultural y natural del país. Entre las atribuciones y funciones especiales de la Aduana está la de prevenir, detectar y enfrentar el fraude comercial, el contrabando y otras infracciones aduaneras en el desarrollo del tráfico comercial de mercancías, viajeros y envíos por vía aérea, marítima y postal.

La AGR y la Universidad de las Ciencias Informáticas (UCI) están trabajando para lograr la informatización de la entidad aduanera desde el año 2004. Siguiendo este objetivo se crea el Proyecto Soluciones para la Aduana, el cual está desarrollando el sistema informático Gestión Integral de Aduanas (GINA), integrado por varios módulos, entre los que se encuentra el módulo **Control de Personas**. Dicho módulo es de vital importancia, pues permite realizar un estudio de la información adelantada de pasajeros, monitorizar el flujo de los viajeros por frontera y acceder a varios reportes que brindan información a los analistas de lucha contra el fraude en la toma de decisiones con mayor efectividad y certeza, entre los que se encuentra el historial de las personas que son de interés aduanal.

Actualmente el módulo Control de Personas está siendo probado en el Aeropuerto Internacional José Martí, sin embargo en el futuro se pretende usar este producto informático en todos los puntos aduanales del país, gracias en gran medida a la implantación del cable de fibra óptica en el territorio nacional.

Con la introducción del cable de fibra óptica pronosticada a implantarse a finales de julio del 2011, en la Aduana se decide no comenzar a desplegar el módulo en cada punto aduanal. Por parte de la dirección del proyecto en conjunto con el Centro de Automatización para la Dirección y la Información de la Aduana General de la República (CADI) se decide concentrar toda la información relacionada del módulo Control de Personas en un servidor central al cual se deberá poder acceder desde todos los puntos aduanales del país. Se desea conocer si al conectarse varios usuarios de todo el país la aplicación seguirá funcionando de manera eficiente, atendiendo a la rapidez del sistema, en cuanto a las transacciones procesadas, tiempo de respuesta al usuario y cantidad de clientes que pueden estar conectados simultáneamente.

Por otra parte no se cuenta con la especificación de los Requisitos no Funcionales (RNF), los cuales son propiedades o cualidades que el software debe cumplir, influyendo directamente en su calidad final. Los RNF determinan que el software sea atractivo, usable, rápido, confiable y eficiente. Concretamente reviste gran importancia el RNF Eficiencia, medido entre otros aspectos, por el rendimiento que debe poseer el sistema al ser accedido por varias conexiones simultáneas, la cantidad máxima de usuarios conectados que soportará el mismo y su tiempo de respuesta. Los resultados que se tienen de la eficiencia del módulo son los obtenidos de la puesta en práctica en el aeropuerto internacional José Martí, por lo que se desconocen los aspectos de la eficiencia mencionados anteriormente si el sistema es usado por un mayor número de usuarios situados en diferentes puntos del país. No se puede asegurar que el sistema funcione correctamente bajo condiciones extremas que puedan afectar su desempeño.

Por lo anteriormente planteado surge el siguiente **problema científico**: La validación realizada al módulo Control de Personas del sistema GINA no permite asegurar que el mismo funcione eficientemente en los entornos de aplicación.

Debido a lo planteado anteriormente se tiene como **objeto de estudio** la validación del requisito no funcional eficiencia.

Como **campo de acción** la validación del requisito no funcional eficiencia del módulo Control de Personas del sistema GINA de la Aduana General de la República de Cuba.

Para resolver el problema planteado se propone como **objetivo general** validar el requisito no funcional Eficiencia del módulo Control de Personas del sistema GINA.

Se tiene como **objetivos específicos**:

- ✓ Realizar el marco teórico de la investigación.
- ✓ Especificar los requisitos no funcionales del módulo Control de Personas.
- ✓ Validar el RNF Eficiencia mediante pruebas de rendimiento.
- ✓ Analizar los resultados de la solución propuesta.

Como **tareas investigativas**:

- ✓ Identificar los tipos de prueba más afines al análisis de calidad que se realizará.
- ✓ Estudiar las herramientas de pruebas automatizadas existentes.
- ✓ Realizar entrevistas a los funcionarios de la aduana.
- ✓ Especificar los requisitos no funcionales del módulo Control de Personas.
- ✓ Validar el RNF Eficiencia.
- ✓ Elaborar y ejecutar los casos de prueba necesarios para el análisis de calidad.
- ✓ Analizar los resultados arrojados en la investigación.

Durante el transcurso del trabajo de diploma y para su desarrollo se emplearon varios **Métodos de Investigación**, para dar cumplimiento a las tareas planteadas anteriormente, los cuales se exponen a continuación.

Métodos teóricos

Histórico - Lógico: Para disponer de información acerca de las tendencias actuales de la validación de requisitos no funcionales, además de la utilización de herramientas para realizar pruebas automatizadas.

Análisis-Síntesis: Para la consulta de la documentación existente acerca de la validación de requisitos y análisis de eficiencia mediante pruebas y el estudio de los conceptos empleados.

Métodos Empíricos

Entrevista: Se realizaron entrevistas con los analistas de la aduana para obtener la información necesaria, determinándose en ellas las características de la propuesta de solución teniendo en cuenta las necesidades de la misma, ayudando a obtener una mayor satisfacción con el producto realizado.

Estructura del documento

El presente trabajo se encuentra estructurado en tres capítulos:

Capítulo 1 Fundamentación teórica: Se realiza un estudio de los conceptos fundamentales relacionados con la calidad del software, la ingeniería de requisitos, los niveles de prueba, los tipos de pruebas por niveles, las herramientas para pruebas automatizadas de software, y los aspectos relacionados con el rendimiento para medir el nivel de eficiencia de un sistema.

Capítulo 2 Diseño de la propuesta: En el capítulo se realiza la especificación de los requisitos no funcionales de software para la solución Control de Personas. Se estudia el sistema a probar y se define el entorno para llevar a cabo las pruebas. También se generan los artefactos necesarios como el documento de especificación de requisitos de software y los diseños de casos de pruebas para la herramienta escogida. Se ejecutan las pruebas y se documentan.

Capítulo 3 Análisis de resultados: En este capítulo se analizan los resultados obtenidos al ejecutar las pruebas al módulo Control de Personas mediante gráficos, tablas e imágenes para llegar a conclusiones que facilitará la toma de decisiones.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción del Capítulo

En este Capítulo se brindan los conceptos fundamentales relacionados con la Ingeniería de Requisitos y la Calidad de Software, además de profundizar en conceptos como pruebas de software, niveles de pruebas y tipos de pruebas. Por otra parte se analizan algunas de las herramientas que existen para realizar pruebas automatizadas. El estudio realizado permite abordar el problema de investigación con una mejor preparación.

1.2 Ingeniería de Requisitos

La Ingeniería de Requisitos comprende el proceso de recopilar, analizar, entender, y lograr especificar con claridad las necesidades que tiene el cliente, para la posterior construcción del sistema. Un software al cual se le realice una correcta Ingeniería de Requisitos durante su ciclo de desarrollo, representa una solución adecuada para mitigar o minimizar los errores que pudiera presentar una vez finalizado.

El desafío de la Ingeniería del sistema y de los ingenieros del software es importante: ¿Cómo se puede asegurar que se ha especificado un sistema que recoge las necesidades del cliente y satisface sus expectativas? No hay una respuesta segura a esta difícil pregunta, pero un sólido proceso de Ingeniería de Requisitos es la mejor solución de que se dispone actualmente. (5)

La Ingeniería de Requisitos según Pressman está definida como la ayuda a los ingenieros de software para entender mejor el problema en cuya solución trabajarán. Incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, qué es lo que el cliente quiere y cómo interactuarán los usuarios finales con el software. (5)

Sommerville define la Ingeniería de Requisitos como el proceso de descubrir, analizar, documentar y verificar estos servicios y restricciones. (2)

Además plantea la Ingeniería de Requisitos como el proceso de desarrollar una especificación de software. Las especificaciones pretenden comunicar las necesidades del sistema del cliente a los desarrolladores del sistema. (2)

Este proceso es de gran importancia para el desarrollo de software y representa una etapa fundamental en el ciclo de desarrollo del software donde quedan definidas las características que debe tener el producto una vez finalizado.

1.2.1 Requisitos

La IEEE, acrónimo en inglés de Institute of Electrical and Electronics Engineers, define requisito como:

- I. Una condición o capacidad que necesita el usuario para resolver algún problema o alcanzar un objetivo.
- II. Condición o capacidad que debe cumplir o poseer un sistema o componente del sistema para satisfacer un contrato, estándar, especificación u otro documento formalmente impuesto.
- III. Una representación documental de una condición o capacidad como en I o en II.

Los requisitos pueden clasificarse en requisitos funcionales y requisitos no funcionales. (7)

Requisitos Funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. Estos describen los detalles de las funcionalidades del sistema, especifican las características de entrada y salida que debe cumplir el sistema a desarrollar.

Requisitos No Funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos los requisitos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requisitos funcionales, es decir una vez se conozca lo que el sistema debe hacer podemos determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser.

1.2.2 Etapas de la Ingeniería de Requisitos

El propósito de la ingeniería de requisitos es hacer que los requisitos alcancen un estado óptimo antes de alcanzar la fase de diseño en el proyecto, para ellos comprende todas las tareas relacionadas con la determinación de las necesidades o de las condiciones a satisfacer para un software nuevo o modificado. A continuación se exponen detalladamente las etapas de la ingeniería de requisitos las cuales son obtención, análisis, especificación, validación y verificación y por último gestión de requisitos. (2)

Obtención o elicitación: El primer paso consiste en comprender las necesidades y problemas del cliente. En la obtención se identifican todas las fuentes de requisitos implicadas en el sistema y en función de las características del entorno y del proyecto se emplean las técnicas más apropiadas para la identificación de las necesidades que deben satisfacerse.

Capítulo I: Fundamentación Teórica

Análisis: Una vez obtenida la información necesaria del entorno, es necesario sintetizarla, darle prioridades, analizar posibles contradicciones o conflictos, descomponer el sistema y distribuir las necesidades de cada parte, delimitar los límites del sistema y definir su interacción con el entorno.

Especificación: Cuando ya se conoce el entorno del cliente y sus necesidades, es necesario plasmarlas en forma de requisitos en los documentos que sirven de base de entendimiento y acuerdo entre cliente y desarrollador. Establecerán tanto la guía de desarrollo como los criterios de validación del producto final. En esta etapa documentar los requisitos es la condición más importante para gestionarlos correctamente.

Verificación y validación: El significado de estos dos términos genera confusiones habitualmente. El criterio básico que los diferencia es que verificación se refiere a la calidad formal, en este caso de los documentos de requisitos (no son ambiguos, no son incompletos, son posibles, verificables, etc.) y validación comprende la adecuación en el entorno de producción, en el caso de la documentación de requisitos, la conformidad por parte del cliente de que reflejan lo que él quiere. Los requisitos deben estar descritos de manera formal y deben ser técnicamente correctos (verificación), y satisfacer las necesidades del sistema, sin omitir ninguna ni incluir funcionalidades innecesarias (validación).

Gestión de Requisitos: Los requisitos cambiarán durante el desarrollo del sistema y es necesario poder trazar en cada cambio todas las partes afectadas, así como poder medir el impacto que cada modificación implica en la planificación del proyecto.

1.2.3 Técnicas para la captura de requisitos

La captura de requisitos es de gran importancia durante el proceso de concepción de un software, para la misma se dispone de un conjunto de técnicas que facilitan la obtención de los requisitos:

Entrevistas: Esta técnica consiste en establecer un canal de comunicación directo entre los clientes y el equipo de desarrollo. Las entrevistas son una forma de conversación, no de interrogación, y se pueden realizar sobre la base de un cuestionario rígido o de una guía más o menos detallada que las orienta hacia puntos bien definidos. Es la técnica de elicitación más utilizada, y de hecho es prácticamente inevitable en cualquier desarrollo. En las entrevistas se pueden identificar claramente tres fases: preparación, realización y análisis. (4)

Cuestionario: Consiste en un conjunto de preguntas presentadas a un grupo de personas para su respuesta. La forma de la pregunta puede influir en las respuestas, por lo que hay que planearlas cuidadosamente. Las preguntas suelen distinguirse en dos categorías: abiertas y cerradas. Las preguntas

Capítulo I: Fundamentación Teórica

abiertas permiten que los encuestados respondan con su propia terminología. Son especialmente útiles en la etapa exploratoria de la investigación, cuando se busca penetrar en el pensamiento del encuestado.

Sistemas existentes: Esta técnica consiste en analizar distintos sistemas ya desarrollados que estén relacionados con el sistema a ser construido. Esta técnica no necesita de la intervención del cliente, ya que mediante Internet se pueden buscar demos de productos que resultan similares y establecer contactos con profesionales que desarrollan sistemas de características comparables, aunque esto requiere de una profunda investigación y análisis. Es recomendable que se le muestre al cliente el sistema luego de haberlo analizado, ya que por su experiencia puede sugerir importantes ideas nuevas.

Tormenta de ideas: Esta técnica permite generar gran cantidad de ideas en breve tiempo. Se desarrolla con un grupo de expertos a los que se les expone un problema, o se les envía un memorándum previo. Las ideas se generan y exponen por los asistentes de forma clara y precisa, evitando discursos, sin que medie ninguna crítica o evaluación de éstas.

En esta atmósfera no crítica, las personas se sienten libres para decir lo que piensan y estas ideas, aún en el caso de que no tuvieran valor, pueden dar origen a otras por asociación. Las ideas se recogen y listan en papeles que se mantienen a la vista de todos para ser valoradas posteriormente.

Grabaciones de video y de audio: Existen dos formas de utilizar las grabaciones: registro y apoyo de las entrevistas para analizar algún proceso en particular. En cuanto a su función de apoyo, son importantes ya que permiten centrar la atención en la entrevista en sí, en vez de distraerse tomando notas de todo lo que se dice. Esta técnica permite analizar los temas con más detenimiento y con una visión más global. Brinda la posibilidad de ver y analizar en detalle ese proceso la cantidad de veces que sea necesario.

1.2.4 Técnicas de validación de requisitos

Revisiones: Esta técnica consiste en la lectura y corrección de la completa documentación o modelado de la definición de requisitos. Con la aplicación de esta técnica solo se puede validar la correcta interpretación de la información transmitida.

Auditorías: Las auditorías se realizan mediante el chequeo de los resultados que se tienen contra una lista de chequeo que se tenga predefinida o que se haya definido al inicio del proceso, es decir, se chequea cada requisito formulado con ayuda de un cuestionario.

Prototipos: Algunas propuestas se basan en obtener de la definición de requisitos sin tener la totalidad de la funcionalidad del sistema, esto permite al usuario hacerse una idea de la estructura de la interfaz del sistema con el usuario. (6)

Un prototipo constituye la versión inicial de un sistema, es la presentación de la interfaz del producto. Representa las funcionalidades que el cliente desea que cumpla su sistema, pero solo es una representación inicial de la solución que se necesita. El uso de prototipos es una técnica bastante útil, mediante su uso se puede obtener mayor retroalimentación en la captura de requisitos evitando futuros cambios.

1.3 Calidad de Software

La Calidad de Software es un factor esencial a la hora de un producto ser aceptado por el cliente. Muchos productos son rechazados por la falta de calidad que presentan. Es importante que los sistemas sean probados antes de pasar a manos del cliente, evitando la inconformidad ante posibles fallos y siendo menos las ineficiencias que presenten los sistemas.

Según (Pressman) la calidad es la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados, y con las características implícitas que se espera de todo software desarrollado profesionalmente. (8)

El Instituto de Ingenieros Eléctricos y Electrónicos definió la calidad de software como: “grado con el que un sistema, componente o proceso cumple con los requerimientos especificados y las necesidades o expectativas del cliente o usuario”. (9)

La calidad del software siguiendo la norma ISO 9126 describe seis características compuestas: Funcionalidad, Fiabilidad, Usabilidad, Eficiencia, Mantenimiento y Portabilidad.

- ✓ Funcionalidad: Capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.
- ✓ Fiabilidad: Capacidad del producto software para evitar fallar como resultado de fallos en el software
- ✓ Usabilidad: Capacidad del producto software que permite al usuario entender si el software es adecuado y cómo puede ser usado para unas tareas o condiciones de uso particulares.

Capítulo I: Fundamentación Teórica

- ✓ Eficiencia: Capacidad del producto software para proporcionar tiempos de respuesta, tiempos de proceso y potencia apropiados, bajo condiciones determinadas.
- ✓ Mantenimiento: Es la capacidad del producto software para serle diagnosticadas deficiencias o causas de los fallos en el software, o para identificar las partes que han de ser modificadas.
- ✓ Portabilidad: Capacidad del producto software para ser adaptado a diferentes entornos especificados, sin aplicar acciones o mecanismos distintos de aquellos proporcionados para este propósito por el propio software considerado.

La norma plantea diferentes aspectos de la calidad:

- ✓ Interna: Medible a partir de las características como el código fuente.
- ✓ Externa: Medible en el comportamiento del producto mediante pruebas.
- ✓ En uso: Durante la utilización efectiva por parte del usuario.

Esta investigación está centrada en verificar la calidad externa del software mediante pruebas para verificar la característica eficiencia que propone la norma ISO 9126.

1.4 Pruebas de Software

Las pruebas de software constituyen un paso fundamental para controlar la calidad de cualquier software. Mediante ellas se encuentran las deficiencias y los errores que presenta el sistema una vez finalizado. No se puede afirmar que aún ejecutando la prueba más completa se esté al margen de la presencia de errores. Las pruebas deben demostrar la presencia de errores, no su ausencia.

“Las pruebas constituyen una 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.” (9)

La prueba de software es uno de los procesos fundamentales dentro del control de calidad del software. “Consiste en la ejecución de un programa bajo ciertos datos de entrada (casos de prueba), para posteriormente comparar las salidas obtenidas con las deseadas (función para la cual fue diseñado)”. (10)

Las pruebas son una parte fundamental en el proceso de desarrollo de software, pues a partir de ellas es posible controlar que los productos cumplan con los requisitos funcionales y que estos funcionen correctamente, además de estos también se pueden probar requisitos no funcionales como fiabilidad,

facilidad de uso, eficiencia, funcionalidad, mantenimiento y portabilidad además de garantizar la calidad de estos productos.

1.4.1 Objetivo de las Pruebas de Software

El objetivo de las pruebas de software es encontrar la presencia de errores. Glen Myers establece una serie de aspectos que se manifiestan como objetivos esenciales de las pruebas de software. La prueba es el proceso de ejecución de un programa con la intención de descubrir un error.

- ✓ Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces
- ✓ Una prueba tiene éxito si descubre un error no detectado hasta entonces.
- ✓ La prueba no puede asegurar la ausencia de defectos, solo pueden mostrar que existen.

El objetivo es diseñar pruebas que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y de esfuerzo. (11)

1.4.2 Principios de las pruebas

Las pruebas se rigen por una serie de principios mostrados a continuación:

- ✓ Hacer un seguimiento de las pruebas hasta los requisitos del cliente (trazabilidad).
- ✓ Deben ser planificadas antes de que empiecen.
- ✓ Plantear y diseñar las pruebas antes de generar ningún código.
- ✓ El principio de Parapeto es aplicable a las pruebas de software (“Donde hay un defecto, hay otros”).
- ✓ Empezar las pruebas en módulos individuales y avanzar hasta probar el sistema entero, empezar por “lo pequeño” y progresar hacia “lo grande”.
- ✓ No son posibles las pruebas exhaustivas.
- ✓ No deben realizarse planes de prueba suponiendo que prácticamente no hay defectos en los programas y, por tanto, dedicando pocos recursos a las pruebas.

Una buena comprensión de estos facilitará el posterior uso de los métodos en un efectivo diseño de casos de prueba. (12)

1.5 Niveles de pruebas

En los niveles de prueba se definen los tipos y métodos de prueba que se deben realizar en cada uno de ellos. Cada prueba debe ser ejecutada en un determinado momento del ciclo de desarrollo del software con un objetivo específico. Cada nivel tiene un conjunto de pruebas definidas.

Unidad

Estas pruebas tienen gran importancia, ya que los resultados obtenidos de ellas repercutirán directamente en la ejecución de las demás pruebas.

Las pruebas de unidad son comprobaciones sobre las unidades más pequeñas del sistema, los paquetes, módulos, subsistemas y funciones. Se verifica que una unidad funciona correctamente por sí misma sin tener en cuenta las relaciones que pueda tener con otras partes del sistema.

Las pruebas unitarias aseguran que un único componente de la aplicación produce una salida correcta para una determinada entrada. Este tipo de pruebas validan la forma en la que las funciones y métodos trabajan en cada caso particular. Se encargan de un único caso cada vez, lo que significa que un único método puede necesitar varias pruebas unitarias si su funcionamiento varía en función del contexto.

Estas pruebas no descubrirán todos los errores del código, estas sólo prueban que el segmento de código que se revisa está lógicamente bien. Permiten fomentar el cambio y realizar pruebas sobre los cambios, asegurándose de que estos no presenten errores. Al llegar a la fase de integración se tiene mayor seguridad de que el código funcione correctamente. Estas son algunas de las características que presentan las pruebas de unidad.

Integración

Los objetivos de las pruebas se centran en identificar errores introducidos por la combinación de programas probados unitaria y/o modularmente, determinar cómo la base de datos de prueba será cargada, verificar que las interfaces entre las entidades externas (usuarios) y las aplicaciones funcionan correctamente, verificar que las especificaciones de diseño sean alcanzadas y determinar el enfoque para avanzar desde un nivel de integración de las componentes al siguiente. Este tipo de prueba maneja dos conceptos fundamentales que a continuación se detallan.

- **Integración descendente (Top-Down)**

Se integran los módulos moviéndose hacia abajo por la jerarquía de control. Comenzando por el módulo principal, los módulos subordinados se van incorporando a la estructura de una de estas dos formas: primero en profundidad, que integra todos los módulos de un camino de control principal de la estructura, o primero en anchura, que incorpora todos los módulos directamente subordinados a cada nivel, moviéndose por la estructura de forma horizontal.

- **Integración ascendente (Bottom-Up).**

Empieza la construcción y la prueba con los módulos de los niveles más bajos de la estructura del programa. Dado que los módulos se integran de abajo hacia arriba, el proceso requerido de los módulos subordinados a un nivel dado, siempre están disponibles y se elimina la necesidad de resguardos.

Sistema

Permite asegurar la apropiada navegación dentro del sistema, ingreso de datos, procesamiento y recuperación. Se ejecuta cuando el software está funcionando como un todo. Es la actividad de prueba dirigida a verificar el programa final, después que todos los componentes de software y hardware han sido integrados. Aceptable cuando el software se encuentra en la Fase de Construcción. En este nivel de prueba se desea comprobar que el sistema cumpla con los requisitos que están establecidos, el funcionamiento del sistema como un todo, la eficiencia y el rendimiento del mismo, además de poder verificar el tiempo de respuesta del sistema bajo determinadas condiciones. Para la generación de casos de prueba de sistema se utilizan técnicas de caja negra. Este tipo de pruebas se suelen hacer inicialmente en el entorno del desarrollador.

Aceptación

Constituye la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido. A veces se llama "Prueba Piloto". Se determina que el sistema cumple con lo deseado y se obtiene la conformidad del cliente. Se pueden distinguir las:

- **Pruebas alfa:** las realiza el usuario en presencia de personal de desarrollo del proyecto haciendo uso de una máquina preparada para tal fin.

- **Pruebas beta:** las realiza el usuario después de que el equipo de desarrollo les entregue una versión casi definitiva del producto.

1.5.1 Tipos de pruebas

En cada uno de los niveles de prueba mencionados anteriormente se realizan diferentes tipos de pruebas, en dependencia de cuán avanzado esté el desarrollo del software y qué aspecto de él sea necesario probar. Seguidamente se muestran algunos de los tipos de pruebas más importantes que existen.

Prueba de usabilidad: Prueba encaminada a factores humanos, estéticos, consistencia en la interfaz de usuario, ayuda sensitiva al contexto y en línea. Se determina además la calidad de la experiencia de un usuario en la forma en la que éste interactúa con el sistema, se considera la facilidad de uso y el grado de satisfacción del usuario.

Prueba de control de acceso: Se basa en dos aspectos como el nivel de seguridad de la aplicación que verifica que un actor solo pueda acceder a las funciones y datos que su usuario tiene permitido y el nivel de seguridad del sistema que verifica que solo los actores con acceso al sistema y a la aplicación están habilitados para accederla.

Prueba de especificación: Este tipo de prueba incluye probar la aplicación contra la documentación que se hizo antes, por ejemplo, que los procesos concuerden con los algoritmos hechos a papel, o que la aplicación tenga todas las funciones que se plantearon.

Prueba de documentación: Revisiones que se realizan a la documentación a partir de listas de chequeo. Se revisa además la correspondencia y continuidad de los documentos con los generados en etapas precedentes.

Prueba de regresión: Determina si los cambios recientes en una parte de la aplicación tienen efecto adverso en otras partes. Se realizan a partir de la 2da iteración de las pruebas, donde se comprueba que las no conformidades detectadas en la iteración anterior hayan sido respondidas y se prueba nuevamente la aplicación para identificar nuevos errores o defectos introducidos al arreglar los encontrados anteriormente.

Prueba funcional: Prueba centrada en validar las funciones que son objeto de prueba como lo que deben ser, ofreciendo los servicios, métodos o casos de usos requeridos. Deben enfocarse en los requisitos funcionales, las pruebas pueden estar basadas directamente en los Casos de Uso (o funciones de

Capítulo I: Fundamentación Teórica

negocio), y las reglas del negocio. Esta prueba es implementada y ejecutada contra diferentes objetos de pruebas, incluyendo unidades, unidades integradas, aplicaciones y sistemas.

Prueba de rendimiento: Enfocadas a monitorear el tiempo en el flujo de ejecución, acceso a datos, en llamada a funciones y sistema para identificar y direccionar los cuellos de botellas y los procesos ineficientes. Compara el rendimiento de un elemento nuevo o desconocido a uno de carga de trabajo de referencia conocido. Determinan los tiempos de respuesta, el espacio que ocupa el módulo en disco o en memoria, el flujo de datos que genera a través de un canal de comunicaciones, entre otros. Existen varios tipos de pruebas de rendimiento, las cuales son muy útiles, siempre y cuando sean usadas a través de herramientas que permitan visualizar los detalles técnicos y gráficos altamente informativos de los resultados de las pruebas. Entre los tipos de pruebas de rendimiento más importantes están las pruebas de cargas, estrés, picos y estabilidad.

Prueba de carga: Usada para validar y valorar la aceptabilidad de los límites operacionales de un sistema bajo carga de trabajo variable, mientras el sistema bajo prueba permanece constante. La variación en carga es simular la carga de trabajo promedio y con picos que ocurre dentro de tolerancias operacionales normales.

Prueba de estrés: Su principal meta es identificar y documentar las condiciones bajo las cuales el sistema falla. Verifica que el sistema funciona apropiadamente y sin errores, bajo condiciones de estrés (Extrema sobrecarga, insuficiente memoria, servicios y hardware no disponible, recursos compartidos no disponibles), cuando existe una memoria baja o no disponible en el servidor, en caso de que estén conectados demasiados clientes realizando las mismas transacciones con los mismos datos.

Prueba de pico: Como su nombre lo indica, observa el comportamiento del sistema variando el número de usuarios, tanto cuando baja como cuando sube drásticamente.

Prueba de estabilidad: Determina si la aplicación puede aguantar una carga esperada continua. Generalmente expone si hay una fuga de memoria en la aplicación informática.

Prueba de instalación: Enfocada a asegurar la instalación en diferentes configuraciones de hardware y software bajo diferentes condiciones (insuficiente espacio en disco, entre otros). Además de verificar y validar que el sistema se instala apropiadamente en cada cliente, bajo condiciones como instalaciones nuevas, nuevas máquinas a las que nunca se les ha instalado el sistema, al actualizar máquinas

Capítulo I: Fundamentación Teórica

previamente instaladas con el sistema o instalar versiones viejas en máquinas previamente instaladas con el sistema.

Prueba de recuperación y tolerancia de fallas: Verificar que los procesos de recuperación (manual o automática) restauran apropiadamente la base de datos, aplicaciones y sistemas, donde los llevan a un estado conocido o deseado. Se debe tener en cuenta aspectos como interrupción de electricidad en el cliente, interrupción de electricidad en el servidor, interrupción en la comunicación con los controladores de disco, ciclos incompletos (procesos de consultas interrumpidos, procesos de sincronización de datos interrumpidos).

Prueba modular: Este tipo de pruebas tiene como objetivo probar la funcionalidad de cada módulo por separado, lo que garantiza detectar y eliminar errores antes de la integración con otros módulos, además de medir la correspondencia entre lo documentado y las funciones que realmente fueron implementadas en el sistema.

Prueba de desarrollador: Prueba diseñada e implementada por el equipo de desarrollo. Estas pruebas pueden realizarse cruzando los programadores, analistas u otro rol, de forma tal que no solo sea la persona la que revise su propio trabajo, pues generalmente una persona ajena es la que más errores puede detectar. Tradicionalmente estas pruebas han sido consideradas solo para la prueba de unidad, aunque en la actualidad en algunos casos pueden ejecutar pruebas de integración. Se recomienda que estas pruebas cubran más que las pruebas de unidad.

Prueba interna: Prueba diseñada e implementada por el equipo de calidad interna del proyecto. Se pueden realizar pruebas de varios tipos, en dependencia de las características del proyecto en cuestión. Se recomienda que las pruebas se realicen con toda la profundidad requerida, revisando todos los artefactos que sean generados en el desarrollo del producto, para poder verificar que el software se está construyendo correctamente y una vez concluido, validar que el producto desarrollado sea el correcto, teniendo en cuenta los requisitos acordados con el cliente.

Prueba de configuración: Enfocada a asegurar que funciona en diferentes configuraciones de hardware y software. Esta prueba es implementada también como prueba de rendimiento del sistema.

Prueba de estructura: Este tipo de prueba se realiza a las aplicaciones web asegurando que todos los enlaces están conectados, el contenido deseado es mostrado y no hay contenido huérfano.

Capítulo I: Fundamentación Teórica

Prueba de volumen: Prueba centrada en verificar las habilidades de los objetos de prueba para manejar grandes cantidades de datos, tanto en entrada como en salida, o residente en la base de datos. Puede incluir un procedimiento que indique el uso de consultas que devuelvan todo el contenido de la base de datos, o cuando la cantidad de datos de entrada excede a la cantidad establecida de cada campo.

Prueba de seguridad: Prueba centrada en asegurar que los datos o sistemas que son objetos de prueba, son accedidos sólo por los actores que tienen permiso para hacerlo. Esta prueba es implementada y ejecutada contra varios objetos de prueba.

Prueba de validación: Son las pruebas realizadas sobre un software completamente integrado para evaluar el cumplimiento con los requisitos especificados.

Prueba de verificación: Se revisa si el resultado corresponde a la especificación del sistema, es decir, si se está construyendo el sistema de manera correcta.

Prueba de operación: Su objetivo es verificar el sistema en operación por un largo período bajo condiciones normales de uso.

Prueba de réplica: Esta prueba está dirigida a la Base de Datos y a probar que funcione correctamente. La enorme variedad de pruebas que existen están enfocadas a diferentes niveles del sistema y en diferentes fases de desarrollo del mismo, permitiendo que el producto obtenido sea lo más eficiente posible y presente la menor cantidad de errores.

1.6 Herramientas automatizadas para la realización de pruebas

La utilización de herramientas automatizadas es de gran importancia para la realización de las pruebas de software ya que permite probar el sistema sin la interacción de las personas, solo interactuando con la herramienta, además permiten una disminución de tiempo, esfuerzo y gasto de recursos. Permite probar de manera simple, rápida y eficiente el sistema sin necesidad de estar el probador atento todo el tiempo al sistema. Permiten generar miles de usuarios sin necesidad de usar este mismo número de computadoras, evitando grandes gastos de recursos innecesariamente para el país. A continuación se brindan una serie de herramientas automatizadas para realizar pruebas de software con sus principales características.

OpenLoad

Es la primera solución rápida de optimización de rendimiento basada en navegador, fácil de usar, para las pruebas de carga y estrés de aplicaciones y sitios web dinámicos. Puede capturar y traducir acciones del usuario de un sitio web. Genera hasta 1000 usuarios simultáneos con el hardware mínimo.

Necesita de la URL del sistema para acceder al mismo y al ser una herramienta que debe ser configurada como localhost se comporta como proxy. Es de libre acceso y adaptable a cualquier plataforma, brinda además al probador una amplia información una vez efectuada la prueba. (15)

OpenSTA

Es un software distribuido para probar arquitecturas. Su función principal es la prueba (HTTP y HTTPS) de ejecución y de carga/tensión. Esto permite crear diferentes patrones de uso en un sitio web, los que son grabados como scripts para ser fácilmente modificados y tornados dinámicos. Por consiguiente, estos scripts pueden ser ejecutados nuevamente en la arquitectura distribuida de OpenSTA para emular a miles de usuarios virtuales. OpenSTA puede ser utilizado para prueba de regresiones en páginas web. (16)

FunkLoad

Es una web funcional y probador de carga, escrito en Python, cuyo principal uso incluye una prueba de funcionamiento de proyectos web así como pruebas de regresión. También es capaz de realizar pruebas de rendimiento de su aplicación por la carga de la aplicación Web y el seguimiento de sus servidores que ayuda a identificar los cuellos de botella, lo que da un informe detallado de la medición del desempeño. FunkLoad también expone los errores en la superficie de prueba, al igual que el volumen de pruebas. (17)

WebLoad

Es una herramienta de código abierto para pruebas de carga, de tensión y de rendimiento. Se puede cargar a la prueba cualquier aplicación de Internet, incluidas las aplicaciones que utilizan la Web 2.0 y AJAX. WebLoad es un gran punto de partida para este proyecto, con una comunidad de más de 1.600 clientes y un historial establecido en cuanto al fondo de su robusta funcionalidad, arquitectura eficiente, extensibilidad, el apoyo, la calidad y su facilidad de uso. (18)

JMeter

JMeter es una herramienta Java desarrollada dentro del proyecto Jakarta, que permite realizar Pruebas de Rendimiento y Pruebas Funcionales sobre Aplicaciones Web. Permite la ejecución de pruebas distribuidas

entre distintos ordenadores, para realizar pruebas de rendimiento. Además activar o desactivar una parte de la prueba, lo que es muy útil cuando se está desarrollando una prueba larga y se desea deshabilitar ciertas partes iniciales que sean muy pesadas o largas. Permite generar un caso de prueba a través de una navegación de usuario.

JMeter como herramienta de prueba dispone de varios componentes que facilitan la elaboración de los escenarios de prueba con la ventaja de simular para cada uno de esos escenarios miles de usuarios. (19)

Wapt Pro

Wapt Pro es una herramienta para realizar pruebas de carga y estrés a una aplicación web, de fácil uso, consistente, permite analizar el rendimiento y encontrar cuellos de botellas según distintas configuraciones. Ofrece simulaciones precisas de la navegación realizada por un usuario, admite diferentes usuarios en una sola prueba, válido para aplicaciones dinámicas y contenidos HTTP/SSL y devuelve detallados informes y datos sobre las pruebas realizadas. (20)

1.7 Procedimiento estudiado para llevar a cabo el proceso de pruebas

Para llevar a cabo el proceso de prueba se estudió el procedimiento propuesto para la realización de pruebas de carga y de estrés para el Departamento Soluciones para la Aduana del año 2009, el cual cuenta con varios aspectos para organizar y especificar las actividades que serán realizadas para lograr exitosamente la realización de las pruebas.

La primera fase del procedimiento es la de Planificación, donde se realiza el estudio de la documentación, la elección de las variables, el estudio de la topología de red, el diseño del entorno de pruebas y el diseño de las pruebas en la herramienta JMeter. La segunda fase es la de Aseguramiento donde se verifica la carga de trabajo, se realiza la configuración y se toman las precauciones necesarias para proceder con las pruebas. La tercera fase es Ejecución, donde se ejecutan las pruebas y se validan los resultados. Por último la cuarta fase Análisis e interpretación de los resultados obtenidos. (22)

1.8 Conclusiones parciales

En este capítulo se han estudiado los conceptos asociados a la Ingeniería de Requisitos así como las etapas que propone la misma para el desarrollo de un software. Existen varias técnicas de captura de requisitos de las cuales se seleccionaron la entrevista y la tormenta de ideas para especificar los RNF del módulo Control de Personas. Todos los requisitos capturados serán validados mediante la técnica revisión

Capítulo I: Fundamentación Teórica

en conjunto con el cliente, el programador y la analista principal. De los RNF capturados, se validará el de Eficiencia, mediante pruebas de rendimiento de carga y estrés que fueron las seleccionadas. Este tipo de prueba se realiza en el nivel de sistema, que es donde se comprueba la eficiencia de un software ya que en este nivel es donde se realizan las pruebas de rendimiento.

Se estudiaron las herramientas automatizadas que existen para realizar pruebas de software, haciendo un enfoque en las utilizadas para realizar pruebas de rendimiento y seleccionando de estas la más acorde para esta investigación. La herramienta propuesta es el software JMeter por las características y funcionalidades que posee para desarrollar pruebas de carga y de estrés con un alto nivel de calidad.

CAPÍTULO 2: DISEÑO DE LA PROPUESTA

2.1 Introducción

En este capítulo se describen las características fundamentales del módulo Control de Personas. Se realiza la especificación de los requisitos no funcionales propuestos para la solución porque estos no estaban especificados en el documento de especificación de requisitos del sistema. Se describe el entorno de pruebas, las especificaciones de software y hardware, la topología de la red disponible para realizar el proceso de pruebas. Además se aplican un conjunto de actividades para la ejecución de las pruebas.

2.2 Descripción del módulo Control de Personas

Se muestra a continuación una breve descripción del módulo Control de Personas, lo cual permite conocer sus principales funcionalidades y las tecnologías usadas en su desarrollo:

El módulo Control de Personas pertenece al sistema Gestión Integral de Aduanas (GINA), el cual es un sistema web que se desarrolla sobre el sistema operativo GNU/Linux aunque es multiplataforma por lo tanto puede funcionar en los sistemas GNU/Linux y en Windows. El lenguaje de programación utilizado es PHP, el framework¹ de desarrollo Symfony y la base de datos Oracle. La interfaz de usuario se desarrolla con las librerías ExtJS para el lenguaje JavaScript.

La solución cumple con las siguientes funcionalidades: monitoriza en tiempo real a las personas al pasar por frontera aérea, gestiona las personas de interés aduanal y permite realizar el estudio de la información adelantada de pasajeros para un vuelo establecido por el usuario. El sistema permite acceder a los hechos históricos de las personas que son de interés para la Aduana además de configurar cuando se considera una persona repetidora teniendo en cuenta la cantidad de viajes en un tiempo determinado y no menos importante la salida de varios reportes, como tributar información que permita definir posibles redes de vínculos, ver el historial de las personas que representan interés aduanal, entre otros muy valiosos para el trabajo que deben desarrollar los usuarios del sistema.

¹ Framework: Palabra técnica para referirse al marco de trabajo a utilizar. Consiste en un software que facilita notablemente el trabajo de los desarrolladores y ahorra tiempo de desempeño. Este puede incluir bibliotecas, soluciones y otras aplicaciones.

2.3 Captura y especificación de los requisitos no funcionales.

Las técnicas de captura de requisitos que se utilizaron para la solución Control de Personas fueron la entrevista y la tormenta de ideas. Mediante reuniones con los desarrolladores del sistema, la analista principal y los clientes, se obtuvieron las características no funcionales que debe cumplir el sistema una vez finalizado. Se escogen estas dos técnicas ya que permiten un mayor flujo de información entre los involucrados. Brindan una mayor retroalimentación con los clientes evitando futuras inconformidades. Las técnicas utilizadas fueron de gran ayuda para la comprensión y análisis de los requisitos para su posterior documentación.

2.3.1 Requisitos especificados

La especificación de requisitos de software es de gran importancia para lograr un sistema informático con calidad. Dentro de los RNF que fueron especificados están: usabilidad, fiabilidad, eficiencia, soporte, seguridad, restricciones de diseño, interfaz y portabilidad. Seguidamente se exponen las especificaciones para cada uno de ellos, los restantes requisitos se encuentran en la plantilla de Especificación de Requisitos de Software propuesta por el Centro de Informatización y Gestión de Entidades (CEIGE). (Ver Anexo 1)

Usabilidad

El sistema desarrollado debe ser manipulado de manera sencilla y eficiente, siendo fácil para el usuario entender su funcionamiento en dos días con ayuda del manual de usuario. Debe satisfacer las necesidades del usuario brindando las funcionalidades que el cliente especificó. Debe permitir a los usuarios acceder a un menú donde encuentre de manera sencilla la funcionalidad que busque en un momento dado. El usuario debe poder interactuar con el módulo sin tener previa preparación, con los conocimientos que posee del negocio debe poder interactuar con el sistema. La página web se adaptará a la resolución de pantalla que tenga configurado el cliente. El idioma de todas las interfaces de la aplicación será el español y las etiquetas contendrán los términos significativos del concepto que deseen transmitir. El sistema deberá mostrar el menú general desde cualquiera de sus páginas.

Fiabilidad

- ✓ *Disponibilidad:* La aplicación debe ser capaz de estar operativa durante el mayor tiempo posible para brindar sus servicios a los usuarios ininterrumpidamente durante el tiempo que estos lo necesiten. El tiempo dedicado al mantenimiento de la aplicación no debe exceder en el día a 30

Capítulo II: Diseño de la Propuesta

minutos debido a que el mantenimiento a la misma se realiza en tiempo de producción utilizando un servidor de pruebas donde los administradores de configuración del CADI realizan la actualización de la solución en tiempo real de ejecución, o sea en tiempo de explotación del sistema.

- ✓ *Tiempo medio entre fallos:* El tiempo mínimo de disponibilidad ante posibles fallos para la solución Control de Personas no debe exceder los 5 minutos al día. Deben existir copias de seguridad y de recuperación ante posibles fallos, la tasa de fallos debe ser muy baja para este sistema debido a la importancia que representa para la seguridad nacional del país.
- ✓ *Tiempo medio de reparación:* La recuperación ante los fallos no debe estar por encima de 15 minutos), un posible fallo podría ser un error en la máquina de inmigración, por lo que no se puede monitorizar el cruce de fronteras.
- ✓ *Exactitud:* La precisión de las salidas del sistema deberán ser de 7 segundos en todos los casos desarrollados en el sistema Control de Personas.

Eficiencia

- ✓ *Tiempo de respuesta por transacción:* En el caso de los reportes se puede tardar hasta 10 segundos debido a que manejan gran cantidad de datos, para las funcionalidades no debe exceder los 5 segundos.
- ✓ *Rendimiento:* La cantidad de datos que pueden ser transferidos son 1000 transacciones por segundos. Entre más corto sea el procesamiento de la transacción, más alto será el rendimiento del sistema.
- ✓ *Capacidad clientes:* El sistema debe soportar hasta 100 clientes simultáneamente, debido que existen 12 aeropuertos internacionales, en el aeropuerto José Martí estarán como promedio 20 usuarios conectados y en los restantes puntos hasta 5 usuarios por cada uno, sumando un total de 55 usuarios. Con esta última cifra suman 75 conexiones, quedando libre 25 usuarios. Se garantiza que el sistema esté apto para ser utilizado por todas las aduanas y no debe existir riesgo de mal funcionamiento en caso de tener que soportar otros usuarios, dejando un margen de 25 capacidades por si es necesario.

Capítulo II: Diseño de la Propuesta

- ✓ *Utilización de recursos:* Las computadoras clientes deberán tener como mínimo 512 Mb de memoria RAM y pueden ser clientes ligeros. En caso del servidor 120 Gb de disco duro y 2 Gb de RAM con un microprocesador de Core 2 Duo a 2,2 MHz o de mayores prestaciones.

Soporte

La aplicación cliente debe ser capaz de correr sobre cualquier plataforma, en caso de que sea Linux se recomienda la versión Ubuntu. Para la parte servidora se recomienda que corra sobre plataforma Linux especialmente la distribución Ubuntu en las versiones LTS (Long Term Support). Se debe utilizar como navegador web el Firefox en una versión 3.5 o superior. Se deben incluir normas de codificación y bibliotecas de clases.

Seguridad

Este sistema debe tener un alto grado de seguridad debido a que maneja información de suma importancia y confidencial para la seguridad nacional. La solución debe reflejar patrones de seguridad teniendo en cuenta la alta sensibilidad de la información que maneja de acuerdo a las especificaciones funcionales dadas y a las políticas, normas y estándares de seguridad requeridas por el sistema.

- ✓ *Autenticación:* El sistema debe verificar la identidad de los usuarios y otras aplicaciones con las que se comunique el software.
- ✓ *Conexión a la base de datos:* No todos los usuarios tienen los mismos privilegios para interactuar con la información de la base de datos, debe existir un administrador, que será el encargado de asignar los permisos.
- ✓ *Roles:* El acceso a cada operación en el sistema debe ser en dependencia del rol que esté autenticado en ese momento.

Restricciones de diseño

Debe permitir reutilizar la información y evitar que se dupliquen datos, debido al volumen de ellos. El lenguaje de programación debe ser PHP 5.2.6 o superior con un gestor de base de datos Oracle 11g o PostgreSQL 8.x o superior a esta versión.

Interfaz

Los requisitos de interfaz deben estar acordes con la descripción de los requisitos funcionales del subsistema donde se detallan todos los requisitos de interfaz ya sea en el Documento de Especificación

de Requisitos Funcionales o en las Salidas del Sistema. Se debe utilizar las librerías ExtJs que logran un entorno amigable y aprovechan recursos en la interacción con el servidor. La interfaz debe usar tecnologías asincrónicas como AJAX y JSON para ganar en velocidad y lograr un mejor aprovechamiento de los recursos disponibles. La interfaz debe ajustarse automáticamente al tamaño y resolución del navegador web utilizado.

Portabilidad

Control de Personas es un sistema multiplataforma que puede correr sobre los sistemas operativos Windows y Linux.

2.3.2 Selección de la técnica de validación de los requisitos

La técnica usada para la validación de los requisitos no funcionales fue la revisión, mediante la lectura de la documentación existente se pudo observar que no existía un documento de especificación de requisitos no funcionales de software para la solución Control de Personas. En los documentos estudiados estaban descritos los requisitos funcionales solamente, haciendo imposible verificar si el sistema posee las propiedades o cualidades esperadas por los clientes. Se realizó la correcta interpretación de los requisitos obtenidos por los analistas, funcionarios del CADI y parte del equipo de desarrolladores del módulo, evitando la ambigüedad y logrando que estos sean claros y completos. Se obtuvo un acta de aceptación de los requisitos especificados (Ver Anexo 2).

Se validará el requisito no funcional eficiencia mediante pruebas de rendimiento al módulo Control de Personas.

2.4 Selección del Nivel de Prueba

Entre los niveles de pruebas existentes se elige el nivel de sistema para realizar las pruebas debido a que el módulo que se va a probar se encuentra funcionando como un todo. En este nivel es donde se realizan las pruebas de rendimiento de un software.

2.4.1 Tipo de Prueba a realizar

Existen numerosos tipos de pruebas de software dentro del nivel sistema. Para dar solución a la problemática se escogen las pruebas de rendimiento, dentro de ellas las de **carga** y **estrés**, para verificar la eficiencia del sistema; ya que para la puesta en práctica de la solución Control de Personas en todos los puntos aduanales del país, se necesita conocer las condiciones bajo las cuales podría fallar el sistema. El

Capítulo II: Diseño de la Propuesta

requisito no funcional Eficiencia, es verificable mediante estas pruebas, siendo este el motivo fundamental de su selección. Como se muestra en el presente trabajo, existen otros tipos de pruebas de rendimiento: las pruebas de pico, que verifican el desempeño de un software cuando sube y baja bruscamente el número de usuarios conectados, y las pruebas de estabilidad, que verifican el desempeño del sistema cuando mantiene un número de usuarios estable. La prioridad del presente trabajo es evaluar el módulo Control de Personas en situaciones de carga máxima, pues si el mismo va a ser usado en todos los puntos aduanales del país, se espera que aumente considerablemente la cantidad de usuarios concurrentes, por lo que una misma funcionalidad puede ser usada por varios usuarios al mismo tiempo, lo que implica que los mismos datos pueden ser accedidos o modificados simultáneamente.

2.5 Entorno de Prueba

Para la realización de las pruebas las características que debe tener el ambiente debe ser lo más parecido posible al entorno donde será utilizado el sistema. El sistema operativo para la computadora cliente donde se van a ejecutar las pruebas es Linux, en la distribución Ubuntu 11.04.

Los servidores tienen como gestor de base de datos Oracle Standard Edition ONE version 11g, están corriendo sobre el sistema operativo Linux. Es utilizada la máquina virtual de Java en su versión 1.3.

Dentro de las características de hardware de la computadora cliente se tiene que el microprocesador es Intel(R) Core(TM) 2 Duo. La capacidad del disco duro es de 160 GB, y la memoria RAM es de 1GB.

Especificaciones de Software y Hardware

Características de Software	
Sistema Operativo	Linux.
Gestor de Base de Datos	Oracle Standard Edition ONE version 11g
Máquina Virtual	Java Virtual Machine 1.3 o superior
Herramienta de prueba	JMeter

Características de Hardware	
Microprocesador	Intel(R) Core(TM)2 Duo
	CPU E4500 a 2.20 GHz
Memoria RAM	1 GB (PC Clientes)
	2GB (Servidores)
Ancho de Banda	10/100 Mbps UTP 5 LAM
Capacidad del disco duro para máquinas clientes y servidores	160 GB(PC Clientes)
	80 GB (Servidores)

Vale destacar que el entorno de prueba determina que la herramienta utilizada arroje resultados fiables, por lo que es de suma importancia tener esto en cuenta.

2.5.1 Estudio de la topología de red

La topología de red que tienen los laboratorios donde se desarrolla el Sistema de Gestión Integral de Aduanas (GINA) es la Topología de Árbol. La conexión en árbol es parecida a una serie de redes en estrellas interconectadas salvo que no tiene un nodo central. En cambio, tiene un nodo de enlace troncal, generalmente ocupado por un switch ó conmutador, desde el que se ramifican los demás nodos. La topología en árbol puede verse como una combinación de varias topologías en estrella. Las computadoras con las que se cuenta en el proyecto se conectan mediante fibra óptica y cable UTP (acrónimo en inglés de Unshielded Twisted Pair), ó Par Trenzado sin Blindaje, en un área de dos laboratorios de producción.

2.5.2 Selección de la herramienta

La herramienta seleccionada es el JMeter debido a que es una herramienta desarrollada bajo los estándares de software libre con la cual se puede trabajar fácilmente. Es una herramienta multiplataforma, lo que permite el desarrollo de pruebas en los sistemas operativos Linux y Windows. La versión 11.04 de Ubuntu donde se van a realizar las pruebas da la posibilidad de instalar desde su repositorio la herramienta. Puede manipular y almacenar las grabaciones para su reutilización en otras pruebas. Presenta una amplia documentación que facilita la interacción con la herramienta. Permite la capacidad de soportar múltiples hilos, generando miles de usuarios. Brinda gráficas de rendimiento, informes de los resultados de las pruebas y la posibilidad de ver los errores de la aplicación web a la que se le está

realizando las pruebas. Esta herramienta brinda resultados confiables, siempre que cuente con los recursos de la computadora necesarios para un buen funcionamiento, siendo la utilizada por la UCI en el Departamento de Pruebas de Calisoft para realizar pruebas de rendimiento a los sistemas informáticos.

2.5.3 Elección de las variables

Para ejecutar pruebas de rendimiento se deben definir las variables de entradas que se utilizarán en la ejecución de las pruebas y las variables de salida. Estas variables deben ser elegidas antes de la ejecución de las pruebas, una vez concluidas estas, se analiza el comportamiento de cada una para llegar a futuras conclusiones. La selección de las variables quedará de la siguiente manera:

Variables de Entrada:

- ✓ Número de hilos: Variable que establece la cantidad de hilos, que a su vez corresponde a los usuarios conectados simultáneamente al sistema.
- ✓ Período de subida: Intervalo de tiempo entre una conexión y otra, es decir, cuando se cumpla el valor determinado, se conecta un hilo hasta llegar al número de hilos establecido.
- ✓ Contador de bucle: Número de veces que se repetirá la prueba.

Variables de Salida:

- ✓ Rendimiento: Para medir el rendimiento del sistema los valores obtenidos se analizarán de la siguiente manera: Rendimiento medido por solicitudes/segundos y rendimiento medido en Kb/segundos.
- ✓ Tiempo promedio de respuesta: Tiempo medio que demora el sistema en dar una respuesta a las solicitudes realizadas.
- ✓ Por ciento (%) de errores: Porcentaje de las respuestas con errores.
- ✓ Capacidad de clientes: Capacidad máxima de clientes conectados a la aplicación.

El valor de la variable *Número de usuarios* se incrementará en el doble de su valor inicial, para ir probando hasta qué punto el sistema es resistente al estrés y la sobrecarga al cual será sometido. Al ejecutar las pruebas se obtendrá una serie de variables, estas son las que brinda el informe agregado de la herramienta. A continuación se brinda una descripción de las mismas:

Capítulo II: Diseño de la Propuesta

Muestras: Número hilos utilizados para cada URL (acrónimo en inglés de Uniform Resource Locator) de la prueba.

Media: Tiempo promedio que ha tardado el sistema para atender todas las solicitudes realizadas por la prueba.

Mediana: Valor determinado cumpliéndose que el 50% de los tiempos promedio para cada URL es menor o igual a dicho valor y el otro 50% es mayor.

Min: Tiempo mínimo de la muestra que se ha utilizado en una determinada URL.

Max: Tiempo máximo de la muestra que se ha utilizado en una determinada URL.

%Error: Por ciento de error de las muestras que tuvieron problemas con sus solicitudes.

Rendimiento: Medido en solicitudes atendidas por segundo/minuto/hora.

Kb/Seg: Rendimiento medido en Kb/segundos.

Estas variables serán de importancia para interpretar el resumen que brinda la herramienta.

2.5.4 Criterio de aceptación de los resultados

Los criterios de comparación que serán utilizados para validar la eficiencia del módulo Control de Personas son los obtenidos en la captura del RNF Eficiencia.

El rendimiento del sistema debe estar por encima de las 1000 solicitudes atendidas por segundo, además de soportar hasta 100 usuarios conectados simultáneamente.

El tiempo de respuesta en caso de ser una operación deberá estar por debajo de los 5 segundos. En caso de ser un reporte debe ser menor o igual a los 10 segundos, debido a que la cantidad de datos que maneja es elevada. Si se obtienen valores por encima de los definidos serán señal de que la funcionalidad que se está comprobando no es eficiente.

2.6 Diseño y configuración de las pruebas en la herramienta

Para la realización de las pruebas con la herramienta automatizada JMeter, se debe tener instalada la Máquina Virtual de Java 1.3 o superior a esta versión. Se debe instalar la herramienta JMeter.

Para obtener una muestra fiel de los elementos de interacción del sistema a probar se realiza la grabación de sus escenarios. Esta opción se realiza a través del elemento Servidor Proxy HTTP (Hypertext Transfer

Capítulo II: Diseño de la Propuesta

Protocol) de la herramienta de prueba. Este elemento Servidor Proxy HTTP permite al JMeter grabar todos los pasos realizados en el software de manera que se generarán los escenarios de prueba de manera automática.

La herramienta da la posibilidad de organizar en una serie de pasos todas las actividades, permitiendo obtener una mayor organización de las pruebas que se ejecutarán. Las pruebas a desarrollar para este sistema web son complejas, debido a que se trata de hacer coincidir el escenario de prueba con el escenario de navegación real. Los pasos a seguir para obtener una mayor organización de las pruebas son:

- ✓ Servidor Proxy HTTP: Permite realizar la grabación de los escenarios de pruebas.
- ✓ Grupo de Hilos: Cantidad de usuarios a simular.
- ✓ Árbol de Resultados: Permite ver todas las respuestas de las muestras, dando la posibilidad al usuario de escoger la forma de ver los resultados, como texto, HTML o XML.
- ✓ Aserción de Respuesta: Brinda un código de respuesta cuando las peticiones fueron atendidas de forma satisfactoria igual a 200, en caso contrario genera otro código indicando la presencia de errores.
- ✓ Informe Agregado: Resumen de los resultados de las pruebas.
- ✓ Gestor de Cookies HTTP: Posee dos funciones, la primera almacenar y enviar las cookies igual que un navegador de internet y la segunda adicionar manualmente una cookie al gestor.

La herramienta se configura antes de comenzar las pruebas, se le añade el elemento Servidor Proxy HTTP para la grabación del escenario. (Ver figura 1)

Capítulo II: Diseño de la Propuesta

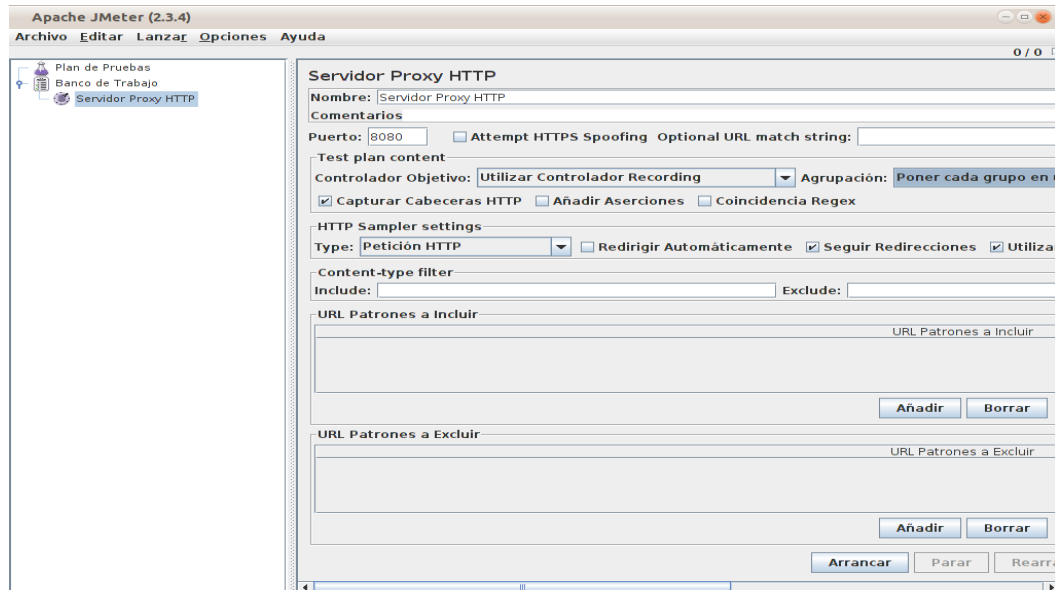


Figura 1: Configuración del Servidor Proxy HTTP.

Luego se adiciona el elemento Grupo de Hilos donde se especificará el número de usuarios para cada prueba.

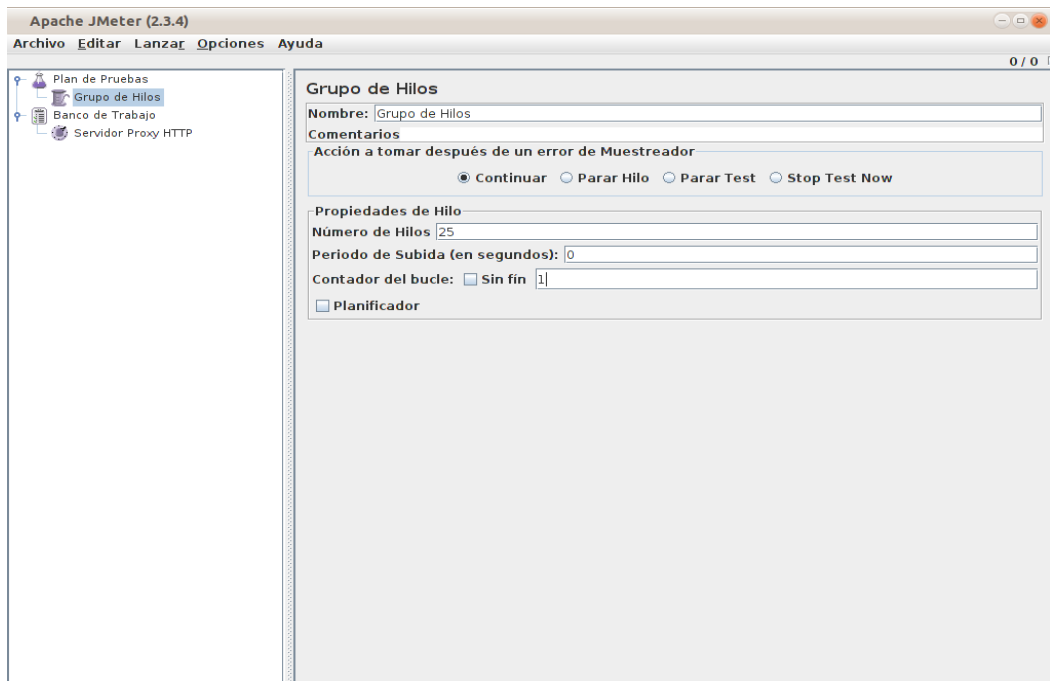


Figura 2: Configuración del número de hilos (Número de Usuarios).

Capítulo II: Diseño de la Propuesta

Se graban los escenarios una vez configurado el navegador presionando la opción arrancar.

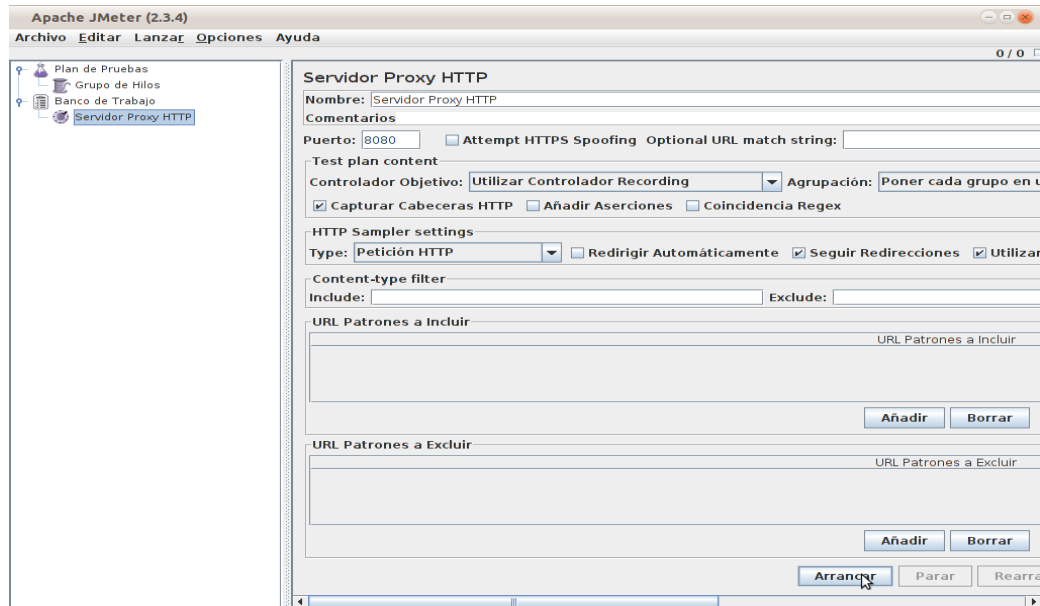


Figura 3: Configuración del número de hilos (Número de Usuarios).

Una vez grabados los escenarios se añaden los elementos Aserción de Respuesta, Mostrar Árbol de Resultados, Gestor de Cookies HTTP y el Informe Agregado. La herramienta está configurada para comenzar las pruebas para cada escenario grabado.

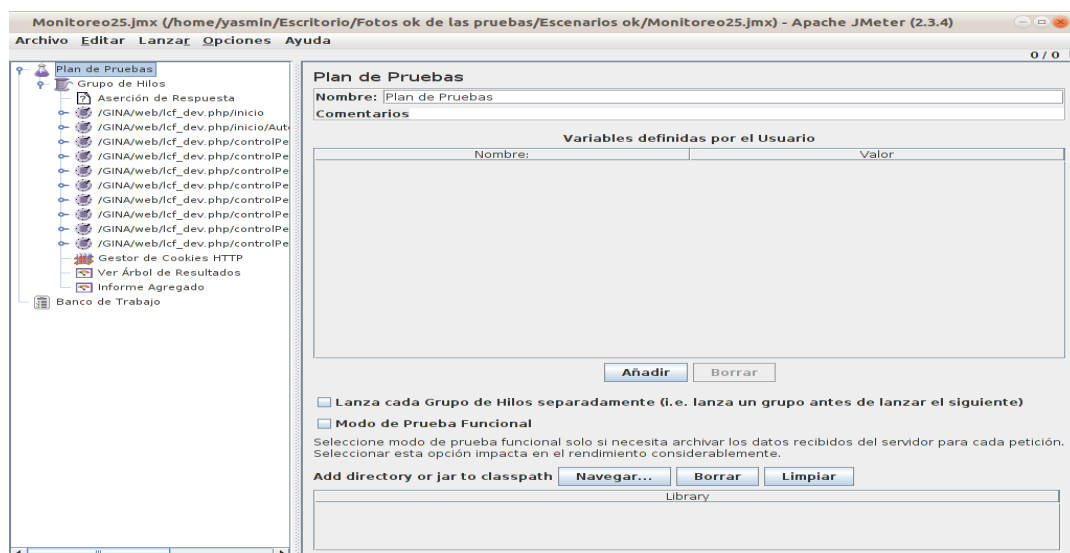


Figura 4: Configuración Final.

2.7 Funcionalidades de la solución Control de Personas

La solución Control de Personas está conformada por 22 funcionalidades, a continuación se brinda una lista con el nombre de cada una de ellas. Estas se encuentran en el Documento de Especificación de Requisitos de Software. Las cuatro funcionalidades de tipo Gestionar tienen la misma complejidad, solo que son realizadas por roles diferentes, por lo que se le realizarán pruebas a una de ellas solamente.

Caso de Uso 1: Gestionar PIA por Analista LCF Unidad

Caso de Uso 2: Gestionar PIA por Jefe LCF Unidad.

Caso de Uso 3: Gestionar PIA por analista LCF AGR.

Caso de Uso 4: Gestionar PIA por Jefe LCF AGR.

Caso de Uso 5: Realizar Estudio API.

Caso de Uso 6: Insertar Requerimiento Puntual.

Caso de Uso 7: Actualizar Categoría

Caso de Uso 8: Captar datos de viajero.

Caso de Uso 9: Definir Repetidor.

Caso de Uso 10: Buscar persona API.

Caso de Uso 11: Definir Propuesta.

Caso de Uso 12: Monitorizar viajeros.

Caso de Uso 13: Registrar incidencia.

Caso de Uso 14: Ver historial.

Caso de Uso 15: Concretar propuesta.

Caso de Uso 16: Mostrar Estudio API.

Caso de Uso 17: Mostrar PIA.

Caso de Uso 18: Mostrar Requerimiento Puntual.

Caso de Uso 19: Mostrar Posibles Redes de Vínculos.

Caso de Uso 20: Mostrar infracciones por fecha.

Caso de Uso 21: Mostrar Infracciones por persona.

Caso de Uso 22: Mostrar Propuestas.

Estas funcionalidades mencionadas anteriormente serán sometidas a pruebas utilizando la herramienta propuesta. De estas funcionalidades los casos de uso número 7 y 8 no serán probados debido a que estos son ejecutados de forma automática por el sistema. La funcionalidad *actualizar categoría* se ejecuta automáticamente, guiado por la fecha de inicio y fin de la gestión realizada. La *captura de los datos de los viajeros* se inicia cada tres segundos, obteniendo los datos que proporciona inmigración de los viajeros que cruzan frontera.

2.8 Diseño de los Casos de Pruebas

El diseño de los casos de prueba es de gran importancia para realizar las pruebas, es la guía que se seguirá para llevar a cabo el proceso de pruebas. Para el desarrollo de las pruebas de carga y de estrés de la solución Control de Personas 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. (Ver Anexo 3)

Para las pruebas de carga la variable que especifica el tiempo entre conexiones siempre va a tener valor 0 porque los usuarios se conectarán todos en el mismo instante de tiempo. La variable *Contador de bucle* tendrá valor 1 debido a que en las pruebas de carga los usuarios se conectarán de manera simultánea y una sola vez. A continuación se muestra un ejemplo de cómo quedarán los casos de prueba para las pruebas de rendimiento para una funcionalidad. Las restantes funcionalidades serán probadas con un conjunto de valores similares.

Pruebas de Carga

Caso de prueba 1:

Nombre del Proyecto: Gestión Integral de Aduanas.

Nombre Módulo: Control de Personas.

Versión: 1.0

Tipo de prueba: Prueba de Carga

Funcionalidad

Funcionalidad	Descripción
Gestionar PIA Por Jefe LCF Unidad	El Jefe de LCF Unidad inserta las propuestas efectuadas como PIA por los analistas de su unidad. Realiza la propuesta para PIA-Control modificando los campos convenientes para registrar la propuesta.

Variables para la 1ra Iteración

Variables	
Número de usuarios concurrentes: [Número de hilos]	25
Tiempo entre conexión y conexión: [Periodo de subida (en segundos)]	0
Número de iteraciones: [Contador del bucle]	1

Variables para la 2da Iteración

Variables	
Número de usuarios concurrentes: [Número de hilos]	50
Tiempo entre conexión y conexión: [Periodo de subida (en segundos)]	0

Número de iteraciones: [Contador del bucle]	1
---	---

Variables para la 3ra Iteración

Variables	Iteración 3
Número de usuarios concurrentes: [Número de hilos]	100
Tiempo entre conexión y conexión: [Periodo de subida (en segundos)]	0
Número de iteraciones: [Contador del bucle]	1

Para la realización de las pruebas de estrés se especifica en las variables el tiempo que va a existir entre las conexiones, indicando la demora entre la conexión de un hilo y otro. La cantidad de veces que se va a ejecutar la simulación de los usuarios se debe especificar en la variable *Número de iteraciones*, para que al ejecutarse las pruebas se alcance el número de hilos que se definió.

Pruebas de Estrés

Caso de prueba 1:

Nombre del Proyecto: Gestión Integral de Aduanas.

Nombre Módulo: Control de Personas.

Versión: 1.0

Tipo de prueba: Prueba de Estrés

Capítulo II: Diseño de la Propuesta

Funcionalidad

Funcionalidad	Descripción
CU6_Gestionar PIA Por Jefe LCF Unidad	El Jefe de LCF Unidad inserta las propuestas efectuadas como PIA por los analistas de su unidad. Realiza la propuesta para PIA-Control modificando los campos convenientes para registrar la propuesta.

Variables de la 1ra Iteración

Variables	Iteración 1
Número de usuarios concurrentes: [Número de hilos]	25
Tiempo entre conexión y conexión: [Periodo de subida (en segundos)]	2
Número de iteraciones: [Contador del bucle]	4

Variables de la 2da Iteración

Variables	Iteración 2
Número de usuarios concurrentes: [Número de hilos]	50
Tiempo entre conexión y conexión:	2

[Periodo de subida (en segundos)]	
Número de iteraciones: [Contador del bucle]	3

Variables de la 3ra Iteración

Variables	
Número de usuarios concurrentes: [Número de hilos]	50
Tiempo entre conexión y conexión: [Periodo de subida (en segundos)]	4
Número de iteraciones: [Contador del bucle]	4

Con los diseños de los casos de prueba, la herramienta JMeter disponible con todas las configuraciones necesarias y la máquina virtual de java superior a la versión 1.3 instalada, comienza el proceso de pruebas.

2.9 Conclusiones parciales

En este capítulo se realizó una breve descripción del módulo Control de Personas del sistema GINA, al cual se le especificaron los RNF de software con la ayuda de las técnicas de captura de requisitos seleccionadas, fueron documentados en el documento de especificación de requisitos de software obteniéndose un acta de aceptación. Estos requisitos fueron validados mediante revisiones formales con la analista principal de la línea donde se desarrolló el módulo, el programador y el cliente. El RNF Eficiencia se validará mediante las pruebas de rendimiento de carga y estrés que fueron las seleccionadas

Capítulo II: Diseño de la Propuesta

para medir dicho requisito. Para ello quedó definido un entorno de pruebas que brinda las características de software y hardware de las computadoras donde se llevaron a cabo las pruebas. Las pruebas fueron realizadas con la herramienta para pruebas automatizadas JMeter, para esta se realizó su configuración y se definieron un conjunto de variables de entrada y salida que permitirán el análisis de los resultados. Quedaron confeccionados los casos de pruebas necesarios para cada una de las funcionalidades del módulo.

CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS

3.1 Introducción

En este capítulo se analizan los resultados de las pruebas de rendimiento de carga y de estrés realizadas a la solución Control de Personas mediante gráficos y tablas. Se analiza la eficiencia del sistema para cada una de las funcionalidades probadas, y se evalúa el resultado general de las pruebas.

3.2 Resultados de las pruebas de Cargas

El módulo Control de Personas fue sometido a pruebas de rendimiento de carga. Todas las funcionalidades del sistema fueron probadas. Las funcionalidades **Actualizar Categoría** y **Captura de los Datos de los Viajeros** se ejecutan de manera automática, estas dos funcionalidades no son objeto de prueba, el sistema las ejecuta sin interacción del usuario. Las variables de salidas que se tendrán en cuenta para el análisis serán **número de muestras**, el **porcentaje de error** que presente la funcionalidad, el **tiempo promedio** para un conjunto de resultados en segundos y el **rendimiento** de la funcionalidad, medido en cantidad de solicitudes atendidas por segundos y en Kb/seg, refiriéndose a la velocidad que alcanzaba cuando atendía esas solicitudes. A continuación se realizará el análisis del resultado de las pruebas para cada funcionalidad probada.

3.2.1 Funcionalidad Buscar Persona API

En la prueba de carga para la funcionalidad Buscar Persona API los resultados obtenidos son los siguientes:

Para 25 usuarios conectados simultáneamente se generaron 175 muestras. Se obtuvo un 14.29% de error. El rendimiento del sistema fue de 7.3/seg en cuanto a las solicitudes atendidas y de 79.3 Kb/seg refiriéndose a la velocidad. El tiempo promedio para el conjunto de resultados: 2.765 seg.

Para 50 usuarios conectados simultáneamente se generaron 350 muestras. Se obtuvo un 24.57% de error. El rendimiento del sistema para esta funcionalidad: 2.5/seg y 23.8 Kb/seg. El tiempo promedio de respuesta: 15.786 seg.

El sistema no pudo responder para 100 usuarios, el servidor no fue capaz de atender la sobrecarga de peticiones realizadas.

El error se encuentra en la dirección /GINA/web/images/loading.gif.

Capítulo III: Análisis de los Resultados

El 14.29% y 24.75% de error obtenidos para 175 y 350 muestras, representan 25 y 85 peticiones respectivamente de los usuarios que no pudieron ser atendidas.

En resumen en esta funcionalidad el tiempo de respuesta para 50 usuarios simultáneos fue superior al valor esperado (5 seg), constituyendo esto un caso crítico. El rendimiento también estuvo por debajo de lo esperado (1000 solicitudes/seg). Por lo antes planteado esta funcionalidad no es eficiente en el entorno de prueba creado.

3.2.2 Funcionalidad Concretar Propuesta

Para la funcionalidad Concretar Propuesta los resultados obtenidos fueron:

Para un total de 25 usuarios se generaron 425 muestras. El tiempo promedio para un conjunto de resultados fue de 0.006 seg. Se obtuvo un 5.88% de error. El rendimiento: 1864/seg y 169.7 Kb/seg.

Al simular 50 usuarios se generan 850 muestras. El rendimiento alcanzó valores de 1828/seg y de 166.4 Kb/seg. El tiempo promedio: 0.010 seg. El porcentaje de error fue de 5.88%.

Al simular 100 usuarios se generan 1700 muestras. El rendimiento alcanzó valores de 1800.8/seg y de 164 Kb/seg. El tiempo promedio: 0.020 seg. El porcentaje de error fue de 5.88%.

En esta funcionalidad los tiempos de respuesta fueron óptimos, siendo todos menores a 1 seg. El rendimiento fue bueno, superando las 1000 solicitudes/seg atendidas.

3.2.3 Funcionalidad Definir Propuesta

Para la funcionalidad Definir Propuesta los resultados obtenidos fueron:

Para un total de 25 usuarios se generaron 225 muestras. El tiempo promedio fue de 0.006 seg. Se obtuvo un 11.11% de error. El rendimiento fue de 1480.3/seg y de 151.8 Kb/seg.

Al simular 50 usuarios se generaron 450 muestras. El rendimiento alcanzó valores de 1601.4/seg y de 164.2 Kb/seg. El tiempo promedio fue de 0.014 seg. El porcentaje de error fue de 11.11%.

Para 100 usuarios conectados se generaron 900 muestras. El tiempo promedio para el conjunto de resultados fue de 0.006 seg. Se obtuvo un 11.11% de error. El rendimiento fue de 1829.3/seg y de 187.6 Kb/seg.

Los tiempos de respuesta fueron óptimos, todos por debajo de 1 seg. El rendimiento a su vez fue bueno, atendándose las solicitudes a más de 1000 por segundo.

3.2.4 Funcionalidad Definir Repetidor

Para la funcionalidad Definir Repetidor los resultados obtenidos fueron:

Para un total de 25 usuarios se generaron 1400 muestras. El tiempo promedio para el conjunto de resultados fue de 0.020 seg. Se obtuvo un 3,57% de error. El rendimiento fue de 898.6/seg y de carga de 14010.6 Kb/seg.

Al simular 50 usuarios se generan 2800 muestras. El rendimiento alcanzó valores de 1042.4/seg y de 16253.6 Kb/seg. El tiempo promedio para el conjunto de resultados fue de 0.027 seg. El porcentaje de error fue de 3,57%.

Para 100 usuarios conectados se generaron 5600 muestras. El tiempo promedio fue de 0.057 seg. Se obtuvo un 3.57% de error. El rendimiento fue 1000.2/seg y de 15594.6 Kb/segundos.

El rendimiento para 25 usuarios quedó relativamente por debajo de las 1000 solicitudes atendidas por segundo, no siendo así para 50 y 100 usuarios concurrentes, por lo que el rendimiento no deja de ser bueno. Los tiempos de respuesta promedio estuvieron por debajo de 1 segundo, considerándose óptimos.

3.2.5 Funcionalidad Realizar Estudio API

Para la funcionalidad Realizar Estudio API los resultados obtenidos fueron:

Para un total de 25 usuarios se generaron 450 muestras. El tiempo promedio fue de 0.004 seg. Se obtuvo un 5.56% de error. El rendimiento fue de 2163.5/seg y de 195.4 Kb/seg.

Al simular 50 usuarios se generan 900 muestras. El rendimiento alcanzó valores de 2013.4/seg y de 181.9 Kb/seg. El tiempo promedio tuvo el valor de 0.010 seg. El porcentaje de error fue de 5,56%.

Para 100 usuarios conectados se generaron 1800 muestras. El tiempo promedio para el conjunto de resultados fue de 0.019 seg. Se obtuvo un 5,56% de error. El rendimiento de las solicitudes fue 1925.1/seg y de 173.9 Kb/seg.

Los tiempos de respuesta son óptimos, siendo menores de 1 seg. Las respuestas a las solicitudes se realizaron aproximadamente a razón de 2000 por segundo, considerándose así valores excelentes.

3.2.6 Funcionalidad Gestionar PIA por Jefe LCF AGR

Para la funcionalidad Gestionar PIA por Jefe LCF AGR los resultados obtenidos fueron:

Capítulo III: Análisis de los Resultados

Para un total de 25 usuarios se generaron 1125 muestras. El tiempo promedio fue de 1.126 seg. Se obtuvo un 6,67% de error. El rendimiento fue de 21.3/seg y de 411.4Kb/seg.

Al simular 50 usuarios se generan 2250 muestras. El rendimiento alcanzó valores de 14.9/seg y de 288.3 Kb/seg. El tiempo de promedio fue de 3.195 seg. El porcentaje de error fue de 6.67%.

El sistema no respondió con 100 usuarios concurrentes. El rendimiento no fue el esperado, aunque los tiempos de respuesta para 25 y 50 usuarios fueron buenos. El servidor no pudo atender las peticiones realizadas para 100 usuarios concurrentes.

3.2.7 Funcionalidad Monitorizar Viajeros

Para la funcionalidad Monitorizar Viajeros los resultados obtenidos fueron:

Para un total de 25 usuarios se generaron 650 muestras. El tiempo promedio alcanzó los 0.002 seg. Se obtuvo un 3.85% de error. El rendimiento fue de 2708.3/seg y de 1024.2 Kb/seg para las solicitudes y velocidad respectivamente.

Al simular 50 usuarios se generan 1300 muestras. El rendimiento alcanzó un valor de 2313.2/seg para las solicitudes atendidas y de 874.7 Kb/seg para la velocidad. El tiempo promedio fue de 0.010. El porcentaje de error fue de 3.85%.

Para 100 usuarios conectados se generaron 2600 muestras. El tiempo promedio fue de 0.015 seg. Se obtuvo un 3.85% de error. El rendimiento fue de 2311.1/seg y de 874 Kb/seg para la cantidad de solicitudes y la velocidad respectivamente.

El rendimiento se mantuvo por encima de las 2000 solicitudes atendidas por segundo, siendo este valor muy bueno. Los tiempos de respuesta se mantuvieron por debajo de 0.5 segundos, por lo que el sistema responde muy rápido para esta funcionalidad.

3.2.8 Funcionalidad Mostrar Estudio API

Para la funcionalidad Mostrar Estudio API los resultados obtenidos fueron:

Para un total de 25 usuarios se generaron 350 muestras. El tiempo promedio fue de 0.004 seg. Se obtuvo un 7.14% de error. El rendimiento fue de 2147.2/seg y de 201.5 Kb/seg.

Al simular 50 usuarios se generan 700 muestras. El rendimiento alcanzó valores de 1949.9/seg y de 182.9 Kb/seg. El tiempo promedio alcanzó el valor de 0.009 seg. El porcentaje de error fue de 7.14%.

Capítulo III: Análisis de los Resultados

Para 100 usuarios conectados se generaron 1400 muestras. El tiempo promedio fue de 0.011 seg. Se obtuvo un 7.14% de error. El rendimiento fue de 1983/seg para las solicitudes y la velocidad fue de 186 Kb/seg.

El rendimiento fue muy bueno, manteniéndose cerca de las 2000 solicitudes atendidas por segundos. El sistema respondió muy rápido brindando un tiempo promedio por debajo 0.5 segundos.

3.2.9 Funcionalidad Mostrar Infracciones por Fecha

Para la funcionalidad Mostrar Infracciones por Fecha los resultados obtenidos fueron:

Para un total de 25 usuarios se generaron 425 muestras. El tiempo promedio fue de 0.004 seg. Se obtuvo un 5.88% de error. El rendimiento fue de 1995.3/seg y de 181.7 Kb/seg.

Al simular 50 usuarios se generan 850 muestras. El rendimiento alcanzó un valor de 1847.8/seg y de 168.2 Kb/seg. El tiempo promedio alcanzó el valor de 0.012 seg para un conjunto de valores. El porcentaje de error fue de 5.88%.

Para 100 usuarios conectados se generaron 1700 muestras. El tiempo promedio fue de 0.014 seg. Se obtuvo un 5.88% de error. El rendimiento fue de 2080.8/seg y de 189.5 Kb/seg.

El sistema respondió en menos de 1 segundo a las solicitudes. El rendimiento fue muy bueno, estando cerca de las 2000 solicitudes atendidas por segundo.

3.2.10 Funcionalidad Mostrar Infracciones por Persona

Para la funcionalidad Mostrar Infracciones por Persona los resultados obtenidos fueron:

Para un total de 25 usuarios se generaron 350 muestras. El tiempo promedio fue de 0.006 seg. Se obtuvo un 7.14% de error. El rendimiento fue de 1842.1/seg y de 172.8 Kb/seg.

Al simular 50 usuarios se generan 700 muestras. El rendimiento alcanzó un valor de 1949.9/seg y de 182.9 Kb/seg. El tiempo promedio fue de 0.009 seg. El porcentaje de error fue de 7.14%.

Para 100 usuarios conectados se generaron 1400 muestras. El tiempo promedio fue de 0.010 seg. Se obtuvo un 7.14% de error. El rendimiento fue de 2020.2/seg y de 189.5 Kb/seg.

El tiempo de respuesta y el rendimiento son óptimos, superando ampliamente los valores esperados.

3.2.11 Funcionalidad Mostrar PIA

Para la funcionalidad Mostrar PIA los resultados obtenidos fueron:

Para un total de 25 usuarios se generaron 250 muestras. El tiempo promedio alcanzó el valor de 0.004 seg. Se obtuvo un 10% de error. El rendimiento fue de 2049.2/seg y de 205.1 Kb/seg.

Al simular 50 usuarios se generan 500 muestras. El rendimiento alcanzó los valores de 2222.2/seg y de 222.4 Kb/seg. El tiempo promedio para un conjunto de resultados fue de 0.005 seg. El porcentaje de error fue de 10%.

Para 100 usuarios conectados se generaron 1000 muestras. El tiempo promedio fue de 0.008 seg. Se obtuvo un 10% de error. El rendimiento fue de 2040.8/seg y de 204.3 Kb/seg.

El rendimiento estuvo por encima de las 2000 solicitudes/segundo independientemente del número de usuarios conectados y los tiempos de respuesta fueron muy buenos, por debajo de 0.5 segundos.

3.2.12 Funcionalidad Mostrar Posibles Redes de Vínculos

Para la funcionalidad Mostrar Posibles Redes de Vínculos los resultados obtenidos fueron:

Para un total de 25 usuarios se generaron 275 muestras. El tiempo promedio alcanzado fue de 2.181 seg. Se obtuvo un 9.09% de error. El rendimiento fue de 10.4/seg y de 14 Kb/seg.

Al simular 50 usuarios se generaron 550 muestras. El rendimiento alcanzó un valor de 6.4/seg y de 8.6 Kb/seg para las solicitudes atendidas y la velocidad respectivamente. El tiempo promedio fue de 6.28 seg. El porcentaje de error fue de 9.09%.

Con una carga de 100 usuarios, el sistema no respondió. El servidor no fue capaz de dar respuesta a las solicitudes simultáneas.

Los tiempos promedio de respuesta fueron buenos, por debajo de los 10 seg para 25 y 50 usuarios. El rendimiento mostró valores críticos por debajo de las 11 solicitudes atendidas por segundo.

3.2.13 Funcionalidad Mostrar Propuestas

Para la funcionalidad Mostrar Propuestas los resultados obtenidos fueron:

Para un total de 25 usuarios se generaron 250 muestras. El tiempo promedio tuvo el valor 0.004 seg. Se obtuvo un 10% de error. El rendimiento fue de 2066.1/seg y de 206.8 Kb/seg para cantidad de solicitudes atendidas y velocidad respectivamente.

Capítulo III: Análisis de los Resultados

Al simular 50 usuarios se generan 500 muestras. El rendimiento alcanzó valores de 2100.8/seg para las solicitudes y de 210.3 Kb/seg. El tiempo promedio fue de 0.004 seg. El porcentaje de error se mantuvo en 10%.

Para 100 usuarios conectados se generaron 1000 muestras. El tiempo promedio fue de 0.005 seg. Se obtuvo un 10% de error. El rendimiento fue de 1742.2/seg y de 174.4 Kb/seg.

El rendimiento estuvo por cerca de las 2000 solicitudes/segundo independientemente del número de usuarios conectados y los tiempos de respuesta fueron muy buenos, por debajo de 0.1 segundos.

3.2.14 Funcionalidad Mostrar Requerimiento Puntual.

Para la funcionalidad Mostrar Requerimiento Puntual los resultados obtenidos fueron:

Para un total de 25 usuarios se generaron 1725 muestras. El tiempo promedio fue de 0.288 seg. Se obtuvo un 1.45% de error. El rendimiento fue de 75.2 /seg para las solicitudes atendidas y de 692.1 Kb/seg para la velocidad.

Al simular 50 usuarios se generan 3450 muestras. El rendimiento alcanzó un valor de 50.4/seg y de 463.9 Kb/seg. El tiempo promedio fue de 0.758 seg. El porcentaje de error fue de 1.45%.

No se pudo realizar la iteración de la prueba con 100 usuarios, debido a que el sistema no respondió con las características del entorno de prueba, el servidor no fue capaz de atender las solicitudes realizadas.

3.2.15 Funcionalidad Registrar Incidencia

Para la funcionalidad Registrar Incidencia:

Para un total de 25 usuarios se generaron 1700 muestras. El tiempo promedio fue de 1.028 seg. Se obtuvo un 3.18% de error. El rendimiento fue de 23.5/seg y de 311.3 Kb/seg para las solicitudes y la velocidad respectivamente.

Al simular 50 usuarios se generaron 3400 muestras. El rendimiento alcanzó valores de 13.8/seg y de 182.9 Kb/seg. El tiempo promedio fue de 3.266 seg. El porcentaje de error fue de 6.82%.

Esta funcionalidad maneja gran cantidad de datos por lo que el sistema no pudo responder al conectarse 100 usuarios y hacer peticiones por cada uno de ellos.

3.2.16 Funcionalidad Insertar Requerimiento Puntual

Para la funcionalidad Requerimiento Puntual se obtuvo el siguiente resultado:

Capítulo III: Análisis de los Resultados

Para un total de 25 usuarios se generaron 475 muestras. El tiempo promedio para un conjunto de resultados fue de 2.033 seg. Se obtuvo un 5.26% de error. El rendimiento fue de 11.6/seg y de 14.0 Kb/seg.

Al simular 50 usuarios se generaron 1425 muestras. El rendimiento alcanzó un valor de 3.7/seg y de 4.4 Kb/seg. El tiempo promedio alcanzó el valor de 3.392 seg. El porcentaje de error fue de 5.26%.

Con 100 usuarios concurrentes el sistema no respondió a las solicitudes. Con el entorno de pruebas actual no se pueden conectar esa cantidad de usuarios. El valor máximo de carga fue de 75 hilos.

3.2.17 Funcionalidad Ver Historial

Para la funcionalidad Ver Historial el sistema se comporta de la siguiente manera:

Para un total de 25 usuarios se generaron 325 muestras. El tiempo promedio fue de 0.005 seg. Se obtuvo un 7.69% de error. El rendimiento fue de 1747.3/seg y de 166 Kb/seg en cuanto a las solicitudes atendidas y la velocidad respectivamente.

Al simular 50 usuarios se generan 650 muestras. El rendimiento alcanzó un valor de 1895/seg y de 180.1 Kb/seg para las solicitudes y la velocidad respectivamente. El tiempo promedio fue de 0.007 seg. El porcentaje de error fue de 7,69%.

Para 100 usuarios conectados se generaron 1300 muestras. El tiempo promedio alcanzó el valor de 0.012 seg. Se obtuvo un 7,69% de error. El rendimiento fue de 2028.1/seg y de 192.7 Kb/seg.

El rendimiento medido en solicitudes/segundo estuvo por encima de 1000. El tiempo de respuesta promedio fue menor de 1 segundo.

3.2.18 Resultados generales de las pruebas de carga

A continuación se mostrarán los gráficos obtenidos que evidencian el *Rendimiento* y el *Tiempo de Respuesta* para 25, 50 y 100 usuarios. Se realizará un análisis para cada grupo de usuarios atendiendo a estas variables.

Capítulo III: Análisis de los Resultados

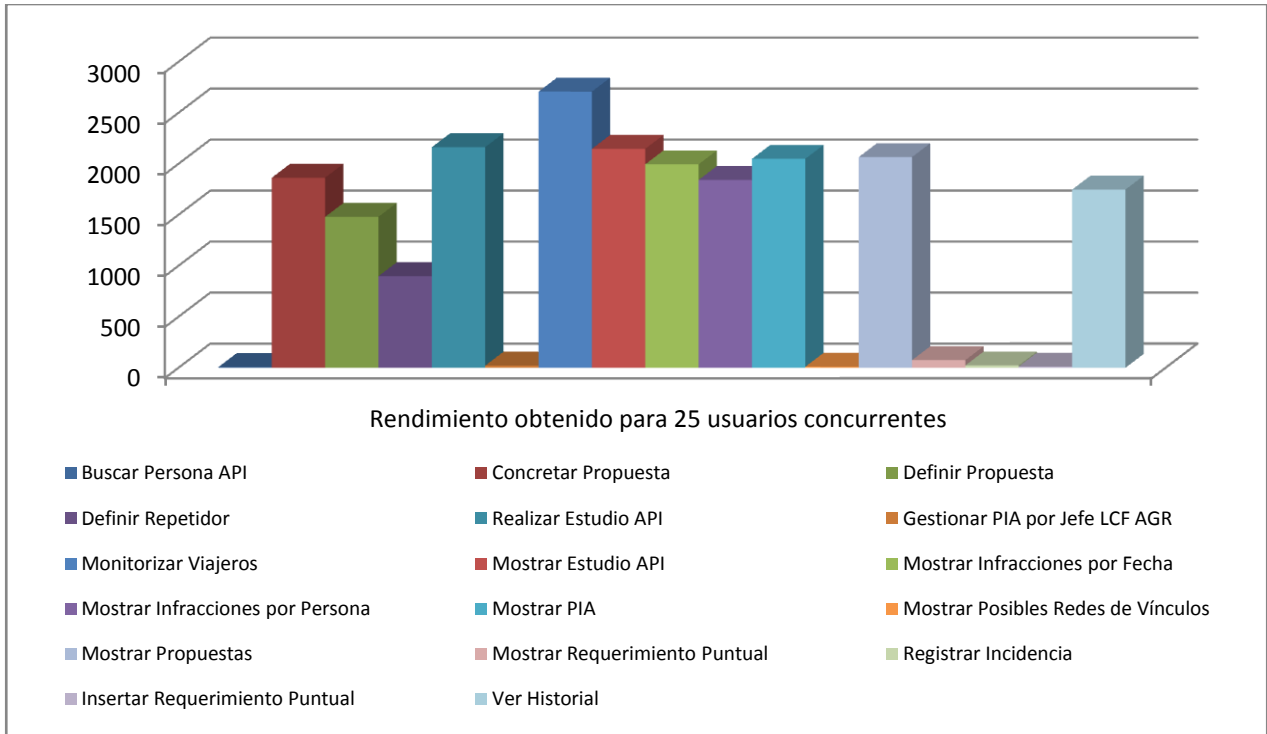


Figura 5: Gráfico que muestra el Rendimiento para 25 usuarios.

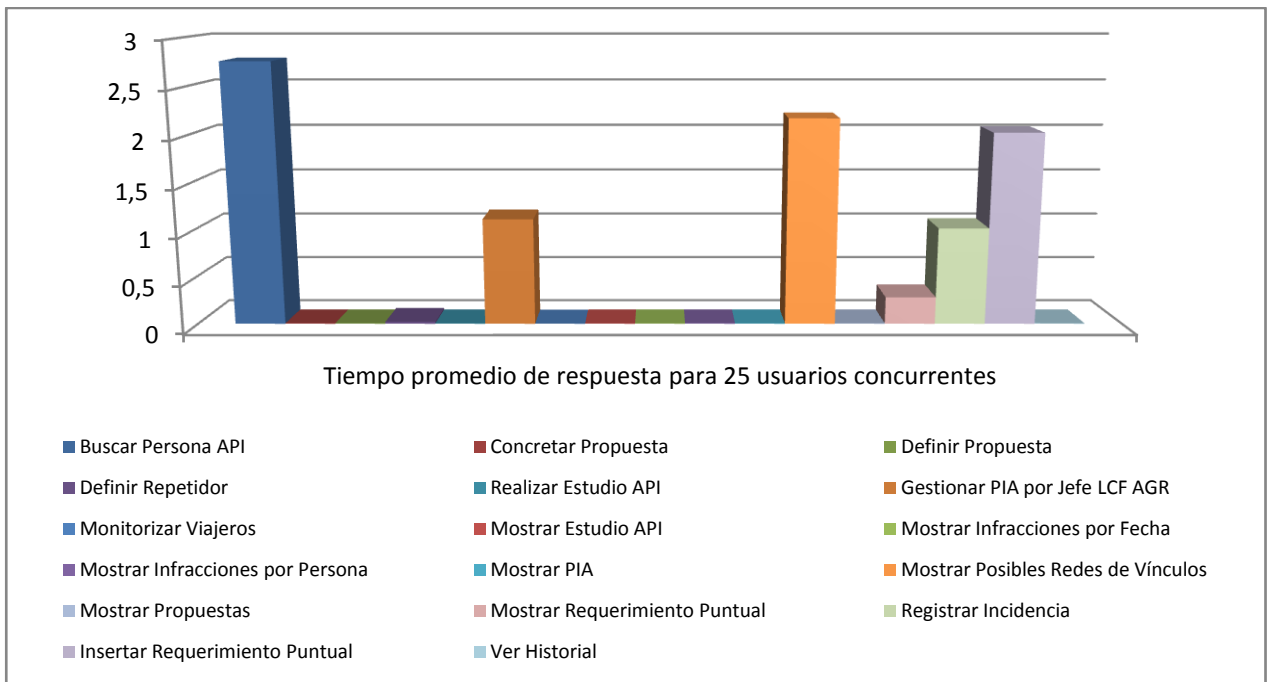


Figura 6: Gráfico que muestra el tiempo promedio de respuesta para 25 usuarios.

Capítulo III: Análisis de los Resultados

De las 17 funcionalidades probadas mediante pruebas de carga, 10 presentaron un rendimiento por encima de las 1000 solicitudes/segundo, que era el valor establecido en la especificación de requisitos de software obtenida. Mientras que 7 funcionalidades estuvieron por debajo de este valor, de ellas 6 tuvieron un rendimiento malo, por debajo de las 100 solicitudes/segundo. Sin embargo los tiempos promedios de respuesta fueron óptimos, siendo todos inferiores a los 3 seg, por lo que se compensa el mal rendimiento anteriormente expuesto, ya que el usuario estaría insatisfecho si tuviera que esperar demasiado tiempo por el sistema. De todas formas se hace necesario revisar las funcionalidades ineficientes, aunque vale destacar que éstas son las de mayor complejidad, ya que manejan un mayor volumen de datos y realizan una mayor cantidad de operaciones.

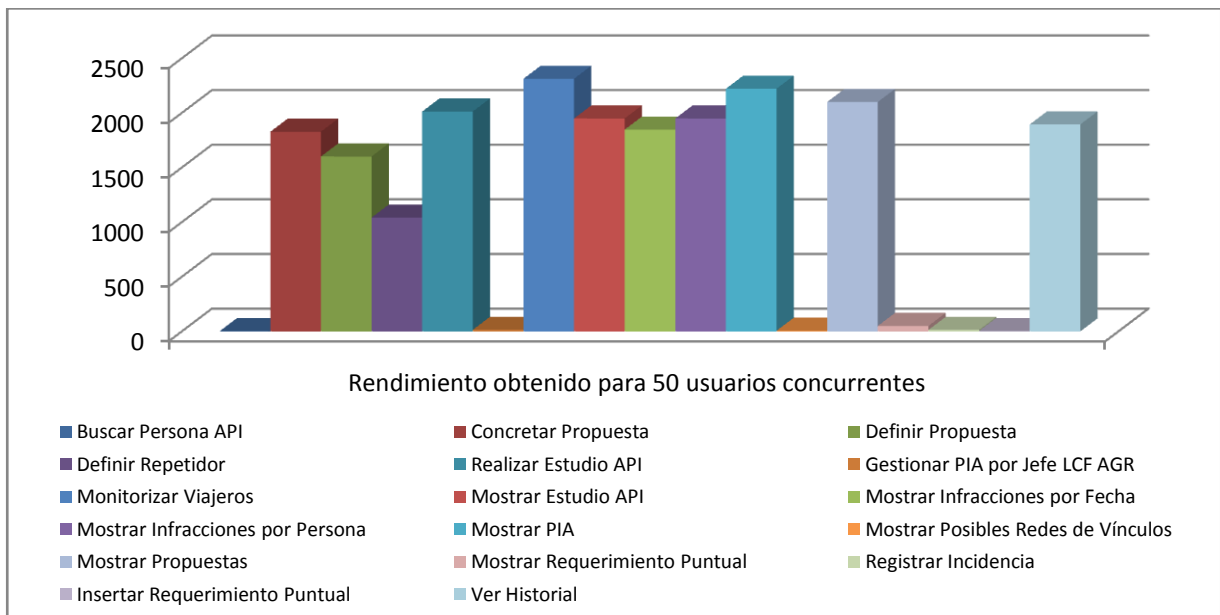


Figura 7: Gráfico que muestra el Rendimiento para 50 usuarios.

Capítulo III: Análisis de los Resultados

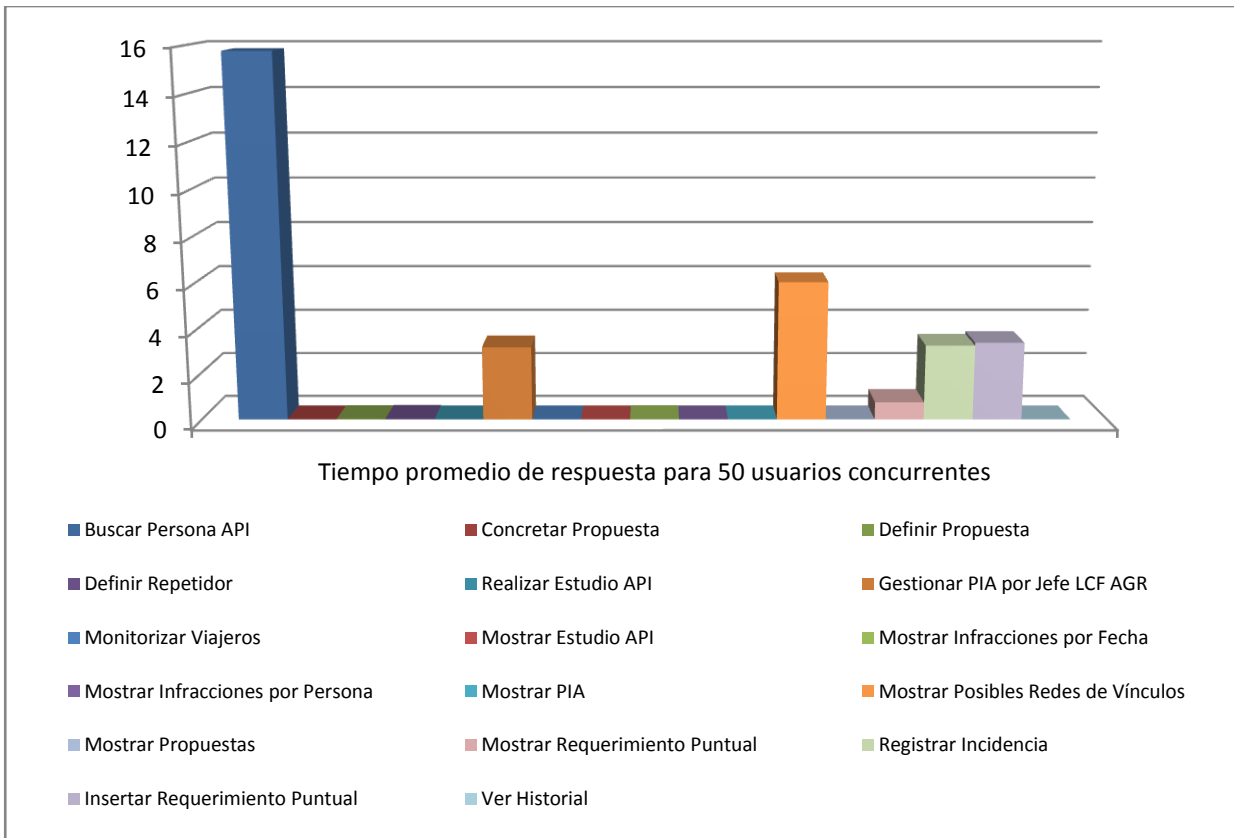


Figura 8: Gráfico que muestra el tiempo promedio de respuesta para 50 usuarios.

Presentaron un buen rendimiento 10 funcionalidades, donde se atendieron las solicitudes a más de 1000 por segundo, no siendo así para las restantes 6 que tuvieron un rendimiento bajo, por debajo de 100 solicitudes atendidas por segundo y una por debajo de las 1000 sol/seg. Coinciden estas últimas con las ineficientes para 25 usuarios, por las mismas razones. El tiempo de respuesta fue elevado para la funcionalidad Buscar Persona API, con un valor de 24.11 seg, muy por encima de los 5 seg esperados; las demás funcionalidades respondieron satisfactoriamente en tiempos menores a 5 seg. Por lo antes planteado, en general el sistema respondió muy rápido a las solicitudes, aunque hay que revisar el caso puntual expuesto.

Capítulo III: Análisis de los Resultados

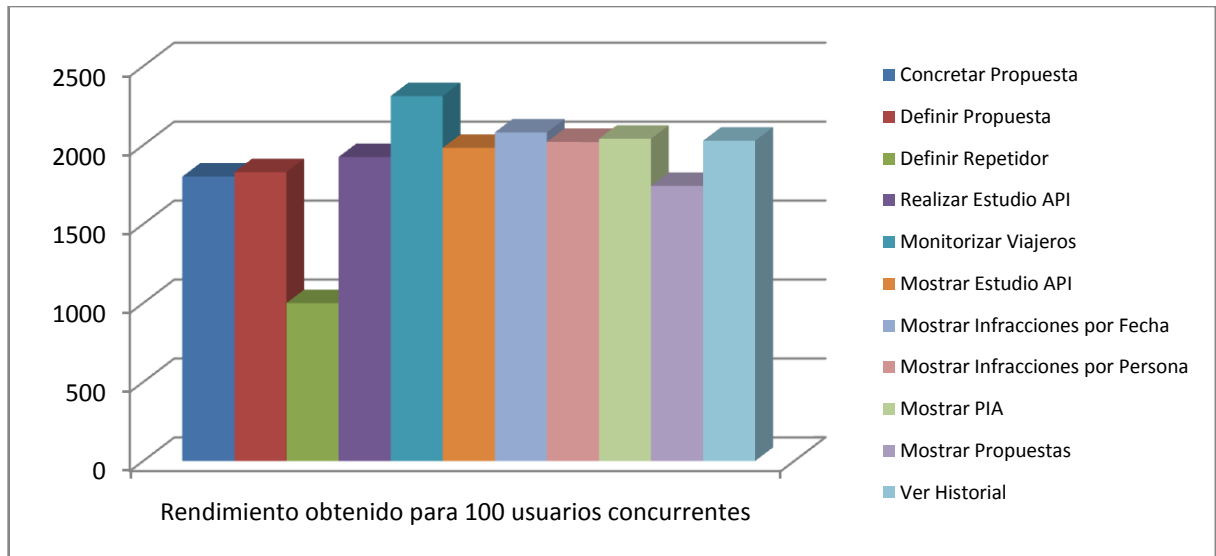


Figura 9: Gráfico que muestra el Rendimiento para 100 usuarios.

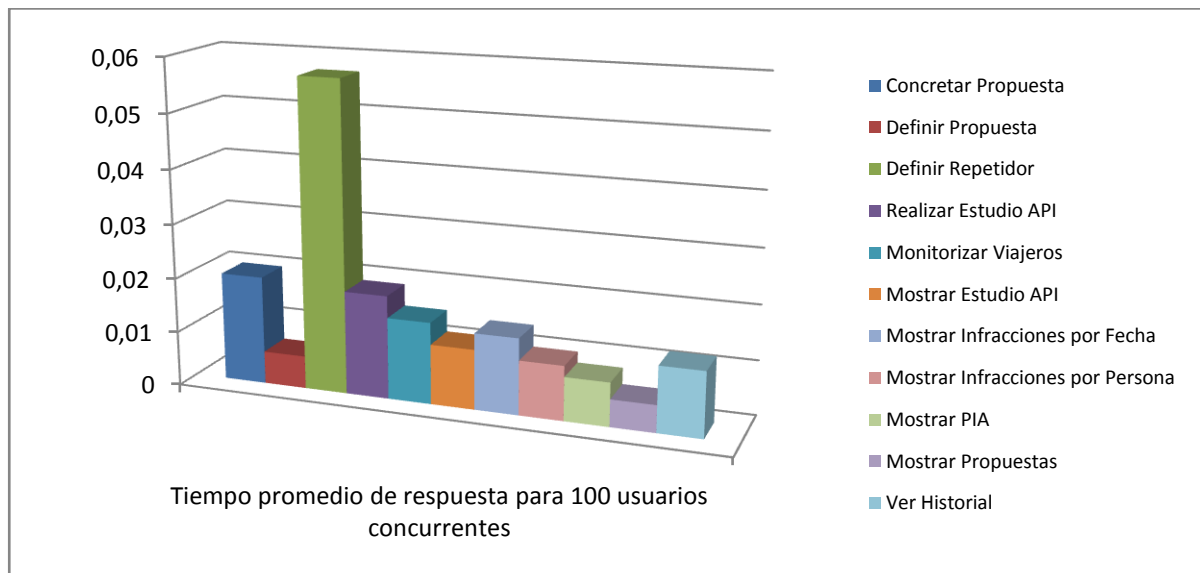


Figura 10: Gráfico que muestra el tiempo promedio de respuesta para 100 usuarios.

Para esta cantidad de hilos, las pruebas mostraron rendimientos buenos para 11 funcionalidades, solo una de estas ligeramente no llegó a las 1000 solicitudes/segundo. Las restantes 6 funcionalidades no son operacionales para 100 hilos, como era de esperar coinciden estas con las ineficientes para 25 y 50 hilos. Los tiempos de respuesta para esas 11 funcionalidades fueron muy buenos, inferiores al segundo, pero

Capítulo III: Análisis de los Resultados

para las que el sistema no respondió es obvio que los tiempos promedios iban a ser inaceptables. Hay que destacar que el entorno de pruebas, realmente no ha sido el mejor, ya que no se acerca al que tendrá la aplicación cuando esté en pleno funcionamiento.

En todas las pruebas se presentó un error constante, referido a una imagen que nunca pudo cargarse, aunque no es determinante este hecho, pues no incide en el resultado que obtiene el usuario. Sin embargo las funcionalidades Buscar Persona API, para 50 hilos, y Registrar Incidencia, para 25 y 50 hilos, presentaron otros errores, con porcentajes de 24.57%, 3.18% y 6.82% respectivamente, lo que representa que algunos usuarios no pudieron obtener el resultado esperado, fueron peticiones a las que el servidor no pudo dar respuesta. Se hace necesario revisar estas posibles inconformidades.

3.3 Resultados de las pruebas de Estrés

Se realizaron pruebas de rendimiento de estrés a la solución Control de Personas. Las funcionalidades del sistema fueron probadas obteniendo resultados que serán expuestos a continuación. Al igual que para las pruebas de carga las funcionalidades actualizar categoría y la captura de los datos de los viajeros se ejecutan de manera automática y no son objeto de prueba. Las variables de salida que se tendrán en cuenta para el análisis serán las mismas variables utilizadas en las pruebas de carga. Estas pruebas se hicieron con 100 hilos, lo cual simula 100 usuarios interactuando con el sistema.

3.3.1 Funcionalidad Buscar Persona API

# Muestras	Tiempo Promedio	% Error	Rendimiento	Kb/seg
700	2.927	14.29	8.1/seg	88.2

El rendimiento fue extremadamente bajo, pero el sistema respondió rápidamente.

3.3.2 Funcionalidad Concretar Propuesta

# Muestras	Tiempo Promedio	% Error	Rendimiento	Kb/seg
1700	0.002	5.88	803/seg	73.1

El rendimiento estuvo por debajo de las 1000 solicitudes/segundo, aunque el tiempo de respuesta promedio fue sumamente bueno comportándose por debajo de 1 segundo.

Capítulo III: Análisis de los Resultados

3.3.3 Funcionalidad Definir Propuesta

# Muestras	Tiempo Promedio	% Error	Rendimiento	Kb/seg
900	0.001	11.11	441.4/seg	45.3

El rendimiento no tuvo el valor esperado, sin embargo el tiempo de respuesta fue muy bueno.

3.3.4 Funcionalidad Definir Repetidor

# Muestras	Tiempo Promedio	% Error	Rendimiento	Kb/seg
900	0.001	11.11	446.7/seg	45.8

Como puede observarse el tiempo promedio de respuesta fue excelente, muy por debajo del segundo, pero el rendimiento fue menor a la mitad del valor esperado.

3.3.5 Funcionalidad Realizar Estudio API

# Muestras	Tiempo Promedio	% Error	Rendimiento	Kb/seg
1800	0.002	5.56	832.6/seg	75.2

El rendimiento fue inferior al valor esperado, pero el tiempo de respuesta fue bueno.

3.3.6 Funcionalidad Gestionar PIA por Jefe LCF AGR

# Muestras	Tiempo Promedio	% Error	Rendimiento	Kb/seg
4500	1.102	6.67	22.2/seg	427.9

Mantuvo un buen tiempo de respuesta promedio, sin embargo el rendimiento fue muy malo.

3.3.7 Funcionalidad Monitorizar Viajeros

# Muestras	Tiempo Promedio	% Error	Rendimiento	Kb/seg
2600	0.001	3.85	1203.7/seg	445.2

Esta funcionalidad se realizó con un rendimiento mayor al valor esperado y también tuvo un buen tiempo de respuesta.

Capítulo III: Análisis de los Resultados

3.3.8 Funcionalidad Mostrar Estudio API

# Muestras	Tiempo Promedio	% Error	Rendimiento	Kb/seg
1400	0.001	7.14	676.7/seg	63.5

El tiempo de respuesta fue excelente, como puede observarse, pero el rendimiento volvió a estar por debajo de lo esperado.

3.3.9 Funcionalidad Mostrar Infracciones por Fecha

# Muestras	Tiempo Promedio	% Error	Rendimiento	Kb/seg
1700	0.001	5.88	810.7/seg	73.8

El rendimiento estuvo cerca de las 1000 solicitudes/segundo, pero aún así fue inferior a este valor. El tiempo promedio a su vez fue muy bueno.

3.3.10 Funcionalidad Mostrar Infracciones por Personas

# Muestras	Tiempo Promedio	% Error	Rendimiento	Kb/seg
1400	0.001	7.14	677.0/seg	63.5

Excelente el tiempo promedio de respuesta, pero el rendimiento no fue el esperado.

3.3.11 Funcionalidad Mostrar PIA

# Muestras	Tiempo Promedio	% Error	Rendimiento	Kb/seg
1000	4.89	10	494.1/seg	49.5

El tiempo promedio de respuesta es superior al de las funcionalidades anteriores, sin embargo no deja de ser un valor bueno, inferior a los 5 seg. El rendimiento a su vez, es menor al esperado.

3.3.12 Funcionalidad Mostrar Posibles Redes de Vínculos

# Muestras	Tiempo Promedio	% Error	Rendimiento	Kb/seg
1100	2.179	9.09	11.1/seg	15

Capítulo III: Análisis de los Resultados

El tiempo de respuesta no constituyó un problema, pero el rendimiento obtenido fue malo para los valores que se esperaban.

3.3.13 Funcionalidad Mostrar Propuesta

# Muestras	Tiempo Promedio	% Error	Rendimiento	Kb/seg
1000	0.001	10	491.4/seg	49

Se obtuvo un tiempo promedio de respuesta excelente, el rendimiento no fue tan malo en comparación con otros obtenidos.

3.3.14 Funcionalidad Mostrar Requerimiento Puntual

# Muestras	Tiempo Promedio	% Error	Rendimiento	Kb/seg
6900	2.048	1.45	12.1/seg	111.8

El rendimiento para esta funcionalidad con 100 usuarios fue malo, pero mantuvo un buen tiempo promedio de respuesta.

3.3.15 Funcionalidad Mostrar Registrar Incidencia

# Muestras	Tiempo Promedio	% Error	Rendimiento	Kb/seg
1900	2.112	5.32	9.2/seg	11.1

Con esta funcionalidad sucede lo mismo que en el caso anterior. Aunque el rendimiento fue peor.

3.3.16 Funcionalidad Insertar Requerimiento Puntual

# Muestras	Tiempo Promedio	% Error	Rendimiento	Kb/seg
4300	1.549	4.77	14.4/seg	302.3

El rendimiento se comportó muy por debajo de lo esperado, el tiempo promedio de respuesta para 100 usuarios conectados fue bueno.

3.3.17 Funcionalidad Ver Historial

# Muestras	Tiempo Promedio	% Error	Rendimiento	Kb/seg
------------	-----------------	---------	-------------	--------

Capítulo III: Análisis de los Resultados

1300	0.002	7.69	629.5/seg	59.8
------	-------	------	-----------	------

El rendimiento no es el esperado para esta funcionalidad, pues es menor que 1000/seg, pero el tiempo promedio de respuesta fue excelente.

3.3.18 Resultados generales de las pruebas de estrés

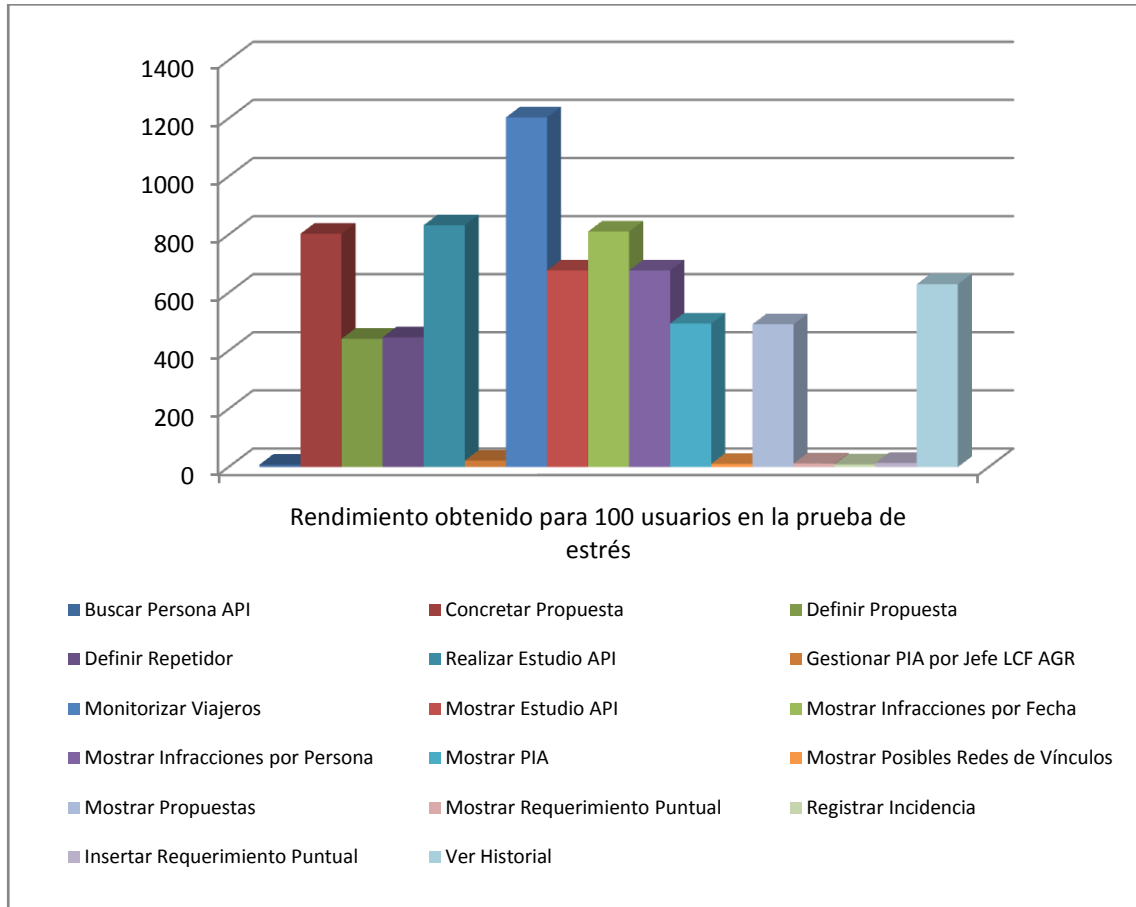


Figura 11: Gráfico que muestra el rendimiento hasta 100 usuarios en la prueba de estrés.

Capítulo III: Análisis de los Resultados

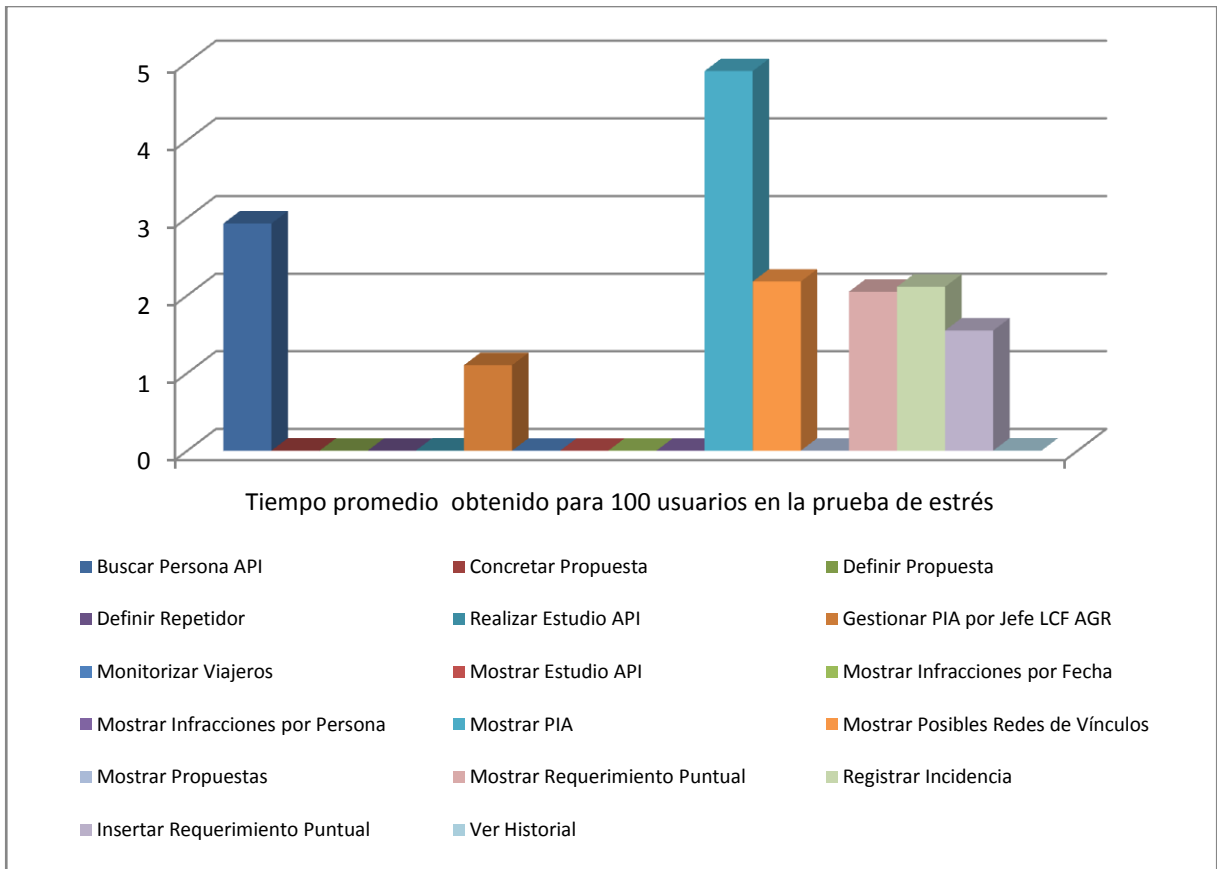


Figura 12: Gráfico que muestra el tiempo promedio de respuesta para 100 usuarios en la prueba de estrés.

Con la realización de las pruebas de estrés a las funcionalidades del sistema se observó que el rendimiento no es bueno. Los tiempos promedio de respuestas se mantuvieron por debajo de los 5 segundos, siendo tiempos excelentes, porque cumplen con los valores establecidos. Se fueron conectando usuarios hasta alcanzar la cifra de 100, pero solo una funcionalidad, Monitorizar Viajeros, tuvo un rendimiento por encima de las 1000 solicitudes/segundo. Otras 3 funcionalidades, Concretar Propuesta, Realizar Estudio API y Mostrar Infracciones por Fecha, estuvieron cerca de las 800 solicitudes/segundo. Las restantes 12 funcionalidades no fueron eficientes. El error obtenido fue el mismo que se detectó en las pruebas de carga, no siendo un error relevante, debido a que es una imagen que nunca se cargó. Las pruebas demostraron que solo se pueden conectar hasta 100 usuarios, para 150 y 200 usuarios el sistema no respondió. Estos 100 usuarios pudieron conectarse pero el rendimiento del sistema no fue el esperado.

Capítulo III: Análisis de los Resultados

3.4 Características del servidor central de la AGR

El rendimiento del sistema está estrictamente ligado con el hardware de la computadora que realizará la función de servidor, ya que éste determina el desempeño del mismo. La computadora utilizada como servidor para estas pruebas posee características que le infieren menor calidad que las que poseerá el entorno real donde correrá la aplicación. Las características que poseerá el servidor real son las que se brindan a continuación:

Características de Hardware y Software	
Microprocesador	Dual-Core AMD Opteron
	4 núcleos a 1,8GHz c/u
Memoria RAM	4 Gb
Capacidad del disco	2 discos de 320 Gb espejados por software
Sistema Operativo	Distribución: Ubuntu
	Descripción: Ubuntu 8.04.4 LTS
	Nombre: Hardy Heron

Este servidor tiene la base de datos y el servidor de aplicaciones (apache) juntos.

3.5 Aportes obtenidos

En un proyecto de desarrollo de un software, es de vital importancia especificar los requisitos no funcionales, ya que estos se refieren a las propiedades emergentes del sistema como la fiabilidad, el tiempo de respuesta, la capacidad de almacenamiento, seguridad entre otras.

Por cuestiones de tiempo y priorización de los objetivos se decidió centrar el foco de atención en los requisitos funcionales del módulo Control de Personas. Es por ello que se decide un año después especificar los requisitos no funcionales del módulo para identificar la habilidad esperada de los distintos grupos de usuarios de acuerdo con las funciones que realizan, los conocimientos y habilidades que poseen en el entorno en el que trabajan, así como la velocidad para completar una tarea. También se

Capítulo III: Análisis de los Resultados

contemplan todos los aspectos relacionados con la seguridad, integridad y confidencialidad de los datos que procesa, maneja, almacena y recupera el sistema. Se obtuvo la especificación de estos requisitos no funcionales que ayudan a definir las características que el sistema debe cumplir.

Con las características del entorno donde fueron ejecutadas las pruebas se detectaron funcionalidades que no fueron eficientes para el número mínimo de usuarios con las que fueron probadas. Estas funcionalidades son: Gestionar PIA por jefe LCF AGR, Buscar persona API, Mostrar posibles redes de vínculos, Insertar Requerimiento Puntual, Mostrar Requerimiento Puntual y Registrar Incidencia, la cuales son de gran importancia para los usuarios finales. Se deben revisar las funcionalidades deficientes por parte del equipo de desarrollo para tratar de optimizarlas, además de revisar el error encontrado en todas las pruebas para la imagen que no llega a cargarse, esta se encuentra en la dirección: /GINA/web/images/loading.gif. Por otra parte la utilización del CPU para estas funcionalidades sobrepasó el 70% influyendo negativamente en los resultados obtenidos. Se debe revisar la funcionalidad Buscar Persona API debido a que mostró tiempos de respuestas elevados. Estos resultados obtenidos permitirán a la dirección del CADI valorar si se puede o no centralizar toda la información referente al módulo Control de Personas en un servidor central. La centralización de esta información en un servidor es de gran importancia debido a que atenderá a un gran número de usuarios, sin necesidad de instalar y almacenar la información localmente en cada punto aduanal del país, lo que evitará gastos innecesarios en cuanto a transporte para realizar las actualizaciones de nuevas versiones del módulo, la compra de servidores para cada punto aduanal y la capacitación del personal que trabajará directamente con el sistema.

3.6 Conclusiones parciales

En este capítulo se realizó un análisis de los resultados obtenidos en las pruebas, tanto para las de carga como las de estrés. Dichas pruebas ayudaron a detectar la presencia de errores, que se evidenciaron cuando el sistema no respondió como se esperaba o cuando simplemente no respondió. Las funcionalidades que presentaron problemas de rendimiento y no son funcionales para una carga de 100 usuarios, con el entorno de pruebas creado son: Buscar Persona API, Gestionar PIA por Jefe LCF AGR, Mostrar Posibles Redes de Vínculos, Mostrar e Insertar Requerimiento Puntual y Registrar Incidencia. Estas funcionalidades mostraron valores que no cumplieron con las especificaciones de requisitos, demostraron ser ineficientes en el entorno donde fueron probadas. Queda validado el RNF Eficiencia para el módulo Control de Personas del sistema GINA, el mismo no cumple con su especificación para las funcionalidades a las que se ha hecho referencia. Para un entorno similar al utilizado para las pruebas el

Capítulo III: Análisis de los Resultados

módulo Control de Personas no es eficiente, hay funcionalidades que requieren un hardware con características superiores a las que se tenían. El servidor donde estará corriendo el módulo poseerá características superiores a las que se tienen, posibilitando que el sistema se pueda ejecutar con mejores resultados a los obtenidos en las pruebas realizadas.

CONCLUSIONES

- ✓ Concluido el presente trabajo se puede afirmar que los objetivos trazados para el mismo fueron logrados exitosamente.
- ✓ Quedaron especificados los requisitos no funcionales del módulo *Control de Personas* del sistema GINA.
- ✓ Se ejecutaron los casos de prueba necesarios para verificar el rendimiento de las funcionalidades del módulo.
- ✓ Se validó el requisito no funcional Eficiencia mediante las pruebas de rendimiento realizadas.
- ✓ Se realizó una comparación de los resultados obtenidos con los resultados que se esperaban alcanzar.
- ✓ Han sido detectadas y documentadas las funcionalidades deficientes y los errores encontrados para su posterior revisión, en aras de lograr un producto con mayor calidad.

RECOMENDACIONES

Se recomienda:

- ✓ Revisar las funcionalidades deficientes obtenidas para optimizar su código en caso de ser posible.
- ✓ Revisar los errores encontrados para mejorar la calidad del módulo.
- ✓ Realizar las pruebas nuevamente para verificar si los errores fueron corregidos.
- ✓ Continuar con el uso de la herramienta JMeter para pruebas de carga y estrés en el Departamento Soluciones para la Aduana.

BIBLIOGRAFÍA

1. **Olsina, L.** *Metodología cualitativa para la evaluación y comparación de la calidad de sitios web.* 1999.
2. **Bañeres, Juan Palacio.** *Compendio de Ingeniería de Software 1.* 2006.
3. **Jacobson, I.** *Modeling with use cases-Formalizing use-case modelling.* 1995.
4. **M. G Piattini, J. A. Calvo-Manzano, J. Cervera, L. Fernández.** *Análisis y diseño detallado de Aplicaciones informáticas de gestión.* 2006.
5. **Pressman, Roger S.** *Ingeniería de Software. Un enfoque práctico.* 2006.
6. **Pressman, Roger S.** *Ingeniería de Software. Un enfoque Práctico.* 2005.
7. **Sommerville.** *Ingeniería de Software.* 2005.
8. Entorno Virtual de Aprendizaje. [Online] <http://eva.uci.cu/course/view.php?id=102>. [Cited: 02 14, 2011.]
9. **IEEE.** *Computer Dictionary. Computer Society.* 1990.
10. **Cosín, J. D.** *Técnicas cuantitativas para la gestión en la ingeniería del software.* 2007.
11. **Myers.** *The art of software testing.* 1979.
12. **Quesada, Juan Antonio López.** *Fundamentos de Ingeniería del Software.* . 2005.
13. Comité Estatal de Tecnologías de la Información. [Online] [Cited: 02 15, 2011.] <http://www.cetic.guerrero.gob.mx/pics/art/articles/113/file.TiposPruebasSoftware.pdf>.
14. **Pressman, R. S.** *Ingeniería del Software. Un enfoque práctico. Quinta Edición.* 2001.
15. Opendemand Systems. [Online] <http://www.opendemand.com>. [Cited: 03 01, 2011.]
16. Osalt. [Online] <http://www.osalt.com/es/opensta>. [Cited: 03 01, 2011.]
17. Funkload. [Online] <http://funkload.nuxeo.org>. [Cited: 03 01, 2011.]
18. Webload. [Online] <http://www.webload.org>. [Cited: 05 01, 2011.]
19. Apache Jakarta Project. [Online] <http://jakarta.apache.org/jmeter/>. [Cited: 03 01, 2011.]
20. Web Application Load. [Online] <http://www.loadtestingtool.com/pro.shtml>. [Cited: 05 01, 2011.]

21. **Almenares, Liudmila Sánchez.** *Cómo realizar Pruebas de Carga y Estrés en JMeter.* 2008.
22. **Frómeta, Anay C. Barban, Figueredo, Paula Z. Hernández.** *Procedimiento para la realización de pruebas rendimiento de carga y estrés al Sistema Único de Aduanas.* 2009.

GLOSARIO

Ajax: Acrónimo de Asynchronous JavaScript And XML. Es una técnica de desarrollo web para crear aplicaciones informáticas interactivas.

Calidad de Software: Se puede decir que el software tiene calidad si cumple o excede las expectativas del usuario, es decir el cumplimiento de los requisitos.

Cookies: Pequeños archivos que algunos sitios web guardan en la computadora. Las cookies almacenan información sobre el usuario, como nombre de usuario o información de registro, o preferencias de usuario.

CPU: La unidad central de procesamiento o CPU (acrónimo en inglés de Central Processing Unit), también conocido por procesador o microprocesador, es el componente de la computadora y otros dispositivos programables, que interpreta las instrucciones contenidas en los programas y procesa los datos.

ExtJS: Es una biblioteca JavaScript ligera y de alto rendimiento, compatible con la mayoría de navegadores que permite crear páginas e interfaces web dinámicas.

Json: Acrónimo de JavaScript Object Notation. Es un formato ligero para el intercambio de datos

Localhost: Es el nombre reservado que tienen todas las computadoras, routers o dispositivos que disponen de una tarjeta de red Ethernet para referirse a sí mismo.

Hardware: Conjunto de componentes físicos de una computadora, por ejemplo los discos, lectores de discos, monitores, teclados, las impresoras, tarjetas y chips.

URL: Acrónimo en inglés de Uniform Resource Locator (Localizador Uniforme de Recursos). Se trata de la secuencia de caracteres que sigue un estándar y que permite nombrar recursos en Internet o una red más pequeña (Intranet) para su localización.

Proxy: Programa o dispositivo que sirve de intermediario, permite a otros equipos conectarse a una red de forma indirecta a través de él. Cuando un equipo de la red desea acceder a una información o recurso, es realmente el proxy quien realiza la comunicación y a continuación traslada el resultado al equipo inicial. En unos casos esto se hace así porque no es posible la comunicación directa y en otros casos porque el

proxy añade una funcionalidad adicional, como puede ser la de mantener los resultados obtenidos (ejemplo una página web) en una caché que permita acelerar sucesivas consultas coincidentes.

Switch: Palabra que significa conmutador, es un dispositivo que permite la interconexión de redes sólo cuando esta conexión es necesaria. La red está dividida en segmentos por lo que, cuando se envía un mensaje desde un segmento hacia otro segmento determinado, el switch se encargará de hacer que ese mensaje llegue única y exclusivamente al segmento requerido. Opera en la capa 2 del modelo OSI, que es el nivel de enlace de datos. Funcionan como un filtro en la red, mejoran el rendimiento y la seguridad de las redes de área local.