

Universidad de las Ciencias Informáticas



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



*Título: Propuesta de manual de procedimiento de Pruebas de Sistema
y su aplicación en el Proyecto CICPC*

Autoras: Yuniet del Carmen Toll Palma

Ylennis Mendoza Torres

Tutora: Ing. Haydée María Cruz Torres MSc.

“Ciudad de la Habana” Junio 2007

“Año del 49 Aniversario de la Revolución Cubana”

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 Julio del año 2007.

Yuniet del Carmen Toll Palma

Yilennis Mendoza Torres

Firma de la Autora

Firma de la Autora

Ing. Haydée María Torres Cruz. MSc.

Firma de la Tutora

DATOS DE CONTACTO

Tutora: Ing. Haydée María Cruz Torres. MSc.

Correo electrónico: hmacruz@uci.cu

Teléfono: 8372476

Departamento de Ciencias Informáticas

Facultad 8

Asesor: Lic. Lázaro Luis Castillo García, Especialista de Calidad de Desoft S.A.

Correo electrónico: lazaro.castillo@hab.desoft.cu

Teléfono de la empresa Desoft S.A. 835 63 70

Consultante: Ing. Nilet María Soto López

Correo electrónico: nsoto@uci.cu

Líder del Proyecto CICPC.

“las pruebas no pueden asegurar la ausencia de defectos; solo pueden demostrar que existen defectos en el software”

Glen Myers

AGRADECIMIENTOS

Agradecemos a nuestros padres, familiares y amigos(as) que se preocuparon en todo momento por el avance del trabajo, por su apoyo incondicional, deseándonos siempre éxitos en el mismo, para nuestro desempeño profesional en el futuro.

Agradecemos al Ing. Yohandri Ril Gil por su colaboración en el uso de herramientas que facilitaban el diseño y confección del trabajo.

Le agradecemos al Decano de nuestra facultad Dr. Alcides Cabrera Campos por el apoyo que nos brindó en la metodología de la investigación, también a nuestra líder del Proyecto CICPC Ing. Nilet María Soto López por toda la ayuda brindada en la confección de la propuesta de estrategia para el proyecto CICPC y por todas las ideas que nos aportó para mejorar el Manual de Procedimiento, a la segunda líder Irina Cancela Nieto por la gran ayuda que nos brindó en todo el proceso de ejecución de las pruebas de sistema a los casos de estudio y al Ing. Edier García Gutiérrez por su gran apoyo cada vez que lo necesitamos.

Agradecemos a nuestra tutora Ing. Haydée María Torres Cruz por su ayuda durante todo el desarrollo investigativo de este trabajo, y a nuestro asesor Lic. Lázaro Luis Castillo García el cual nos ayudó muchísimo, sin ellos no habríamos podido llegar a confeccionar un trabajo con calidad y profesionalidad.

Mencionándolos no basta para agradecerles todo su apoyo; sin embargo es la única forma de dejar constancia de lo que todos ustedes han hecho por nosotras.

A todos los que de una forma u otra hicieron posible que llegáramos hasta aquí.

Agradecemos a nuestra revolución y a la Universidad por darnos la gran oportunidad de formar parte de este gran proyecto y facilitarnos los medios que permitieron nuestra formación como futuras Ingenieras en Ciencias Informáticas.

DEDICATORIA

Autora: Yuniet del Carmen Toll Palma.

Le dedico este trabajo a mis padres María Palma Rodríguez y Erlán Toll Laurencio, los que me dieron la vida, los que ni dedicándole mi vida entera lograría recompensarlos por tan bella obra, a mi esposo Yohandri Ril Gil, por su cariño, comprensión, su apoyo en todo momento, a mi tía Mercedes Castillo González, que siempre me dio su apoyo, consejo y a Dios, por darme toda la paciencia y sabiduría que necesité para esforzarme al máximo para dar todo de mí en este trabajo, sin ellos no hubiese podido llegar hasta aquí, por ello me esforcé todo cuanto pude y les dedico íntegramente todo mi trabajo.

Autora: Yilennis Mendoza Torres

Le dedico este trabajo en especial a mi abuela María D. Hernández Reyes que aunque no pudo ver el fruto de mi esfuerzo terminado, lo hice pensando siempre en cada granito de arena que aportó en mí, en el deseo de verme graduada para que saliera adelante y me hiciera una profesional como siempre anheló. A mi madre Rosa M. Torres Hernández tan dedicada siempre a mí, quién me guío por el mejor camino y me aconseja en todo momento lo que debo hacer y lo que me conviene para el futuro, la que me alienta en los momentos más difíciles, me apoya incondicionalmente y se preocupa en todo momento por mi bienestar y mi futuro profesional. Además le dedico este trabajo a la familia Lesbia Cordero Llerena quiénes me apoyaron, me cuidaron, me aconsejaron, me guiaron como una hija más y pusieron todo su empeño para que realizara mis sueños: ser una futura ingeniera informática. También agradecerles a todos los amigos(as) y todas aquellas personas que hicieron posible que culminaran mis estudios satisfactoriamente y pudiera llegar hasta aquí.

RESUMEN

Este trabajo propone un manual de procedimiento para realizar pruebas de sistema en el Proyecto productivo CICPC. Aparece una breve panorámica acerca de la gestión de calidad de software en el mundo, del proceso de pruebas y específicamente la aplicación de pruebas de sistema, así como los diferentes tipos de prueba que se le realizan al producto de software, además se realiza un análisis de la estrategia a utilizar para las pruebas de sistema, es decir una estrategia para pruebas funcionales con los casos de pruebas correspondientes según los requisitos especificados y otra estrategia para pruebas no funcionales que constan de varios pasos a seguir según los requisitos no funcionales descritos y en caso de la existencia de un requisito particular se confecciona un caso de prueba para ese tipo de prueba, además se investiga sobre los tipos de pruebas de sistema que existen, también se hizo un estudio de algunas herramientas que pueden utilizarse para aplicar pruebas a un software. Además se muestra la validez del procedimiento a través de su aplicación en Casos de Estudio implementados en el Proyecto conjuntamente con un análisis de los principales problemas detectados. Con este manual se logra ganar en organización, experiencia, madurez, rapidez, habilidad y control de las pruebas en el Proyecto.

PALABRAS CLAVE

Calidad, Prueba, Prueba de Sistema, Requisitos Funcionales, Requisitos No Funcionales, Casos de Prueba, Manual de Procedimiento.

TABLA DE CONTENIDOS

INTRODUCCIÓN1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA6

1.1 INTRODUCCIÓN DEL CAPÍTULO6

1.2 CALIDAD DE SOFTWARE6

1.3 PRUEBAS DE SOFTWARE7

1.4 TIPOS DE PRUEBAS10

1.5 TIPOS DE PRUEBAS DE SISTEMA11

 1.5.1 Pruebas de Stress o Resistencia11

 1.5.2 Pruebas de Rendimiento o Carga11

 1.5.3 Pruebas de Seguridad11

 1.5.3 Prueba de Volumen Funcional.....12

 1.5.4 Prueba de Recuperación12

 1.5.5 Prueba de Compatibilidad12

 1.5.6 Prueba de Usabilidad.....12

 1.5.7 Pruebas de Concurrencia o Contención13

 1.5.8 Prueba de Funcionalidad.....13

 1.5.9 Prueba de Disponibilidad y Red13

 1.5.10 Prueba de Instalación14

 1.5.11 Prueba de Configuración.....14

 1.5.12 Prueba de Negativas14

 1.5.13 Prueba de Vulnerabilidad.....14

 1.5.14 Prueba de Fiabilidad de Sistema14

1.6 ESTRATEGIA DE PRUEBA DE SOFTWARE15

 1.6.1 Descripción de la estrategia de prueba de software15

 1.6.2 Estrategia de Prueba Funcional (requisitos funcionales).....16

 1.6.2.1 Técnicas de diseño de pruebas16

1.6.2.1.1 Técnicas de caja negra19

 1.6.3 Estrategia de Prueba no Funcional (requisitos no funcionales).....22

1.8 DESCRIPCIÓN DEL SOFTWARE OBJETO DE PRUEBA31

 1.8.1 Sistema integrado de información policial.....31

 1.8.2 Portal Web34

1.9 CONCLUSIONES DEL CAPÍTULO35

CAPÍTULO 2: MANUAL DE PROCEDIMIENTO.....37

2.1 INTRODUCCIÓN DEL CAPÍTULO37

2.2 CICLO COMPLETO DE LAS PRUEBAS37

 2.3.1 Planificación de las pruebas.39

 2.3.2 Diseño de las pruebas de Sistema46

 2.3.2.1 Estrategia de Prueba Funcional (requisitos funcionales).....47

 2.3.2.2 Estrategia de Prueba no Funcional (requisitos no funcionales).....52

 2.3.3 Ejecución de las Pruebas del Sistema56

 2.3.4 Evaluación del resultado de las pruebas de Sistema57

2.4 CONCLUSIONES DEL CAPÍTULO57

CAPÍTULO 3: EVALUACIÓN DE LOS RESULTADOS59

3.1 INTRODUCCIÓN DEL CAPÍTULO59

3.1.1 Organización y planificación de las pruebas.....	59
3.2 RESULTADOS OBTENIDOS DEL CASO DE ESTUDIO DE FICHA TÉCNICA DEL PROYECTO DE CICPC	60
3.2.1 Resultados obtenidos.....	61
3.2.1.1 Pruebas Funcionales.....	61
3.2.1.2 Pruebas de Seguridad	74
3.2.1.3 Pruebas de Stress o Resistencia	76
3.2.1.4 Pruebas de Disponibilidad y Red	77
3.2.1.5 Pruebas de Compatibilidad	79
3.2.2 Representación Gráfica de los resultados.....	84
3.2.2.2 Representación Gráfica según estas subcaracterísticas o atributos de calidad	86
3.2.3 Análisis de los resultados de las pruebas realizadas	91
3.3 RESULTADOS OBTENIDOS DEL CASO DE ESTUDIO SOPORTE TÉCNICO DEL PROYECTO DE CICPC.....	92
3.3.1 Resultados Obtenidos.....	93
3.3.1.1 Pruebas Funcionales.....	93
3.3.1.2 Pruebas de Stress o Resistencia	103
3.3.1.3 Pruebas de Seguridad	104
3.3.1.4 Pruebas de Fiabilidad.....	105
3.3.1.5 Pruebas de Compatibilidad	107
3.3.1.6 Pruebas Rendimiento.....	109
3.3.2 Representación Gráfica.....	110
3.3.3 Representación gráfica según estas subcaracterísticas o atributos de calidad	113
3.3.4 Análisis de las pruebas realizadas	119
3.4 ANÁLISIS DE LA DISTRIBUCIÓN DE LOS PROBADORES POR TIPO DE PRUEBA EN AMBOS CASOS DE ESTUDIO.....	120
3.5 ANÁLISIS DE LOS RESULTADOS OBTENIDOS EN AMBOS CASOS DE ESTUDIO FICHA TÉCNICA Y SOPORTE TÉCNICO.....	121
3.6 CONCLUSIONES DEL CAPÍTULO	123
CONCLUSIONES	124
RECOMENDACIONES.....	125
REFERENCIAS BIBLIOGRÁFICAS	126
BIBLIOGRAFÍA	128
GLOSARIO DE TÉRMINOS	132

INDICE DE FIGURAS

FIG. 1 ENFOQUE FUNCIONAL Y ENFOQUE ESTRUCTURAL.....	18
FIG. 2 TÉCNICAS ASOCIADAS A LAS ETAPAS DEL PROCESO DE PRUEBA.	28
FIG. 3 CICLO DE VIDA DE RUP	30
FIG. 4 CICLO COMPLETO DE LAS PRUEBAS	38
FIG. 5 DISTRIBUCIÓN DE PROBADORES.....	84
FIG. 6 NO CONFORMIDADES	85
FIG. 7 SUGERENCIAS.....	85
FIG. 8 NO CONFORMIDADES POR PROBADORES.....	86
FIG. 9 SUGERENCIAS POR PROBADORES	86
FIG. 10 NO CONFORMIDADES POR TIPO DE PRUEBA Y ATRIBUTOS DE CALIDAD.....	89
FIG. 11 SUGERENCIAS POR TIPO DE PRUEBA Y ATRIBUTOS DE CALIDAD	90
FIG. 12 PUNTUACIÓN.....	90
FIG. 13 ESCALA.....	91
FIG. 14 DISTRIBUCIÓN DE PROBADORES DE SOPORTE TÉCNICO.....	111
FIG. 15 NO CONFORMIDADES	112
FIG. 16 SUGERENCIAS	112
FIG. 17 NO CONFORMIDADES POR PROBADORES	113
FIG. 18 SUGERENCIAS POR PROBADORES.....	113
FIG. 19 NO CONFORMIDADES POR TIPO DE PRUEBA Y ATRIBUTOS DE CALIDAD.....	117
FIG. 20 SUGERENCIAS POR TIPO DE PRUEBA Y ATRIBUTOS DE CALIDAD	117
FIG. 21 PUNTUACIÓN.....	118
FIG. 22 ESCALA.....	119
FIG. 23 RESULTADOS GENERALES.....	123

INDICE DE TABLAS

TABLA 1 EJEMPLO DE ENTRADA DE PROGRAMA	21
TABLA 2 CARACTERÍSTICAS, SUBCARACTERÍSTICAS Y ATRIBUTOS DE CALIDAD SEGÚN LA ISO 9126	23
TABLA 3 DESCRIPCIÓN DE LAS CLASES VÁLIDAS Y CLASES INVÁLIDAS	47
TABLA 4 DESCRIPCIÓN DEL CASO DE PRUEBA.....	48
TABLA 5 EJEMPLO DE CASO DE PRUEBA.....	49
TABLA 6 EJEMPLO PARA IDENTIFICAR CLASES Y LA GENERACIÓN DE CASOS DE PRUEBAS	51
TABLA 7 DESCRIPCIÓN DE LA TABLA DE NO CONFORMIDADES PARA PRUEBAS FUNCIONALES.....	52
TABLA 8 EJEMPLO DE CASOS DE PRUEBA PARA LA PRUEBA DE SEGURIDAD	53
TABLA 9 DESCRIPCIÓN DE DE LA TABLA DE NO CONFORMIDADES PARA PRUEBAS NO FUNCIONALES.....	55
TABLA 10 RESULTADOS DE LAS PRUEBAS DE SISTEMA AL CASO DE ESTUDIO FICHA TÉCNICA.....	61
TABLA 11 RESULTADOS DE LAS PRUEBAS DE SISTEMA AL CASO DE ESTUDIO FICHA TÉCNICA.....	74
TABLA 12 RESULTADOS DE LAS PRUEBAS DE SISTEMA AL CASO DE ESTUDIO FICHA TÉCNICA.....	76
TABLA 13 RESULTADOS DE LAS PRUEBAS DE SISTEMA AL CASO DE ESTUDIO FICHA TÉCNICA.....	77
TABLA 14 RESULTADOS DE LAS PRUEBAS DE SISTEMA AL CASO DE ESTUDIO FICHA TÉCNICA.....	79
TABLA 15 DATOS POR TIPO DE PRUEBA.....	84
TABLA 16 CARACTERÍSTICAS Y ATRIBUTOS DE LA ISO 9126 ASOCIADOS A LAS PRUEBAS DE SISTEMA.....	87
TABLA 17 MÉTRICAS DE CALIDAD POR TIPO DE PRUEBAS DE SISTEMA Y ATRIBUTOS ASOCIADOS SEGÚN LA ISO 9126.....	87
TABLA 18 ESCALA	91
TABLA 19 RESULTADOS DE LAS PRUEBAS DE SISTEMA AL CASO DE ESTUDIO SOPORTE TÉCNICO	93
TABLA 20 RESULTADOS DE LAS PRUEBAS DE SISTEMA AL CASO DE ESTUDIO SOPORTE TÉCNICO	103
TABLA 21 RESULTADOS DE LAS PRUEBAS DE SISTEMA AL CASO DE ESTUDIO SOPORTE TÉCNICO	104
TABLA 22 RESULTADOS DE LAS PRUEBAS DE SISTEMA AL CASO DE ESTUDIO SOPORTE TÉCNICO	105
TABLA 23 RESULTADOS DE LAS PRUEBAS DE SISTEMA AL CASO DE ESTUDIO SOPORTE TÉCNICO	107
TABLA 24 RESULTADOS DE LAS PRUEBAS DE SISTEMA AL CASO DE ESTUDIO SOPORTE TÉCNICO	109
TABLA 25 DATOS PARA MEDIR LAS PRUEBAS	111
TABLA 26 CARACTERÍSTICAS Y ATRIBUTOS DE LA ISO 9126 ASOCIADOS A LAS PRUEBAS DE SISTEMA.....	114
TABLA 27 MÉTRICAS DE CALIDAD POR TIPO DE PRUEBAS DE SISTEMA Y ATRIBUTOS ASOCIADOS SEGÚN LA ISO 9126.....	114
TABLA 28 ESCALA	118
TABLA 29 MÉTRICAS DE CALIDAD POR TIPO DE PRUEBAS DE SISTEMA Y ATRIBUTOS ASOCIADOS SEGÚN LA ISO 9126.....	121
TABLA 30 MÉTRICAS DE CALIDAD POR TIPO DE PRUEBAS DE SISTEMA Y ATRIBUTOS ASOCIADOS SEGÚN LA ISO 9126.....	121

INTRODUCCIÓN

La entrega de los productos con calidad le aporta al país y a la institución un gran prestigio, eleva el desarrollo en el campo tecnológico, y aumenta la fiabilidad en el trabajo realizado.

Hoy en día la calidad está de moda, en todos los aspectos, pero especialmente en el desarrollo de software. El interés por la calidad está creciendo de forma continua, a medida que los clientes se vuelven más selectivos y comienzan a rechazar los productos poco fiables o que realmente no dan respuesta a sus necesidades (Quesada 2007).

Cuando se va a definir la calidad del software se pueden adoptar diferentes aproximaciones. Como primera aproximación está la importancia de saber diferenciar entre la calidad del Producto software y la calidad del Proceso de desarrollo. No obstante, las metas que se establezcan para la calidad del producto van a determinar las metas a establecer para la calidad del proceso de desarrollo, ya que la calidad del producto va a estar en función de la calidad del proceso de desarrollo. Sin un buen proceso de desarrollo es casi imposible obtener un buen producto (Quesada 2007).

Según Luis Fernández *“la calidad del software debe ser una disciplina más dentro de la Ingeniería del software”* (Fernández 1995). El principal instrumento para garantizar la calidad de las aplicaciones sigue siendo el Plan de Calidad. El plan se basa en normas o estándares genéricos y en procedimientos particulares (Fernández 1995).

Hay algo muy cierto y es que el proceso de Pruebas de Software o Proceso de Verificación se ejecuta en el sistema con el objetivo de detectar fallos (Quesada 2007).

Javier J. Gutiérrez y otros (Gutiérrez 2006) plantean:

“La fase de pruebas de sistema tiene como objetivo verificar el sistema de software para comprobar si este cumple sus requisitos. Dentro de esta fase pueden desarrollarse varios tipos de prueba en función de los objetivos de las mismas. Algunos tipos son pruebas funcionales, prueba de usabilidad, prueba de rendimiento, prueba de seguridad, etc. Las pruebas funcionales verifican que el sistema de software ofrece a los actores humanos la funcionalidad recogida en su especificación”.

En el mundo de las pruebas de software con el objetivo de medir la calidad de todo producto, existen varios niveles de pruebas. Se encuentran las Pruebas de Unidad que consisten en la prueba de cada módulo de forma aislada del resto del sistema, Pruebas de Integración que son las que se realizan a medida que los diferentes módulos del sistema se integran en el mismo, Pruebas de Sistema que se realizan cuando se han integrado todos los módulos, y tiene como objetivo principal comprobar que el sistema satisface los requisitos del usuario y Pruebas de Aceptación que son las que se realizan una vez que el sistema se ha implantado en su entorno real de funcionamiento (Antonio 1999). El presente trabajo propone un conjunto de pasos a seguir para realizar las pruebas de sistema a un producto de software determinado.

Las pruebas de sistema tienen una amplia diversidad, se realizan para comprobar que los requisitos pedidos por el usuario funcionen correctamente, a continuación se muestran las siguientes (Solutions 1996):

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

Para garantizar y controlar la calidad en el proyecto CICPC durante el proceso de las pruebas de sistema, es necesario contar con una serie de pasos que puedan servir de guía en el momento de llevar a cabo dichas pruebas de forma organizada. Debido, tanto al volumen de las pruebas como a la complejidad de las mismas, que van desde entrar grandes cantidades de datos, verificar la funcionalidad del sistema, hasta verificar la rapidez o el tiempo de respuestas de las aplicaciones, surge la necesidad de tener un manual de procedimiento para planificar, diseñar, ejecutar y evaluar estas pruebas, permitiendo el control y seguimiento durante esta etapa. Actualmente la dirección de calidad de la Universidad de las Ciencias Informáticas no cuenta con una estrategia que oriente cómo realizar este tipo de prueba.

Dada la **situación** anteriormente planteada se definió el **problema científico** como la no existencia de un manual de procedimiento que permita aplicar eficientemente las Pruebas de Sistema en el Proyecto productivo Cuerpo de Investigaciones Científicas, Penales y Criminalísticas (CICPC).

Tomando como **objeto de investigación** las Pruebas de Sistema en el proceso de verificación de un producto de Software.

El **campo de acción** en el que estuvo enmarcada la investigación consistió en las Pruebas de Sistema a aplicar en el Proyecto CICPC.

Para el desarrollo de la investigación se trazó como **objetivo general** definir un manual de procedimiento para realizar Pruebas de Sistema en el Proyecto productivo Cuerpo de Investigaciones Científicas, Penales y Criminalísticas (CICPC). El cual llevó a la definición de los siguientes **objetivos específicos**:

- ✓ Seleccionar los tipos de prueba de sistema a aplicar en el Proyecto CICPC.
- ✓ Elaborar el manual de procedimiento para ejecutar cada tipo de prueba.
- ✓ Analizar los resultados obtenidos en la aplicación de las pruebas mediante Casos de Estudio y comprobar la completitud de la estrategia.

Se partió de la siguiente **pregunta científica**:

Cómo realizar las pruebas de sistema en el Proyecto Productivo Cuerpo de Investigaciones Científicas, Penales y Criminalísticas (CICPC) de manera eficiente para lograr un producto con la menor cantidad de errores posibles.

Por lo que se necesita conocer:

- ✓ ¿Qué tipos de prueba de sistema se necesitan realizar?
- ✓ ¿Cómo se realiza cada tipo de prueba?

Para darle cumplimiento a todos los objetivos propuestos con anterioridad, se llevaron a cabo una serie de **tareas** en la investigación tales como:

- Investigar sobre los diferentes tipos de pruebas de sistema a aplicar al Producto de Software.
- Seleccionar los tipos de pruebas de sistema a aplicar al Proyecto CICPC de acuerdo a los objetivos de calidad definidos.
- Definir los pasos, requisitos, plantillas y registros a utilizar en los diferentes procedimientos para la aplicación de las pruebas seleccionadas.
- Conformar el manual de procedimiento.
- Aplicar los procedimientos propuestos en diferentes Casos de Estudio.
- Analizar los resultados obtenidos en la aplicación de las pruebas
- Refinar la estrategia a partir del análisis de los resultados de su aplicación.

Entre los **Métodos de Trabajo Científico** utilizados se destacan los siguientes:

1. **Métodos generales:** El **método histórico-lógico** y el **dialéctico** para el estudio crítico de los trabajos anteriores, y para utilizar estos como punto de referencia y comparación de los resultados alcanzados.
2. **Métodos lógicos:** El método **analítico-sintético** al descomponer el problema de investigación en elementos por separado y profundizar en el estudio de cada uno de ellos, para luego sintetizarlos en la solución de la propuesta.

3. **Método Explicación:** El **método de explicación** porque se elaborará un manual para lo cuál se requiere de una explicación del por qué y el cómo se hará.

El presente trabajo consta de tres capítulos, donde el primer capítulo trata sobre la investigación teórica realizada de los aspectos más importantes que enmarcan todo lo relacionado con las pruebas de sistema, el segundo capítulo se describe la propuesta del manual de procedimiento para pruebas de sistema y en el último capítulo se aplica el manual obteniéndose resultados a partir de la aplicación a los casos de estudio implementados en el Proyecto CICPC.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción del capítulo

En este capítulo se abordarán temas como la calidad de un producto de software, sus principales conceptos y diferentes puntos de vista acerca del tema. Otro aspecto a tratar lo constituyen las pruebas que se le aplican a un producto de software para determinar su calidad, teniendo en cuenta los niveles de pruebas por los que debe pasar un producto, y tomando como centro del análisis el nivel de prueba de sistema. Existen distintos tipos de pruebas de sistema, de las que se hizo un estudio para determinar cuáles conformarían la estrategia. De esta forma se realizarían las pruebas de sistema ordenadamente, sin pérdida de tiempo y de información. Para completar el análisis es preciso ubicarse en el entorno donde se realizaron las pruebas, en este caso a las dos aplicaciones informáticas del Proyecto CICPC, portal Web y sistema integrado de información policial.

1.2 Calidad de software

Varios autores plantean diversas definiciones acerca de la Calidad de Software tales como:

La Calidad del software son todas las metodologías y herramientas y su único fin es producir software de gran calidad (Lovellette 1999), esta afirmación concluye con los puntos siguientes:

- Los requisitos del software son la base de las medidas de calidad. La falta de concordancia con los requisitos es una falta de calidad.
- Los estándares o metodologías definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la ingeniería del software. Si no se sigue ninguna metodología siempre habrá falta de calidad.
- Existen algunos requisitos implícitos o expectativas que a menudo no se mencionan, o se mencionan de forma incompleta (por ejemplo el deseo de un buen mantenimiento) que también pueden implicar una falta de calidad.

“La calidad del software es el grado con el que un sistema, componente o proceso cumple los requisitos especificados y las necesidades o expectativas del cliente o usuario”. (IEEE 1990).

“Concordancia del software producido con los requisitos explícitamente establecidos, con los estándares de desarrollo prefijados y con los requisitos implícitos no establecidos formalmente, que desea el usuario” (Pressman. 1998).

Las autoras de este trabajo consideran que la definición de Pressman es la más completa, pues la calidad está dada por la capacidad de entender, comprender y traducir las palabras del usuario en lenguaje ingenieril y cumplir con las exigencias del mismo.

Juan Carlos Granja Álvarez considera que el tema calidad ha evolucionado mucho. A finales de los años sesenta siempre que se hablaba de mejorar un software solo se pensaba en mejorar el código olvidando por completo todo lo que le rodeaba. A partir de la década de los ochenta se comienza a tomar cuenta los aspectos de especificaciones, diseño, evaluación y gestión del software. Fue entonces en 1986 que surge un modelo de madurez de la capacidad, el *Capability Maturity Model (C.M.M.)*. Luego en 1987 la *International Organization for Standardization (ISO)* constituye otro aporte. Luego publicó toda la serie ISO 9000: 9001, 9002, 9003 que indican los elementos del sistema de calidad y la 9004 que aporta indicaciones y elementos del sistema de calidad (Álvarez 2000)

No es menos cierto que en la actualidad hay que centrar la atención en la "mejora de los procesos" como forma de aumentar los niveles de calidad de los productos software (Álvarez 2000).

El profesor de la universidad de Madrid Luis Fernández considera el aseguramiento de la calidad del software un elemento esencial para el éxito de un Proyecto de desarrollo (Fernández 1995).

1.3 Pruebas de software

La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación (Software 2005).

El objetivo de las pruebas del sistema es comprobar la integración del sistema de información globalmente, verificando el funcionamiento correcto de las interfaces entre los distintos subsistemas que lo componen y con el resto de los sistemas de información con los que se comunica. En la realización de estas pruebas es importante comprobar la cobertura de los requisitos, dado que su incumplimiento puede

comprometer la aceptación del sistema por el equipo de operación responsable de realizar las pruebas de implantación del sistema, que se llevarán a cabo en el proceso Implantación y Aceptación del Sistema (Métrica 2007).

Según Kendall *“las pruebas se realizan a lo largo del desarrollo del sistema y no simplemente al final. Esto significa sacar a la luz problemas no conocidos y no demostrar la perfección de programas”* (Kendall 1997).

Aunque la realización de las pruebas es un proceso tedioso constituye una serie de pasos esenciales que ayudan a asegurar la calidad del sistema. Las pruebas se realizan por subsistemas o módulos de programa según avanza el trabajo, se hacen en diferentes niveles y a diversos intervalos. Antes de que el sistema sea puesto en producción, todos los programas deben ser probados, revisados con datos de prueba y revisados para ver si los módulos trabajan juntos entre ellos. También debe ser probado el sistema trabajando como un todo. Esto incluye probar las interfaces entre subsistemas, la corrección de la salida y la utilidad y comprensibilidad de la documentación de la salida del sistema. Los programadores, analistas, operadores y usuarios juegan papeles diferentes en los diversos aspectos de la prueba (Kendall 1997).

La realización de pruebas es una actividad en la cual un sistema o componente es ejecutado bajo condiciones o requisitos especificados, los resultados son observados y registrados, y se realiza una evaluación del sistema o componente (Software 2005).

Glem Myers establece varias normas que pueden servir adecuadamente como objetivos de la prueba (Myers 2005):

1. La prueba es un proceso de ejecución de un programa con la intención de descubrir un error.
2. Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
3. Una prueba tiene éxito si descubre un error no detectado hasta entonces.

Pressman afirma que el objetivo de la prueba es: “diseñar pruebas que saquen a la luz diferentes clases de errores con la menor cantidad de tiempo y espacio” (Pressman 1995).

Por otro lado, el colectivo de profesores del Departamento de Ingeniería de Software de la Universidad de las Ciencias Informáticas, concluye que: *“las pruebas no pueden asegurar la ausencia de defectos; solo pueden demostrar que existen defectos en el software”* (Software 2005).

Según Davis, Edward y Myers (Myers 1979; Davis 1995; Edward 1996) los principios más importantes sobre las pruebas de software son:

1. La principal dificultad del proceso de prueba es decidir cuándo parar.
2. Evitar casos de pruebas no planificados, no reutilizables y triviales a menos que el programa sea verdaderamente sencillo.
3. Una parte necesaria de un caso de prueba es la definición del resultado esperado.
4. Los casos de pruebas tienen que ser escritos no solo para condiciones de entrada válidas y esperadas sino también para condiciones no válidas e inesperadas.
5. Los casos de pruebas tienen que ser escritos para generar las condiciones de salida deseadas.
6. El número de errores sin descubrir es directamente proporcional al número de errores descubiertos.
7. Los pruebas deberían empezar por “lo pequeño” y progresar hacia “lo grande”.
8. *Con la excepción de las pruebas de unidad e integración, un programa deberá ser probado por la persona u organización que lo desarrolló.*
9. Se debe designar al programador más creativo para realizar las pruebas.

Las siguientes definiciones son algunas de las recogidas en el diccionario de la IEEE en relación a las pruebas (IEEE 1990):

Pruebas: “es una actividad en la cual un sistema o uno de sus componentes se ejecuta dos veces en circunstancias previamente especificadas, los resultados se observan, se registran y se realiza una

evaluación de algún aspecto”. Para Myers (Myers 1979), probar (o la prueba) es el “proceso de ejecutar un programa con el fin de encontrar errores”. El nombre “prueba”, además de la actividad de probar, se puede utilizar para designar “un conjunto de casos y procedimientos de prueba” (IEEE 1990).

Caso de prueba: “un conjunto de entradas, condiciones de ejecución y resultados esperados desarrollados para un objetivo particular, ejemplo: ejercitar un camino concreto de un programa o verificar el cumplimiento de un determinado requisito”. También se puede referir a la documentación en la que se describen las entradas, condiciones y salidas de un caso de prueba.

Defecto: “un defecto en el software como, por ejemplo, un proceso, una definición de datos o un paso de procesamiento incorrectos en un programa”.

Fallo: “La incapacidad de un sistema o de alguno de sus componentes para realizar las funciones requeridas dentro de los requisitos de rendimiento especificados”.

Error: tiene varias acepciones:

- ✓ La diferencia entre un valor calculado, observado o medido y el valor verdadero, especificado o teóricamente correcto. Por ejemplo, una diferencia de dos centímetros entre el valor calculado y el real.
- ✓ Un defecto. Por ejemplo, una instrucción incorrecta en un programa.
- ✓ Un resultado incorrecto. Por ejemplo, un programa ofrece como resultado de la raíz cuadrada de 36 el valor 7 en vez de 6.
- ✓ Una acción humana que conduce a un resultado incorrecto (una metedura de pata). Por ejemplo, que el operador o el programador pulse una tecla equivocada.

1.4 Tipos de pruebas

Angélica de Antonio (Antonio 1999) afirma que la prueba es aplicada para diferentes tipos de objetivos, en diferentes escenarios o niveles de trabajo. Los niveles de pruebas son los siguientes:

De acuerdo con el estándar IEEE 1012-1986 (IEEE 1986) el conjunto mínimo de pruebas que se deben realizar son: Prueba de Unidad, Prueba de Integración, Prueba de Sistema, Prueba de Aceptación y

Prueba de Regresión. Todas estas pruebas se realizan con el objetivo de comprobar el buen funcionamiento del software y verificar su calidad.

Este trabajo se enfoca en las **Pruebas de Sistema** que se realizan cuando se han integrado todos los módulos, y su objetivo es comprobar que el sistema satisface los requisitos del usuario, tanto los funcionales como los no funcionales (Salazar 2004).

1.5 Tipos de pruebas de Sistema

Las Pruebas de Sistema, se realizan después de la construcción del sistema. Las mismas prueban a fondo el sistema, comprobando su funcionalidad e integridad globalmente, en un entorno lo más parecido posible al entorno final de producción (Gutiérrez 2005).

1.5.1 Pruebas de Stress o Resistencia

Es el tipo de prueba que se enfoca en comprobar cuál es el comportamiento del sistema bajo condiciones anormales, por ejemplo de carencia de recursos de memoria, procesador, sistemas externos con los que interactúa y carga excesiva de trabajo. El objetivo de estas pruebas es identificar las partes débiles del sistema (AVATAR 2007).

1.5.2 Pruebas de Rendimiento o Carga

Está diseñada para probar el rendimiento del software en tiempo de ejecución dentro del contexto de un sistema integrado.

Es el tipo de prueba que se enfoca en comprobar los tiempos de respuesta del sistema en una cantidad limitada de escenarios de trabajo (a nivel de números de usuarios y números de transacciones), bajo una configuración de hardware y software constante (AVATAR 2007).

1.5.3 Pruebas de Seguridad

Intenta verificar que los mecanismos de protección incorporados en el sistema lo protegerán, de hecho, de acceso impropios.

Es el tipo de prueba que se encarga de certificar que los datos y las funciones del sistema solo son accesibles por los actores debidamente autorizados (AVATAR 2007).

1.5.3 Prueba de Volumen Funcional

Este tipo de prueba se dedica a certificar la capacidad del sistema de manejar volúmenes de datos extremos, acorde a lo descrito en el documento de especificaciones funcionales (AVATAR 2007).

1.5.4 Prueba de Recuperación

La prueba de recuperación consiste en crear un evento de fallas o pérdida de datos, para que los usuarios vuelvan a cargar y recuperar a partir de una copia de respaldo. Con ello, se determina si los procedimientos de recuperación, son los más adecuados para cuando el sistema falle y no se pierdan los datos (Informática 1997).

1.5.5 Prueba de Compatibilidad

Consiste en asegurar que el software sea compatible con versiones determinadas de sistemas operativos, Web Server, base de datos y demás componentes constituyentes de la arquitectura de la aplicación. Esta prueba también es conocida por otros autores como prueba de operabilidad y prueba de configuración (AVATAR 2007).

1.5.6 Prueba de Usabilidad

Se trata de pruebas efectuadas con usuarios, con el objetivo de determinar si la organización de los contenidos y las funcionalidades que se ofrecen desde el Sitio Web son entendidas y utilizadas por los usuarios de manera simple y directa. Revisan una serie de factores con el fin de establecer si cumplen con las necesidades de los usuarios del sitio. Una prueba de usabilidad es contar con unos 5 usuarios en un laboratorio, que realizarán una navegación "asistida" por la aplicación a probar. El encargado de la prueba tomará nota de qué problemas encuentran los usuarios para realizar las tareas que se les hayan indicado, y así conocer qué errores de diseño tiene la aplicación (Montero 2002).

1.5.7 Pruebas de Concurrencia o Contención

Es el tipo de prueba que se enfoca en certificar la capacidad del sistema de atender múltiples solicitudes de parte de los actores que acceden a un mismo recurso (un dato que esté almacenado en memoria, un conjunto de registros de bases de datos o una interfaz con un dispositivo de hardware o un sistema externo) (AVATAR 2007).

1.5.8 Prueba de Funcionalidad

El objetivo de esta prueba es verificar la función del sistema al fijar la tensión en la validación de las funciones, métodos, servicios y casos de usos. Para validar la aplicación debe cumplir con los siguientes parámetros (AVATAR 2007):

- ✓ Cumpla con los requisitos funcionales especificados en el diseño de la solución.
- ✓ Cumpla con los requisitos NO funcionales especificados en el diseño de la solución.
- ✓ Cumpla con las restricciones de entrada y salida de la información especificada en el diccionario de Datos.
- ✓ Cumpla íntegramente con la estructura referencial especificada en el Mapa de Navegación.

1.5.9 Prueba de Disponibilidad y Red

El objetivo de esta prueba es verificar el comportamiento de la aplicación cambiando la infraestructura de red al aplicar diferentes configuraciones, o retardos. Para validar la aplicación debe cumplir con los siguientes parámetros:

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

Es la probabilidad de que un programa funcione de acuerdo con los requisitos en un momento dado (Teruel 2007).

1.5.10 Prueba de Instalación

Esta prueba se encarga de verificar que el sistema puede ser instalado en la plataforma del cliente y funciona correctamente (Desoft S.A. 2005).

1.5.11 Prueba de Configuración

Verifica que el sistema funciona en diferentes entornos de configuración. Ejemplo: Diferentes configuraciones de red (Torres 2006).

1.5.12 Prueba de Negativas

Se encargan de intentar provocar que el sistema falle para así detectar sus debilidades. Se especifican Casos de Pruebas que intentan utilizar el sistema en formas para las que no ha sido diseñado, Ejemplo: Configuraciones de red incorrectas, capacidad de hardware insuficiente, carga de trabajo "imposible", etc. (Torres 2006).

1.5.13 Prueba de Vulnerabilidad

Es el tipo de prueba que se encarga de comprobar las vulnerabilidades del entorno de comunicaciones donde el sistema va a funcionar (la red), con el propósito de establecer los planes de contingencia adecuados y presupuestar los planes de adquisición correspondiente (AVATAR 2007).

1.5.14 Prueba de Fiabilidad de Sistema

Este tipo de prueba considera la fiabilidad de un sistema como la probabilidad de que el mismo funcione o desarrolle una cierta función, bajo condiciones fijadas y durante un período de tiempo determinado. De la misma forma se define la fiabilidad de un dispositivo, aparato o persona, como casos particulares de "sistemas" en el ámbito de la ingeniería (Wikipedia 2006).

1.6 Estrategia de prueba de software

El proceso de ejecución de Pruebas debe ser considerado durante todo el ciclo de vida de un Proyecto, para así obtener un producto de alta calidad. Su éxito dependerá del seguimiento de una Estrategia de Prueba adecuada (Anna. C Grimán 2007).

Una estrategia de prueba integra técnicas de casos de prueba en una serie de pasos planeados que llevan a la construcción exitosa del software. Describe los pasos que hay que llevar a cabo como parte de la prueba, cuándo se deben planificar y realizar estos pasos, y cuántos recursos se requieren (Pruebas 2007).

1.6.1 Descripción de la estrategia de prueba de software

Cualquier estrategia de prueba debe incorporar la planeación de las pruebas, el diseño de casos de prueba, la ejecución de pruebas y evaluación de los datos resultantes (Pruebas 2007).

Una estrategia de prueba debe ser lo suficientemente flexible para promover la creatividad y la adecuación necesaria a los sistemas de software, y lo suficientemente rígida para promover una planeación razonable y un seguimiento administrativo a medida que el proyecto progresa (Pruebas 2007).

La estrategia de pruebas debe estar dirigida a encontrar los errores más importantes con la mayor rapidez y el menor costo posible. Al involucrar más pruebas y medidas de detección, se puede definir mejor la estrategia. Las duplicidades u omisiones involuntarias que tengan lugar entre diferentes pruebas pueden ser evitadas coordinando a los probadores y las actividades de pruebas, así como fijando el alcance de la prueba (Collado 2003).

También Collado (Collado 2003) plantea una serie de deficiencias al realizar las Pruebas sin estrategia, tales como: la falta de motivación, lo cual hace que las pruebas se tornen incómodas, aburridas; también está el problema producido al probar todo al mismo tiempo, conocido como “Big-Bang”. Todo lo anterior, acompañado de los fallos que puedan ocurrir por otras partes, hacen que se vuelva más costoso corregir un error. Es más difícil diagnosticar las causas de fallos y por tanto el resultado no sería otro que un producto defectuoso.

1.6.2 Estrategia de Prueba Funcional (requisitos funcionales)

1.6.2.1 Técnicas de diseño de pruebas

Para Oscar García y Crescencio Bravo (Rubio and Santos 2007) las técnicas de Diseño de Pruebas están dadas por la imposibilidad de realizar la mayor cantidad de Prueba, pues se deben seguir determinados criterios para seleccionar los casos de prueba. Es por eso que el objetivo de las técnicas para el Diseño de Pruebas es garantizar con el mayor grado de confianza posible la detección de los defectos del software.

Según Lázaro Castillo García (Desoft S.A. 2005), Especialista de Calidad de Desoft S.A., una Colección de Casos de Pruebas, Procedimientos de Pruebas y Componentes de Pruebas conforman un Modelo de Prueba.

Los Casos de Prueba especifican una forma de probar el sistema, incluyendo los datos de entrada y resultados esperados. Un Caso de Prueba puede derivarse de un Caso de Uso del Modelo de Casos de Uso o de una realización de un Caso de Uso del Modelo de Diseño.

En la práctica lo que se prueba puede estar dado por un requisito o varios requisitos del sistema cuya implementación justifica una prueba que es posible realizar y resulta poco costosa.

Casos de Prueba (CP):

- ✓ Un CP que especifica cómo probar un Caso de Uso (CU) o un escenario específico de un CU. Este tipo de CP incluye la verificación del resultado de la interacción entre los actores y el sistema, que se satisfacen las precondiciones y poscondiciones especificadas por el CU y que se sigue la secuencia de acciones especificada en el mismo. Este tipo de CP basado en un CU funciona como una “Caja Negra”, o sea, verifica un comportamiento observable externamente (Desoft S.A. 2005).

Muchos de estos casos de pruebas pueden identificarse también como parte de los CU del sistema, dado que se corresponden casi siempre con especificaciones suplementarias o requisitos no funcionales del sistema (Desoft S.A. 2005).

Un Procedimiento de Prueba (PP) especifica cómo realizar uno o varios CP o partes de estos. Puede constituir una instrucción para un individuo de cómo realizar un CP manualmente o puede especificar cómo interaccionar manualmente con una herramienta de automatización de pruebas para crear componentes ejecutables de prueba (Desoft S.A. 2005).

Un Componente de Prueba automatiza una o varias Suite de Pruebas. Pueden ser desarrollados utilizando lenguaje de guiones o de programación, o ser grabados con una herramienta de automatización de pruebas. Se utilizan para probar los componentes en el modelo de implementación, proporcionando entradas de prueba, controlando y monitoreando la ejecución de los componentes a probar, para luego informar los resultados de las pruebas.

La Suite de Pruebas será entonces un conjunto de CP que validarán una determinada funcionalidad, módulo, o subsistema, según como sea más adecuado dividir su ejecución (Desoft S.A. 2005).

El propósito de la actividad de diseño de pruebas es:

- ✓ Identificar y describir los Casos de Pruebas
- ✓ Identificar y estructurar las Suites de Pruebas

Existen tres enfoques principales para el diseño de casos de prueba, ellos son:

1. El enfoque estructural o de caja blanca: este tipo de enfoque es para realizar las pruebas de unidad las cuales se le realizan al código del software (Pressman 2006).
2. El enfoque funcional o de caja negra.

García y Santos (Rubio and Santos 2007) afirman que en el enfoque funcional las pruebas están centradas en los requisitos funcionales del software y que para elegir bien un Caso de Prueba se debe tener en cuenta los siguiente requisitos:

- ✓ Que cubra un conjunto extenso de otros casos posibles.
- ✓ Indique algo acerca de la ausencia o la presencia de defectos en el conjunto específico de entradas que prueba, así como de otros conjuntos similares.
- ✓ Reduzca el número de otros casos necesarios para que la prueba sea razonable.
- ✓ Implice que el caso ejecute el máximo número de posibilidades de entrada diferentes para así reducir el total de casos.

La figura muestra los enfoques estructural (caja blanca) y funcional (caja negra), donde se aclara en que consiste el enfoque de caja blanca es en proporcionar que se cumpla al menos uno de los caminos básicos del módulo que se prueba y el enfoque de caja negra proporciona que se cumplan los requisitos funcionales descritos en el sistema.

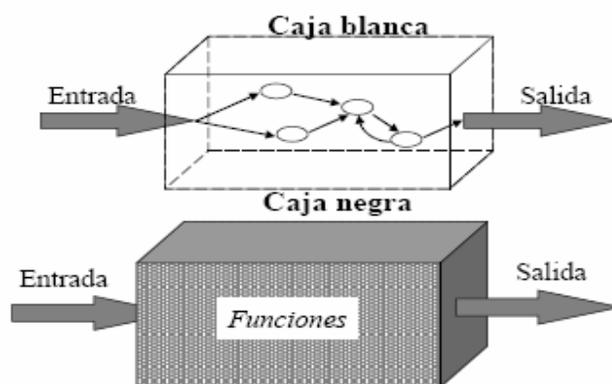


Fig. 1 Enfoque funcional y enfoque estructural

Tanto Joaquín Almunia (Almunia 2007) como Juan Antonio López Quesada (Quesada 2007) consideran que la prueba funcional o de caja negra se centra en el estudio de la especificación del software, del análisis de las funciones que debe realizar, de las entradas y de las salidas. Lamentablemente, no se pueden ejecutar todas las posibilidades de funcionamiento y todas las combinaciones de entradas y de salidas. Se deben buscar criterios que permitan elegir un subconjunto de casos cuya ejecución aporte una cierta confianza en detectar los posibles defectos del software. Como se puede observar este proceso no es sencillo, lo que convierte a las pruebas en una disciplina difícil que está demasiado lejos de parecerse a la imagen de actividad rutinaria que suele sugerir (Almunia 2007).

Para fijar estas pautas de diseño de pruebas, Myers (Myers 1979) definió qué es un caso de prueba bien elegido:

- ✓ El que reduce el número de otros casos necesarios para que la prueba sea razonable. Esto implica que el caso ejecute el máximo número de posibilidades de entrada diferentes para así reducir el total de casos.

- ✓ Cubre un conjunto extenso de otros casos posibles, es decir, indica algo acerca de la ausencia o la presencia de defectos en el conjunto específico de entradas que prueba, así como de otros conjuntos similares.

Las pruebas de caja negra se centran en lo que se espera de un módulo, es decir, intentan encontrar casos en que el módulo no se corresponde con su especificación. Por ello se denominan pruebas funcionales, y el probador se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo por dentro (Mañas 1994) .

Las pruebas de caja negra están especialmente indicadas en aquellos módulos que van a ser interfaz con el usuario (en sentido general: teclado, pantalla, ficheros, canales de comunicaciones, etc.) (Mañas 1994).

3. El enfoque aleatorio consiste en utilizar modelos (en muchas ocasiones estadísticos) que representen las posibles entradas al programa para crear a partir de ellos los casos de prueba.

1.6.2.1.1 Técnicas de caja negra

Las técnicas de prueba de caja negra existentes son:

I. Particiones o Clases de Equivalencia

Es el método que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba, cada caso debe cubrir el máximo número de entradas, debe tratarse el dominio de valores de entrada dividido en un número finito de clases de equivalencia que cumplan con las siguientes propiedades:

La prueba de un valor representativo de una clase permite suponer razonablemente que el resultado obtenido (existan defectos o no) será el mismo que el obtenido probando cualquier otro valor de la clase.

Almunia (Almunia 2007) define una serie de pasos a seguir para identificar las posibles clases de equivalencia de un programa a partir de su especificación, los mismos son:

1. Identificación de las condiciones de las entradas del programa, es decir, restricciones de formato o contenido de los datos de entrada.
2. A partir de ellas, se identifican clases de equivalencia que pueden ser:
 - ✓ De datos válidos.
 - ✓ De datos no válidos o erróneos.

La identificación de las clases se realiza basándose en el principio de igualdad de tratamiento: todos los valores de la clase deben ser tratados de la misma manera por el programa.

3. Además de Almunia (Almunia 2007), el profesor Juan Antonio López Quesada (Quesada 2007) plantean algunas de las reglas que ayudan a identificar las clases de equivalencia, ellas son:

- ✓ Si se especifica un rango de valores para los datos de entrada (por ejemplo: “el número estará comprendido entre 1 y 49”), se creará una clase válida ($1 \leq \text{número} \leq 49$) y dos clases no válidas ($\text{número} < 1$ y $\text{número} > 49$).
- ✓ Si se especifica un número de valores (por ejemplo, “se pueden registrar de uno a tres propietarios de un piso”), se creará una clase válida ($1 \leq \text{propietarios} \leq 3$) y dos no válidas ($\text{propietarios} < 1$ y $\text{propietarios} > 3$).
- ✓ Si se especifica una situación del tipo “debe ser” o booleana (por ejemplo, “el primer carácter debe ser una letra”), se identifican una clase válida (“es una letra”) y una no válida (“no es una letra”).
- ✓ Si se especifica un conjunto de valores admitidos (por ejemplo, “pueden registrarse tres tipos de inmuebles: apartamentos, casas y locales comerciales”) y se sabe que el programa trata de forma diferente cada uno de ellos, se identifica una clase válida por cada valor (en este caso son tres: apartamentos, casas y local) y una no válida (cualquier otro caso: por ejemplo, plaza de garaje).
- ✓ En cualquier caso, si se sospecha que ciertos elementos de una clase no se tratan igual que el resto de la misma, deben dividirse en clases menores.

La aplicación de estas reglas para la derivación de clases de equivalencia permite desarrollar los casos de prueba para cada elemento de datos del dominio de entrada.

4. Oscar García y Crescencio Bravo (Rubio and Santos 2007) coinciden con Almunia (Almunia 2007) en que el uso de las clases de equivalencia para identificar los casos de prueba correspondientes, está compuesto por fases, ellas son:

- ✓ Asignación de un número único a cada clase de equivalencia
- ✓ Hasta que todas las clases de equivalencia hayan sido cubiertas por (incorporadas a) casos de prueba, se tratará de escribir un caso que cubra tantas clases válidas no incorporadas como sea posible

- ✓ Hasta que todas las clases de equivalencia no válidas hayan sido cubiertas por casos de prueba, escribir un caso para una única clase no válida sin cubrir

Ejemplo de clases de equivalencia:

Entrada de un Programa:

- _ Código de Área: Número de tres cifras que no comienza ni por 0 ni por 1
- _ Nombre: Seis caracteres
- _ Orden: cheque, depósito, pago factura, retirada de fondos

Tabla 1 Ejemplo de entrada de programa

Condición de entrada	Clases Válidas	Clases Inválidas
Código área	$200 \leq \text{código} < 999$	Código < 200
Nombre para identificar la operación	Seis caracteres	Menos de 6 caracteres Más de 6 caracteres
Orden	Cheque Depósito Pago factura Retirada de fondos	Ninguna orden válida

II. Análisis de Valores Límite

Pressman (Pressman 2006) plantea que no hay una identificación clara de los valores límite, afirma que suele haber más errores en los valores límites que en los típicos.

Cuando se habla de Valores Límites lo que se dice es que se eligen casos de prueba en los límites o bordes de la clase que se está probando. (Ej.: Si una condición de entrada o salida exige un rango entre B y R, se realizan los casos de prueba para los valores B y R, y además a los valores que están por encima de ellos, es decir A y S).

La técnica de valores límites complementa a la técnica de Partición Equivalente.

Las diferencias entre ambas técnicas fueron definidas por Oscar García y Crescencio Bravo (Rubio and Santos 2007):

_ En lugar de seleccionar «cualquier» elemento como representativo de una clase de equivalencia, se requiere la selección de uno o más elementos tal que los márgenes se sometan a prueba.

_ Más que concentrarse únicamente en el dominio de entrada (condiciones de entrada), los casos de prueba se generan considerando también el espacio de salida.

III. Conjetura de Errores

Se enumera una lista de posibles equivocaciones típicas que pueden cometer los desarrolladores y de situaciones propensas a ciertos errores:

- ✓ El valor cero es una situación propensa a error tanto en la salida como en la entrada.
- ✓ En situaciones en las que se introduce un número variable de valores, conviene centrarse en el caso de no introducir ningún valor y en el de un solo valor. También puede ser interesante una lista que tiene todos los valores iguales.
- ✓ Es recomendable imaginar que el programador pudiera haber interpretado algo mal en la especificación.
- ✓ También interesa imaginar lo que el usuario puede introducir como entrada a un programa.

1.6.3 Estrategia de Prueba no Funcional (requisitos no funcionales)

Ana Grimán afirma que una vez establecidas los pasos a seguir durante el proceso de prueba, se procede a asociar las técnicas de pruebas de software que garantizarán los Requisitos No Funcionales (RNF). Para identificar los RNF se partió del Modelo de Calidad del Sistema. Este modelo permite identificar las características de calidad que deben ser evaluadas en un software. Estas características tienen a su vez subcaracterísticas asociadas. Se tomaron en cuenta las siguientes características: **Fiabilidad, Usabilidad, Eficiencia, Mantenibilidad y Portabilidad** (Anna. C Grimán 2007).

A continuación una breve relación de las características, subcaracterísticas y atributos de calidad de la ISO 9126 conformada por la Dra. Coral Calero (Calero 2003):

Estos elementos de calidad planteados por la ISO 9126 son los que definen cuán baja puede ser la calidad de un software determinado. Basados en datos estadísticos a través de las métricas de calidad se logra una visión real de cómo está la calidad del producto, ya sea Bajo, Medio o Alto.

Tabla 2 Características, subcaracterísticas y atributos de calidad según la ISO 9126

Características	Subcaracterísticas	Atributos
Funcionalidad	Compleitud	<ul style="list-style-type: none"> - Configuración total (existencia de todos los programas y ficheros del software) - Contenido total (existencia de todos los requerimientos funcionales)
	Consistencia	<ul style="list-style-type: none"> - Automatización (empleo de herramientas de software para el desarrollo) - Uniformidad de la estructura, del contenido y del formato de los elementos componentes del software (uniformidad de las pantallas, menús, ayudas, reportes, mensajes, opciones) - Uniformidad del vocabulario, de la simbología y de otras convenciones utilizadas - Uniformidad de retorno al procesamiento (restablecimiento de pantallas, menús, ayudas)

	Corrección	<ul style="list-style-type: none"> -Utilización correcta del idioma. -Correspondencia de las descripciones con los objetos (nivel de ayuda solicitado, mensajes de error según el fallo detectado, etc.) -Funcionamiento correcto (ausencia de errores críticos del software, errores de ejecución, ciclos infinitos, resultados incorrectos, interrupción de la ejecución o salidas abruptas)
	Integridad	<ul style="list-style-type: none"> -Auditabilidad (instrumentación de controles de la auditoria informática) - Seguridad (definición de controles de acceso adecuado al software, subsistema, función, fichero de datos, etc.) - Auto chequeo (diseño sobre la autoprotección contra virus informáticos)
	Normalización	<ul style="list-style-type: none"> - Normalización de la codificación - Normalización del vocabulario - Normalización de la simbología -Normalización del aseguramiento de la calidad -Normalización la actividad objeto de investigación.
Confiabilidad	Exactitud	<ul style="list-style-type: none"> - Exactitud de los cálculos
	Recuperabilidad	<ul style="list-style-type: none"> - Opciones de recuperabilidad (pérdida o deterioro de datos y

		elementos componentes del software, errores del operador, etc.)
	Tolerancia de errores o fallos	<ul style="list-style-type: none"> - Verificación de la memoria interna y externa (para la instalación del producto, la reconfiguración del producto o para la salva de la información) - Validación previa de condiciones potenciales de errores (particiones de trabajo inexistentes, disponibilidad e integridad de los ficheros, disponibilidad de los periféricos) - Tratamiento de los errores (procedimientos para la detección y corrección de errores internos del software) Procesamiento degradado (procedimientos para el funcionamiento degradado en caso de fallos no recuperables como la ausencia de ficheros o deficiencias del hardware)
Usabilidad	Comprensibilidad	<ul style="list-style-type: none"> - Terminología (utilización de la terminología acorde a la actividad objeto de investigación o de acuerdo a la actividad del usuario) - Ayuda en línea (presencia de diferentes niveles de ayuda: sistema, pantalla, campo, etc.) - Interfaz usuario adecuada

		(representación de los objetos con análogos en el ambiente del usuario, utilización de iconos, etc.)
	Facilidad de entrenamiento	<ul style="list-style-type: none"> - Existencia de un DEMO para el producto de software - Existencia de un Tutorial para el producto de software
	Operabilidad	<ul style="list-style-type: none"> - Utilidad de las ayudas (utilidad de la información que brinda la ayuda) - Operabilidad de la ayudas (facilidad de desplazamiento en la ventana de ayuda) -Carga automática (generación durante la instalación del fichero de comandos para la ejecución del producto de software) - Minimización del trabajo extra del usuario (durante la operación y en caso de fallos) - Facilidades de operación opcional (selección de opciones mediante el cursor de barras, video invertido, brillantez de las opciones, Mouse, etc.)
Eficiencia	Comportamiento del tiempo	- Eficiencia en el tiempo de ejecución y de respuesta requeridos
	Comportamiento de los recursos	- Eficiencia en la utilización de los recursos (memoria RAM, utilización de compactadores)
Mantenibilidad	Auto documentación	-Documentación interna (comentarios de los programas y nombres

		descriptivos adecuados)
	Claridad	- Claridad de los mensajes - Claridad de los nombres de objetos (programas, pantallas, módulos, funciones, etc.)
	Concisión	- Programación procedimental (buenas prácticas de codificación)
	Diagnosticabilidad	- Localización del fallo o error (información suficiente para localizar y corregir la causa del error)
Portabilidad	Adaptabilidad	- Independencia del ambiente de hardware (características particulares del hardware) - Independencia del ambiente de software (sistema operativo, lenguaje de programación)
	Instalabilidad	- Existencia de un módulo de instalación (instalación automática del software) - Existencia de un módulo de configuración (configuración automática del software)
Reusabilidad	Generalidad	- Tipificación de funciones
	Modularidad	- Independencia funcional (la ejecución de cada elemento del software debe ser independiente) - Diseño modular

Como bien se muestra en la figura estas son las etapas del proceso de prueba indicando que en todas las etapas se realizan pruebas a través de técnicas. En la etapa de pruebas de sistema está dividido por

enfoques el **enfoque estructural** (Caja Blanca) se lleva a cabo a través de técnicas tales como: prueba de camino básico, pruebas bucles, el **enfoque funcional** (Caja Negra) se lleva a cabo a través de las técnicas Particiones o Clases de Equivalencia, análisis de valores límites y conjeturas de errores y el **enfoque aleatorio** se lleva a cabo a través de modelos (en muchas ocasiones estadísticos) que representen las posibles entradas al programa para crear a partir de ellos los casos de prueba. En la figura solo se muestran los enfoques funcional y el estructural, el aleatorio no, porque es poco usado, los que más se utilizan son los representados.

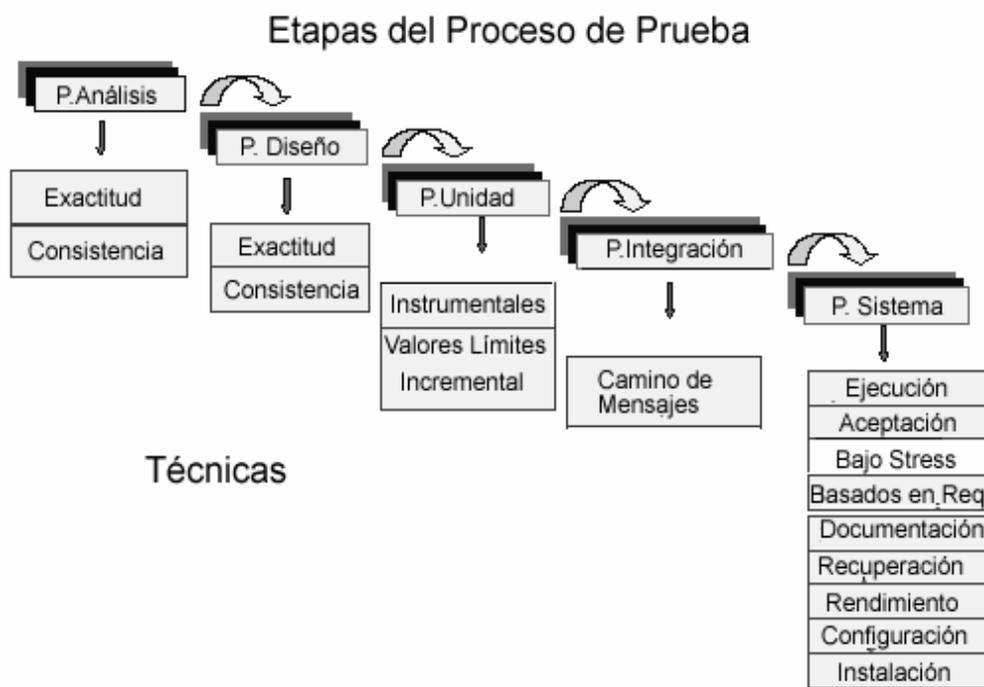


Fig. 2 Técnicas asociadas a las Etapas del Proceso de Prueba.

1.7 Proceso en que se realizan las pruebas de sistema a un producto de software siguiendo RUP.

El **Proceso Unificado de Rational** o RUP (Rational Unified Process), es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos (Wikipedia 2007).

El RUP es un producto de Rational (IBM). Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso) (Wikipedia 2007).

El RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al terminar cada ciclo. Cada ciclo se divide en fases que finalizan con un hito donde se debe tomar una decisión importante (Wikipedia 2007), las fases son: Inicio, Elaboración, Construcción y Transición.

Fase de Construcción

La fase de Construcción por definición tiene como finalidad culminar el análisis y diseño de los requisitos que hayan quedado pendientes de la fase de elaboración y completar el desarrollo del sistema siguiendo la arquitectura del software ya definida (RUP 2003). Los requisitos que se puedan quedar pendientes de analizar y diseñar deben ser excepcionales, debe haber surgido como adicional al alcance original y no deben tener impacto alguno en la arquitectura. Los objetivos son: implementar las versiones del sistema con un equilibrio entre calidad y rapidez, así como también minimizar los costos de desarrollo optimizando el uso de recursos y previniendo retrasos, re-trabajo, desperdicio (AVATAR 2007).

Fase de Transición

La fase de Transición se centra en implantar el producto en su entorno de operación. Ejemplo, si un producto va a salir al mercado, el equipo de Proyecto distribuye una versión beta usuarios típicos pertenecientes a organizaciones “clientes beta” que sean representativas. Si un producto va a distribuirse a un único cliente (o quizás a una serie de usuarios en una organización grande), el equipo instala el producto en un solo sitio (Ivar Jacobson 2004).

Jacobson, Booch y Rumbaugh (Ivar Jacobson 2004) afirman que: *“La fase de transición finaliza con el lanzamiento del producto”*.

La fase de Transición tiene como finalidad asegurar que el software esté disponible para sus usuarios finales (RUP 2003). Los objetivos de esta fase es probar que el sistema cumpla con los requisitos funcionales y técnicas requeridas, y cumplir con la lista de verificación que da por culminado el producto.

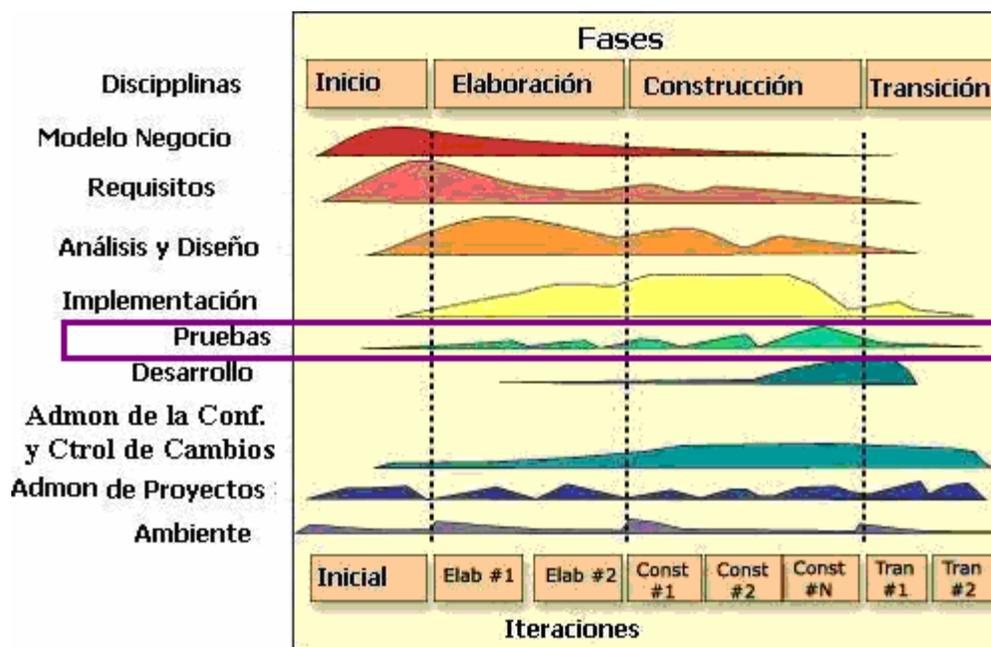


Fig. 3 Ciclo de Vida de RUP

En cada fase se realizan varias iteraciones. Dentro de cada iteración el desarrollo transita por diferentes disciplinas. Una disciplina es una colección de actividades relacionadas con un área de atención dentro de todo el Proyecto. El grupo de actividades que se encuentran dentro de una disciplina principalmente son una ayuda para entender el Proyecto desde la perspectiva clásica de cascada.

Las disciplinas que propone RUP son:

- ✓ Modelado de Negocios
- ✓ Requisitos
- ✓ Análisis y Diseño
- ✓ Implementación
- ✓ Pruebas, Transición
- ✓ Configuración y Administración del Cambio
- ✓ Administración de Proyectos y Ambiente.

Disciplina “Pruebas”

Esta disciplina actúa como un proveedor de servicios a las otras disciplinas en muchos aspectos. Pruebas se enfoca principalmente en la evaluación y aseguramiento de la calidad del producto, desarrollado a través de las siguientes prácticas:

- ✓ Encontrar fallas de calidad en el software y documentarlas.
- ✓ Recomendar sobre la calidad percibida en el software.
- ✓ Validar y probar las suposiciones hechas durante el diseño y la especificación de requerimientos de forma concreta.
- ✓ Validar que el software trabaja como fue diseñado.
- ✓ Validar que los requisitos son implementados apropiadamente.

Las autoras de este trabajo se centraron solamente en la metodología RUP (Rational Unified Process), porque ha sido la metodología seleccionada para el desarrollo del Proyecto CICPC. Dicha metodología propone que el sistema se realice de una forma iterativa e incremental, teniendo en cuenta los casos de uso para dirigir el proceso de implementación de las funcionalidades, y una arquitectura base bien definida previa al desarrollo, además el desarrollo iterativo permite ser adaptable a las condiciones del cliente y a realizarle los cambios que necesite, lo cual no quiere decir, que el proceso se realice de manera incontrolada y desorganizada, sino que se necesita un equilibrio entre la estabilización de un conjunto de requisitos y la realidad de los requisitos cambiantes.

1.8 Descripción del software objeto de prueba

El proyecto CICPC consiste en el desarrollo de dos aplicaciones informáticas, un sistema integrado de información policial y un portal Web. Ambos están concebidos para el personal del CICPC, subdelegaciones estatales. En los siguientes epígrafes se describe cada una de estas aplicaciones por separado.

1.8.1 Sistema integrado de información policial

En el Informe Técnico del Software (López 2006) la solución que se propone es un nuevo sistema integrado de información policial, tomando como referencia el sistema existente SIIPOL, de forma tal que

las funciones actuales sean mantenidas, incrementando las posibilidades de uso de la información, y agregando nuevas funciones que el actual sistema no concibe.

El sistema centralizará, en varios módulos o subsistemas, la automatización de los procesos que se llevan a cabo en las dependencias que pertenecen al Cuerpo de Investigaciones Científicas, Penales y Criminalísticas (CICPC). La organización y cantidad de subsistemas estará en la medida de las necesidades, por lo que no deben coincidir exactamente con los que existen actualmente en el SIIPOL; en principio los subsistema deberán organizarse por los procesos, y no por áreas de trabajo, es decir, por ejemplo: existirá un único subsistema para la toma de la denuncia, independientemente de que existan varias áreas que lleven a cabo este proceso con sus correspondientes particularidades; y un único subsistema para la sustanciación del expediente investigativo independientemente del tipo de delito investigado y el área que lleve a cabo la investigación (López 2006).

A continuación se enuncian algunos de los subsistemas que puede tener la aplicación, en función del estudio preliminar realizado para determinar el alcance (López 2006):

- ✓ Subsistema de denuncia e investigación policial.
- ✓ Subsistema de seguimiento de información policial.
- ✓ Subsistema de apoyo a la investigación.
- ✓ Subsistema de control de funcionarios.
- ✓ Subsistema de INTERPOL
- ✓ Subsistema de asuntos internos.
- ✓ Subsistema de reportes y estadísticas.
- ✓ Subsistema de administración.
- ✓ Subsistemas utilitarios.

El sistema deberá cumplir con las siguientes prestaciones generales (López 2006):

- ✓ Acceso “en línea” desde todas las dependencias que pertenezcan al CICPC, a un servidor central en un ambiente cliente/servidor vía Web.

- ✓ Funcionalidades para que las 4 coordinaciones nacionales, las 24 delegaciones, las 88 subdelegaciones, la supervisión de delegaciones estatales, la inspectoría general, la asesoría jurídica, y la policía internacional; puedan recopilar y controlar la información operativa que se genere como parte de sus procesos de trabajo en el sistema.
- ✓ Almacenamiento seguro para toda la información que se ingrese, y tratamiento centralizado para la obtención de reportes estadísticos confiables para la toma de decisiones.
- ✓ Entre 2 y 30 usuarios como promedio por dependencia del CICPC operando el sistema y demandando servicios del servidor central.
- ✓ El tiempo de respuesta no debe sobrepasar al que permite el sistema policial actual; y la estabilidad en cuanto a la información almacenada también debe mantenerse.
- ✓ La Base Datos (BD) inicial deberá incluir el almacenamiento de la información de carácter policial que tiene el actual sistema policial para que el sistema pueda ofrecer información operativa desde el principio.

Al proponer como solución una Aplicación Web se presupone algunas características que son indispensables para estos tipos de soluciones como (López 2006):

- ✓ Acceso al sistema a través de un navegador Web, en cualquier sistema operativo que se instale en la PC cliente.
- ✓ Un servidor de Bases de Datos y un servidor de Aplicación que trabajen en conjunto para permitir todas las funcionalidades que puede brindar el sistema.
- ✓ Una red de comunicación constante entre todas las PC clientes y el servidor de Aplicaciones.
- ✓ Comunicación directa entre el servidor de Aplicaciones y el servidor de BD.

También se pueden definir otras características que son fundamentales para que el sistema pueda darle respuesta a los procesos antes mencionados en el alcance; estos requisitos adicionales serán clasificados de acuerdo al objetivo que persiguen (López 2006).

1.8.2 Portal Web

También en el Informe Técnico del Software se concibe el Portal Web como parte del desarrollo del sistema de gestión, considerándolo como un subsistema especial, ya que presentará ciertas características diferentes, incluso hasta en la tecnología de desarrollo (López 2006).

Una parte importante dentro de la estrategia comunicacional es precisamente el desarrollo de un Portal Web que acompañe al software de gestión, a través de servicios adicionales que estén orientados al uso de la población en general, tanto venezolana como extranjera. El portal Web no es simplemente una herramienta informática, sino un producto comunicacional sumamente poderoso que puede ser utilizado para democratizar la información, buscando los siguientes objetivos:

- ✓ Reflejar la imagen e identidad del Cuerpo de Investigaciones Científicas, Penales y Criminalísticas de la República de Venezuela.
- ✓ Mejorar los niveles de información que actualmente las personas tienen de las labores que se realizan en el CICPC, tanto policiales como de relaciones inter-institucionales.
- ✓ Abrir una nueva forma de comunicación con la población, para escuchar sus inquietudes, sus sugerencias, y poder sondear de esta forma el estado de opinión con respecto al trabajo de la institución.
- ✓ Realizar trabajo preventivo mediante la publicación de consejos, y descripción de Modus Operandis, que permitan a la población enfrentarse preventivamente al delito.
- ✓ Mejorar la asesoría a la población en materia de orientación general.

El Portal estará orientado a brindar los siguientes servicios:

- ✓ Información general institucional, localización geográfica y de contactos.
- ✓ Información relacionada a la historia del CICPC (galería de directores, cambios más importantes realizados desde la creación de la institución, etc.)
- ✓ Información de orientación general como: ¿cuales son las vías para ingresar al CICPC?, ¿donde debe dirigirse una persona cuando es víctima de un delito?
- ✓ Publicación de boletines y noticias de diferentes temas, por ejemplo:

- Delitos importantes (modus operandis)
 - Actividades y relaciones institucionales entre organismos del ministerio
 - Logros en la investigación criminal
- ✓ Recepción de quejas y sugerencias de la población.
 - ✓ Respuestas a las preguntas más frecuentes realizadas a la institución.
 - ✓ Publicación de información no restringida que almacene el sistema policial, como el estado de una investigación, el listado de vehículos u objetos recuperados, de las personas desaparecidas, de las personas buscadas por el CICPC, algunos resultados estadísticos, etc.
 - ✓ Publicación de encuestas de opinión y cuestionarios (Pruebas).
 - ✓ Información de procedimientos, normas, reglamentos y disposiciones de carácter general e instructivo sobre las investigaciones y la sustanciación penal, por las que deben regirse todos los funcionarios.

Para brindar los servicios antes mencionados, el portal deberá tener algunas características funcionales como:

- ✓ Empleo de editores WYSWYG para la edición rápida y sencilla de noticias.
- ✓ Establecimiento de políticas de publicación sobre tiempo de actualidad de las noticias y canales de permisos para la publicación.
- ✓ Historial de noticias publicadas en el sitio.
- ✓ Interfaz de comunicación en línea con el sistema de gestión policial.
- ✓ Publicación de información en imágenes, y otros formatos de información más complejos (Word, Excel, .PDP).

1.9 Conclusiones del capítulo

Después de haber hecho el análisis bibliográfico de la investigación se pueden arribar a las siguientes conclusiones acerca de la calidad del software, siendo esta el grado con el que un proceso cumple los

requisitos especificados y las necesidades o expectativas del cliente o usuario. En el mundo de la calidad es muy amplio el tema de las pruebas ya que es una forma idónea de medir calidad a un sistema. Es decir, la realización de pruebas es una actividad en la cual un sistema es ejecutado bajo ciertas condiciones, los resultados son observados y registrados, y se realiza una evaluación del sistema. Existen un conjunto de pruebas divididas por niveles: pruebas unitarias, pruebas de integración, pruebas de sistema y pruebas de aceptación, este trabajo se enfocó en el estudio de las pruebas de sistema tales como: las pruebas de stress o resistencia, prueba de rendimiento o carga, prueba de seguridad, prueba de instalación, prueba de concurrencia, prueba de vulnerabilidad, prueba de red, entre otras que se aplican a un software. Para ejecutar estas pruebas es necesario elaborar una estrategia de prueba que ayude y oriente a la realización de las pruebas de sistema. Cualquier estrategia de prueba debe incorporar la planificación de las pruebas, el diseño de casos de prueba, la ejecución de pruebas y evaluación de los datos resultantes. Dado el nivel de complejidad, seguridad, etc. que deben tener las aplicaciones informáticas a desarrollar en el proyecto es necesario prestar total atención a las pruebas en cada uno de sus niveles y sus procesos.

CAPÍTULO 2: MANUAL DE PROCEDIMIENTO

2.1 Introducción del capítulo

Las autoras de este trabajo, acordaron realizar un manual de procedimiento o estrategia de instrucciones precisas, desde la planificación, diseño, ejecución y evaluación de las pruebas para realizar las mismas y así detectar los errores existentes. Los temas abordados en este capítulo son referentes a la realización de una estrategia de pruebas con todos los casos de pruebas para cada una de las metas propuestas por cada tipo de pruebas de sistema que se le realizarán a un producto de software del Proyecto CICPC. Primeramente se define la estrategia, para poder obtener todos los casos de pruebas necesarios para la ejecución satisfactoria de las pruebas al software. Este capítulo muestra la esencia del manual definido para el Proyecto CICPC, el cual ayuda y orienta durante el proceso de ejecución de las pruebas.

2.2 Ciclo completo de las pruebas

A pesar de que la propuesta de Félix Oscar García Rubio y Crescencio Bravo Santos es eficiente, se propone hacer uso de este ciclo completo para la realización de las pruebas, está compuesto por elementos fundamentales, tales como: la planificación de las pruebas que para su elaboración es preciso tener conocimiento acerca del Proyecto y del software que se esté desarrollando para poder elaborar el Plan de Pruebas, que ayudaría a Diseñar los Casos y Procedimientos de Pruebas, una vez que se tiene todo listo se procede a la Ejecución de las Pruebas y se obtienen los Resultados, que en dependencia de estos resultados se sabrá si la prueba tuvo éxito o no, y entonces se procede al análisis de los errores encontrados (Rubio and Santos 2007).

A continuación se muestra un diagrama de actividades que describe cada una de las acciones que realizan los responsables de las pruebas, tanto el probador, el analista de pruebas como el diseñador de las pruebas. El analista de las pruebas es quien se encarga de buscar la información del proyecto y del software, luego selecciona las pruebas que se le aplicarán al software y se las envía al diseñador de las pruebas que tiene como función diseñar los casos y procedimientos de pruebas, una vez diseñadas las pruebas se las envía al probador que es el que ejecuta las pruebas, obtiene los resultados esperados y los inesperados también y se lo informa al analista que luego realiza un análisis de los resultados obtenidos de las pruebas aplicadas al sistema. Las actividades más importantes son las que se resaltan en el

diagrama: determinar las pruebas a realizar, diseñar las pruebas, diseñar los casos y procedimientos de pruebas, ejecutar las pruebas, obtener los resultados y analizar los errores.

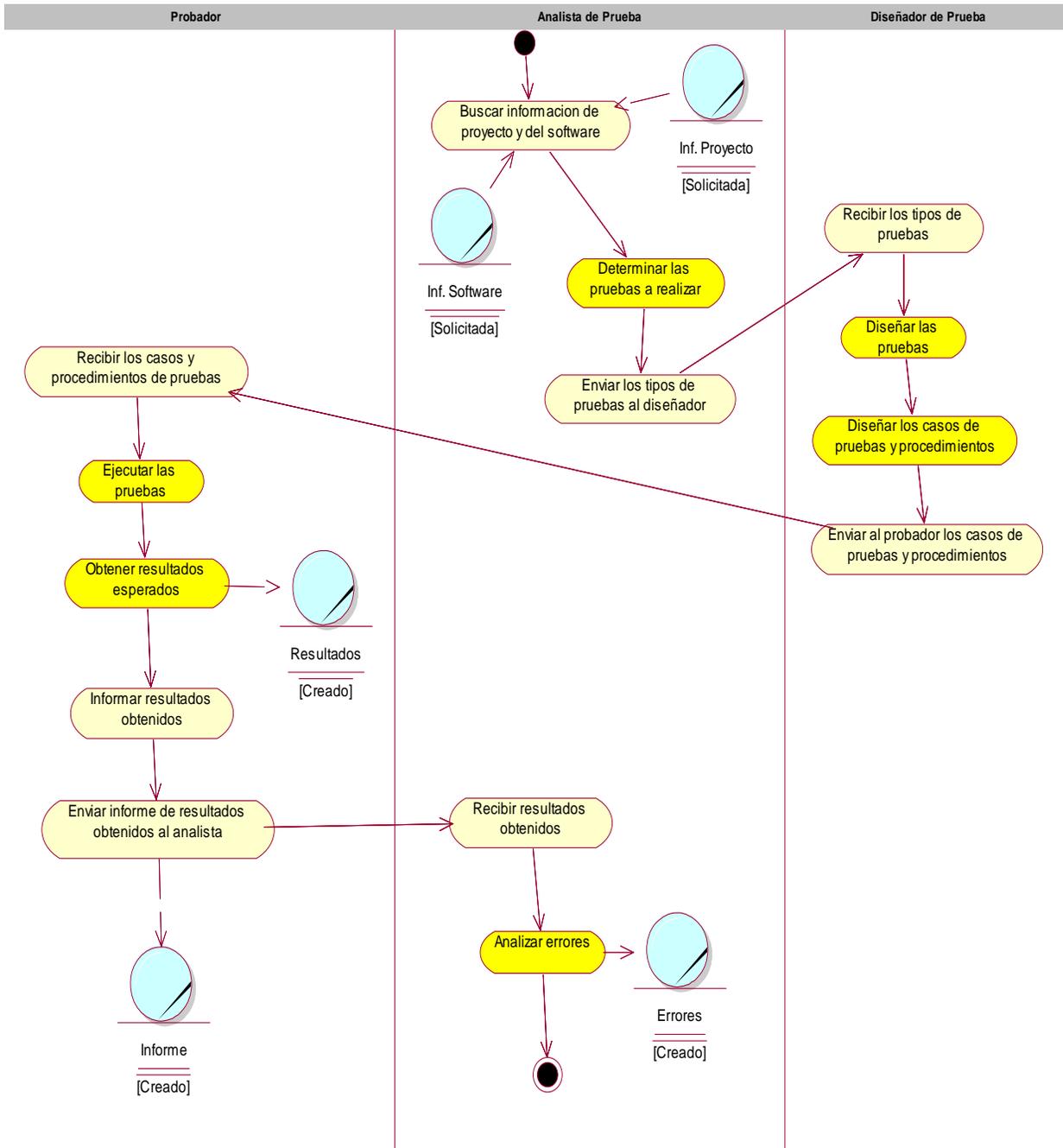


Fig. 4 Ciclo completo de las pruebas

2.3 Estrategia de Pruebas de Sistema.

La Estrategia de Pruebas de Software propuesta está compuesta por las siguientes actividades:

- Planificación de las pruebas.
- Diseño de las pruebas.
- Ejecución de las pruebas.
- Evaluación de las pruebas.

Los responsables de realizar cada una de estas actividades en el Proyecto CICPC son los siguientes:

- **Analista de Prueba:** Es responsable de identificar y definir las pruebas requeridas, monitoreando el proceso de prueba y el resultado de en cada ciclo de prueba.
- **Diseñador de pruebas:** Es responsable de definir le método de prueba y asegurar su implementación exitosa.
- **Probador:** Es responsable de desarrollar las pruebas, lo que incluye ejecutar y registrar los resultados de la prueba.

2.3.1 Planificación de las pruebas.

La planificación de las pruebas se realiza con antelación para lograr organización durante su ejecución, para evitar pérdida de tiempo, confusiones, que los probadores se aburran, y no le den la seriedad que lleva la actividad, para poner en práctica todos los pasos, es preciso realizar las pruebas sin perder un solo detalle del proceso.

Los encargados de realizar el Diseño de los Casos de Pruebas son los analistas de sistema, porque ellos son los que realizan la captura de requisitos y tienen un amplio dominio de cómo debe funcionar el sistema. Luego los analistas de sistema deben entregar estos documentos a los analistas y diseñadores de las pruebas en la etapa de preparación de las mismas.

Preparación de las Pruebas de Sistema

En esta etapa se preparan todos los recursos necesarios para realizar las pruebas de sistema, de acuerdo a las características del entorno establecidas en el plan de pruebas. Para ello se asegura la disponibilidad del entorno y de los datos necesarios para ejecutar estas pruebas.

Artefactos

De entrada

- Plan de Pruebas del Proyecto CICPC.

De salida

- Entorno de Pruebas de Sistema para el Proyecto CICPC.

Participantes

- Equipo de Calidad y el líder del Proyecto CICPC.
- Analista y Diseñadores de Pruebas del Proyecto CICPC

Definir los tipos de pruebas que se le aplicarán al software, explicando bien sus objetivos y metas que se persigue para el Proyecto CICPC.

Las autoras de este trabajo proponen el empleo de estas pruebas seleccionadas para aplicar en el Proyecto CICPC según sus características, pues contamos con la experiencia de haber sido probadas en los Casos de Estudios implementados y arrojaron resultados satisfactorios sacando a la luz un gran número de errores que demostraron la eficiencia de haber aplicado estas pruebas y no otras.

Por tanto, es aconsejable que en la primera iteración del proceso de pruebas de sistema se realicen este grupo de pruebas: Funcionalidad, Seguridad, Disponibilidad y Red, Rendimiento o Carga, Compatibilidad, Resistencia o Stress, Usabilidad y Fiabilidad.

1. Prueba de Funcionalidad

Objetivo

Verificar la función del sistema al fijar la tensión en la validación de las funciones, métodos, servicios y casos de usos.

Metas

Validar que la aplicación:

- ✓ Cumpla con los requisitos funcionales especificados en el diseño de la solución por medio de Casos de Uso
- ✓ Cumpla con los requisitos No funcionales especificados en el diseño de la solución.
- ✓ Cumpla con las restricciones de entrada y salida de la información especificada en el diccionario de Datos
- ✓ Cumpla íntegramente con la estructura referencial especificada en el Mapa de Navegación.

2. Prueba de Seguridad.

Objetivo

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

Metas

Validar que en la aplicación:

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

3. Prueba de Disponibilidad y Red.

Objetivo

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

Metas

Validar que en la aplicación:

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

4. Prueba de Rendimiento o Carga

Objetivo

Verificar el rendimiento del software en tiempo de ejecución dentro del contexto de un sistema integrado y así como la verificación de la capacidad del sistema para manejar volúmenes de datos extremos de acuerdo a la velocidad que se especifique para el sistema.

Metas

Validar en la aplicación:

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

5. Prueba de Compatibilidad

Objetivo

Verificar el funcionamiento del sistema sobre diferentes componentes de software.

Metas

Validar en la aplicación:

- ✓ Sistemas Operativos.
- ✓ Navegadores.

6. Prueba de Resistencia o Stress

Objetivo

Verificar cómo se comporta el sistema bajo condiciones anormales:

Metas

Validar en la aplicación:

- ✓ Carencia de sistemas externos con los que interactúa el sistema.
- ✓ Aplicar carga excesiva de trabajo al sistema (extremas sobrecarga)

- ✓ Hardware no disponible.
- ✓ Recursos compartidos no disponibles.
- ✓ Además verifica qué hace el sistema cuando el usuario no hace lo que supuestamente debe hacer.

7. Prueba de Usabilidad

Objetivo

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

Metas

Validar en la aplicación:

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

8. Prueba de Fiabilidad

Objetivo

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

Meta

Validar en la aplicación:

- ✓ El sistema deberá estar disponible las 24 horas.
- ✓ El tiempo permitido que el sistema puede estar fuera de operación después de un fallo no debe exceder 2 días.

Se escogieron estas pruebas conjuntamente con la líder del Proyecto CICPC antes de comenzar la captura de requisitos, ya que la líder era la que más conocimiento tenía sobre las características y detalles que pidió el cliente para el software. Es decir, en lo que consistía ambas aplicaciones informáticas: el portal Web y el sistema integrado de información policial. Por lo que ya teniendo estos datos la líder

recomendó que las pruebas propuestas en el manual, no debieran faltar independientemente de que se le agregaran otras una vez que se terminara la captura de requisitos.

Procedimientos para preparar el entorno para realizar las pruebas a los Casos de Estudios del Proyecto CICPC:

Realizar una reunión con todo el Equipo de Calidad dirigida por el líder de rol de Calidad del proyecto para organizar las pruebas a los Casos de Estudios, llegando a los siguientes acuerdos:

1. Debe contar con la cooperación de todos los miembros del equipo de calidad para cumplir con esta tarea.
2. Distribuir los tipos de pruebas a cada uno de los miembros del equipo de calidad con el objetivo de que cada cual dirija y oriente a los probadores encargados de realizar un tipo de prueba en específico:

- ✓ Prueba Funcional
- ✓ Prueba de Disponibilidad y Red
- ✓ Prueba de Rendimiento o Carga
- ✓ Prueba de Compatibilidad
- ✓ Prueba de Stress o Resistencia
- ✓ Prueba de Seguridad
- ✓ Prueba de Usabilidad
- ✓ Prueba de Fiabilidad

Se escogen estas pruebas para aplicársela a los Casos de Estudios porque con estas se lograría medir toda la funcionalidad y además se probaría la estrategia propuesta para luego aplicarla al proyecto CICPC con estas pruebas.

3. Instalar la herramienta VMWare en un laboratorio cuyas computadoras cumplan con los siguientes requisitos mínimos.

- ✓ Más de 768 MB de memoria RAM.
- ✓ Más de 10 GB de espacio libre en disco duro.
- ✓ Conexión LAN 100mps.

4. Verificar que todas las máquinas cumplen con todo los requisitos establecidos en el punto 3.
5. Los administradores deben enviar un reporte con el estado de las máquinas.
6. Decidir cuántas máquinas se necesitan para cada tipo de prueba.

7. Hacer la distribución de las máquinas por cada tipo de prueba.
8. Si se necesitan más máquinas solicitar al líder del proyecto que gestione otro laboratorio o decidir utilizar diferentes horarios para realizar las pruebas en la misma cantidad de máquinas.
9. Solicitar al líder del proyecto que cite la cantidad de probadores necesarios según cronograma u horario definido.
10. Circular la plantilla de No Conformidades.
11. Para la organización, hay que especificar que el responsable de cada tipo de prueba se reúne 30 minutos antes de comenzar para explicar el procedimiento, los objetivos, la forma de llenar la plantilla, etc. Que además debe estar constantemente observando a los probadores y estar disponible para cualquier situación que se pueda presentar.
12. Que además debe estar presente el responsable del subsistema que se está probando.
13. Tener la localización de la aplicación – url de cada Caso de Estudio.
14. Contar con los probadores necesarios para realizar todas las pruebas de sistema propuestas a la aplicación.
15. Definir con la líder del Proyecto CICPC la fecha propuesta de las pruebas.
16. Publicar en el servidor del Proyecto la máquina virtual ya configurada, luego copiarla en todas la máquinas del laboratorio.
17. Verificar que estén copiada la máquina virtual 2 horas antes de comenzar las pruebas en todos las PC del laboratorio.
18. Comenzar a preparar el entorno del laboratorio una hora antes. Además es necesario realizar los siguientes:
 - ✓ Reparar el IP de las máquinas virtuales para que no de conflicto IP y tome el IP por asignación del DHCP del laboratorio donde se prepara para realizar las pruebas.
 - ✓ Ir a Opciones de Internet/Conexiones/Configuración de LAN/Opciones avanzadas/y poner a continuación *.uci.cu;10.*.*
19. Instalar en la máquina virtual de las PCs donde se harán las pruebas de compatibilidad los siguientes programas:
 - ✓ Navegadores
 - Mozilla
 - Netscape

- Opera
- Explorer
- ✓ Sistemas Operativos
 - XP
 - Windows 2003 Server
 - Linux

Es necesario verificar que los navegadores estén configurados correctamente para evitar que no funcione.

Las autoras de este trabajo comparten el criterio de Oscar García y Crescencio Bravo al considerar más adecuada la técnica de Particiones o Clases de Equivalencia para el diseño de casos de prueba para esta estrategia.

2.3.2 Diseño de las pruebas de Sistema

Las pruebas de sistema se dividen en dos tipos: las pruebas funcionales y las pruebas no funcionales.

Las pruebas de funcionales se encargan de verificar la funcionalidad del sistema basado en la especificación de casos de usos o requisitos funcionales.

Las pruebas no funcionales son las encargadas de verificar el rendimiento, el stress, configuración, seguridad, etc. del sistema, los cuales conforman los requisitos no funcionales específicos que debe cumplir una aplicación determinada.

Las pruebas funcionales requieren el uso y la aplicación de una técnica de caja negra, siguiendo esta técnica se confeccionan los casos de pruebas para probar las condiciones de entrada tanto válidas como inválidas.

Las pruebas no funcionales no es de obligatorio cumplimiento el uso del diseño de casos de prueba que contenga condiciones de entrada para clases válidas y clases inválidas, pues esto depende de la aparición o no de la descripción de uno o varios requisitos no funcionales y a la hora de efectuar las pruebas de este tipo sea preciso adicionarlas para verificar la calidad del sistema

El diseño se realizará siguiendo dos estrategias, una Estrategia de Prueba Funcional y una Estrategia de Prueba No Funcional, las cuales se detallan en las siguientes secciones.

2.3.2.1 Estrategia de Prueba Funcional (requisitos funcionales)

El diseño de las pruebas se debe describir para cada uno de los tipos de pruebas de sistema con los siguientes datos:

Nombre de la Prueba: *[Nombre de la prueba que se aplicará ejemplo (stress, etc.)]*

Todo estrategia de prueba debe llevar el nombre de la prueba que se le aplicó para poder identificar el tipo de error encontrado a la hora de recopilar en la tabla de no conformidades y así saber hacia donde se debe enfocar el trabajo para remediar el error.

Tipo de Prueba: *[Pruebas de Sistema]*

El tipo de prueba se refiere en este caso al nivel de prueba, es decir, si es prueba de unidad, si es prueba de integración, prueba de sistema o si es prueba de aceptación.

Entrada: *[Poner las condicion4s de entradas según los casos de usos del sistema]*

En la entrada se explica el cómo se introducirán los datos, que aparecen en el diccionario de datos, los cuales son necesarios solamente para las pruebas de funcionalidad del sistema, es decir, para confección de los casos de pruebas que ayudarán a verificar la funcionalidad de la aplicación en cuestión, aunque para las pruebas no funcionales pueden hacerse casos de pruebas todo depende de la especificación de los requisitos no funcionales del sistema.

Tabla 3 Descripción de las clases válidas y clases inválidas

Condición de Entrada	Clases válidas	Clases inválidas	Resultado Obtenido	Resultado de la prueba	Observaciones
<i>[Es la condición especificada en el diccionario de datos del CU]</i>	<i>[Será la que cumpla con la especificación de un rango de valores para los datos de entrada] [Datos válidos]</i>	<i>[Será la que no cumpla con la especificación de un rango de valores para los datos de entrada] [Datos no válidos o erróneos]</i>	<i>[El resultado que devolvió el sistema como respuesta según la entrada]</i>	<i>[Satisfactorio o insatisfactorio]</i>	<i>[Cualquier sugerencia]</i>

Técnica: *[Explicación de cómo obtener el caso de prueba]*

Del método de caja negra la técnica de particiones de equivalencia es la utilizada para diseñar los casos de prueba que se describen con el formato siguiente:

Tabla 4 Descripción del Caso de Prueba

Tipo de Prueba	<i>[Nombre del tipo de prueba de sistema: funcional]</i>
Nombre del Caso de Uso (1...n)	<i>[Nombre del CU que se probará]</i>
Descripción:	<i>[Describir brevemente los pasos a seguir para llegar al CU sometido a prueba]</i>
Nombre del Caso de Prueba (1....n)	<i>[Nombre del CP y especificar el escenario donde se encuentra]</i> <i>Ejemplo: El nombre del caso de prueba está dado según la cantidad de flujos tanto básicos como alternos. Ejemplo CPR 1. Registrar consulta médica.</i>
Condición de Entrada:	<i>[Descripción textual de lo que ocurra en el mundo real que hace necesario ejecutar el caso de prueba, precisando los datos de entrada y los comandos a dar por el actor. Descripción textual del estado de la información almacenada.]</i>
Salida Esperada:	<i>[Descripción textual del estado en que queda la información y las alertas que pueden generarse, una vez ejecutado el CU con los valores y el estado especificado en la entrada.]</i>

Si se especifica un número de valores (por ejemplo, “se pueden registrar de uno a tres propietarios de un piso”), se creará una clase válida ($1 \leq \text{propietarios} \leq 3$) y dos no válidas ($\text{propietarios} < 1$ y $\text{propietarios} > 3$).

Si se especifica una situación del tipo “debe ser” o booleana (por ejemplo, “el primer carácter debe ser una letra”), se identifican una clase válida (“es una letra”) y una no válida (“no es una letra”).

Si se especifica un conjunto de valores admitidos (por ejemplo, “pueden registrarse tres tipos de inmuebles: pisos, casa y locales comerciales”) y se sabe que el programa trata de forma diferente cada uno de ellos, se identifica una clase válida por cada valor (en este caso son tres: piso, casa y local) y una no válida (cualquier otro caso: por ejemplo, plaza de garaje).

Especificación

- ✓ Código área: número de 3 dígitos que no empieza por 0 ni por 1.
- ✓ Nombre de identificación: 6 caracteres.
- ✓ Órdenes posibles: “cheque”, “depósito”, “pago factura”, “retirada de fondos”.

Aplicación de las reglas

- Código
 - Número, regla 3, booleana:
 - Clase válida (número)
 - Clase no válida (no es número)
 - Regla 5, la clase número debe subdividirse; por la regla 1, rango, obtenemos:
 - Subclase válida (200 código 999)
 - Dos subclases no válidas (código < 200 ; código > 999)
- Nombre de ID., número específico de valores, regla 2:
 - Clase válida (6 caracteres)
 - Dos clases no válidas (más de 6; menos de 6 caracteres)
- Orden, conjunto de valores, regla 4:
 - Una clase válida para cada orden (“cheque”, “depósito”...); 4 en total.
 - Una clase no válida (“divisas”, por ejemplo).

A continuación se han enumerado las clases identificadas y la generación de casos (presuponiendo que el orden de entrada es: código, nombre, orden) se ofrece a continuación.

Tabla 6 Ejemplo para identificar clases y la generación de casos de pruebas

Condición de entrada	Clases válidas	Clases inválidas
Código área	(1) $200 \leq \text{código} \leq 999$	(2) código <200 (3) código >999 (4) no es número
Nombre para identificar la operación	(5) seis caracteres	(6) menos de 6 caracteres (7) más de 6 caracteres
Orden	(8) «cheque» (9) «depósito» (10) «pago factura» (11) «retirada fondos»	(12) ninguna orden válida

Ejemplo de clases de equivalencia.

Casos de prueba válidos:

300 Nómina Depósito (1) (5) (9)
 400 Viajes Cheque (1) (5) (8)
 500 Coches Pago-factura (1) (5) (10)
 600 Comida Retirada-fondos (1) (5) (11)

Casos de prueba no válidos:

180 Viajes Pago-factura (2) (5) (10)
 1032 Nómina Depósito (3) (5) (9)
 XY Compra Retirada-fondos (4) (5) (11)
 350 A Depósito (1) (6) (9)
 450 Regalos Cheque (1) (7) (8)
 550 Casa &%4 (1) (5) (12)

Procedimientos: *[Cómo proceder en el momento de la prueba]*

Paso 1. Montar el entorno del laboratorio

Paso 2. Montar las herramientas necesarias para la aplicación de la prueba.

Paso 3. Distribuir cantidad de probadores por tipo de Prueba de Sistema a realizar al software.

Paso 4. Una vez montado el entorno de prueba con todos los probadores listos y preparados para realizar la prueba correspondiente en cualquier navegador y plataforma, se procede a la ejecución de la prueba en cuestión.

Paso 5. Recopilar en la tabla de no conformidades todos los errores encontrados, para luego corregirlos.

Salida o Resultado: *[Plantilla para recoger los resultados]*

Tabla 7 Descripción de la tabla de No Conformidades para pruebas funcionales

Tipo de Prueba	<i>[Nombre de la prueba]</i>
Nombre del Probador	<i>[Nombre del probador]</i>
Nº	<i>[Número de la no conformidad]</i>
Elementos	<i>[Tipo de elemento, ejemplo: interfaz]</i>
CU	<i>[Nombre del CU correspondiente a la no conformidad encontrada]</i>
No conformidad	<i>[Describir la no conformidad encontrada]</i>
Localización de la no Conformidad	<i>[El escenario, sección o pantalla del CU asociado, donde se encontró]</i>
Sugerencias	<i>[Comentario que desee poner sobre la no conformidad encontrada]</i>
Atributos asociados	<i>[Especificar la subcaracterística asociada tanto para las no conformidades, como para las sugerencias, por separado. Ejemplo: Completitud (no conformidad), Corrección (sugerencia)]</i>

2.3.2.2 Estrategia de Prueba no Funcional (requisitos no funcionales)

Según Lázaro Luis Castillo García (Castillo 2007), Especialista de Calidad de Desoft S.A. afirma que no es de obligatorio cumplimiento la elaboración de casos de pruebas para requisitos no funcionales, ya que los requisitos no funcionales de una aplicación determinada, valida el rendimiento, el stress, la seguridad, la disponibilidad y red, la fiabilidad, compatibilidad, entre otras que no requieren de una condición de entrada específica sino de explicar el procedimiento del cómo se va a llevar a cabo en la aplicación.

Al igual que puede suceder que en la especificación de los requisitos no funcionales para la aplicación no aparezca la descripción de uno o varios requisitos no funcionales y a la hora de efectuar las pruebas de este tipo sea preciso adicionarlas para verificar la calidad del sistema, aunque en ocasiones si el cliente desea medir alguna condición de entrada específica comprendido entre estos requisitos no funcionales se pueden elaborar casos de pruebas tales como el ejemplo que se muestra a continuación para la prueba de seguridad:

Tabla 8 Ejemplo de Casos de Prueba para la Prueba de Seguridad

Condición de Entrada	Clases válidas	Clases inválidas	Resultado Obtenido	Resultado de la prueba	Observaciones
Contraseña	Un usuario solo puede entrar por medio de su contraseña	Cualquier usuario puede entrar a la plataforma sin necesidad de autenticarse.			
	El usuario deberá registrarse con su contraseña predefinida, al entrar a una página de alta seguridad. Deberá comprobarse que la única forma de acceder a esa página sea por el método de autenticación.	Un usuario cualquiera pueda entrar a la zona restringida copiando la dirección generada a un usuario ya registrado y poniéndola en su barra de direcciones.			

	Un usuario ya autenticado deberá tener la opción de moverse por las páginas sin necesitar una nueva autenticación.	Un usuario autenticado sale de la zona de seguridad por medio del botón atrás no podrá entrar en la misma pagina de seguridad por medio del botón siguiente.			
--	--	--	--	--	--

El diseño de las pruebas se debe describir para cada uno de los tipos de pruebas de sistema con los siguientes datos:

Nombre de la Prueba: *[Nombre de la prueba que se aplicará ejemplo (stress, etc.)]*

Toda estrategia de prueba debe llevar el nombre de la prueba que se le aplicó para poder identificar el tipo de error encontrado a la hora de recopilar en la tabla de no conformidades y así saber hacia donde se debe enfocar el trabajo para remediar el error.

Tipo de Prueba: *[Pruebas de Sistema]*

El tipo de prueba se refiere en este caso al nivel de prueba, es decir, si es prueba de unidad, si es prueba de integración, prueba de sistema o si es prueba de aceptación.

Entrada: *[“Deber ser” para hacer cada prueba] Poner diccionario de datos*

En la entrada se explica el cómo se introducirán los datos, que aparecen en el diccionario de datos, los cuales son necesarios solamente para la realización de las pruebas de funcionalidad del sistema, es decir, para confección de los casos de pruebas que ayudarán a verificar la funcionalidad de la aplicación en cuestión, aunque para las pruebas no funcionales pueden hacerse casos de pruebas todo depende de la especificación de los requisitos no funcionales del sistema.

Procedimientos: *[Cómo proceder en el momento de la prueba]*

Paso 1. Montar el entorno del laboratorio

Paso 2. Montar las herramientas necesarias para la aplicación de la prueba.

Paso 3. Distribuir cantidad de probadores por tipo de Prueba de Sistema a realizar al software.

Ejemplo: Tres probadores para realizar pruebas funcionales, cuatro probadores para realizar pruebas de carga, etc.

Paso 4. Una vez montado el entorno de prueba con todos los probadores listos y preparados para realizar la prueba correspondiente en cualquier navegador y plataforma, se procede a la ejecución de la prueba en cuestión.

Paso 5. Recopilar en la tabla de no conformidades todos los errores encontrados, para luego corregirlos.

Salida o Resultado: *[Plantillas para recoger los resultados]*

Tabla 9 Descripción de de la tabla de No Conformidades para Pruebas no Funcionales

Tipo de Prueba	<i>[Nombre de la prueba]</i>
Nombre del Probador	<i>[Nombre del probador]</i>
Nº	<i>[Número de la no conformidad]</i>
Elementos	<i>[Tipo de elemento, ejemplo: interfaz]</i>
Nombre del Requisito No Funcional	<i>[Nombre del requisito no funcional correspondiente a la no conformidad encontrada]</i>
No conformidad	<i>[Describir la no conformidad encontrada]</i>
Localización de la no Conformidad	<i>[El escenario, sección o pantalla del CU asociado, donde se encontró]</i>
Sugerencias	<i>[Comentario que desee poner sobre la no conformidad encontrada]</i>
Atributos asociados	<i>[Especificar la subcaracterística asociada tanto para las no conformidades, como para las sugerencias, por separado. Ejemplo: Completitud (no conformidad), Corrección (sugerencia)]</i>

2.3.3 Ejecución de las Pruebas del Sistema

El objetivo de esta etapa es comprobar la integración de todos los subsistemas y componentes del sistema de información, así como la interacción del mismo con otros sistemas de información con los que se relaciona, de acuerdo a las verificaciones establecidas para el nivel de pruebas del sistema.

Artefactos

De entrada

- ✓ Producto Software
- ✓ Entorno de Pruebas del Sistema
- ✓ Plan de Pruebas

De salida

- ✓ Resultado de las Pruebas del Sistema

Prácticas

- ✓ Pruebas del Sistema

Participantes

- ✓ Probadores

Para realizar la ejecución de las pruebas de sistema con la calidad requerida es preciso:

- ✓ Los analistas de sistema confeccionen los casos de pruebas que se le entregan a cada probador para verificar la funcionalidad de la aplicación.
- ✓ Reunirse con los probadores para explicarle cada uno de los pasos de la descripción de los casos de pruebas y como se procede a la hora de llenar las plantillas de No Conformidades acordadas por las responsables de calidad para recopilar la información y los resultados referentes a cada caso de prueba correspondiente.
- ✓ Informarle a cada probador que tipo de prueba le corresponde realizar y las informaciones necesarias para cumplir con la tarea.

2.3.4 Evaluación del resultado de las pruebas de Sistema

El objetivo de esta actividad es analizar los resultados de las pruebas del sistema de información y efectuar su evaluación. Dicha evaluación recoge el grado de cumplimiento de las mismas, y consiste en:

- ✓ Comparar los resultados obtenidos con los esperados.
- ✓ Identificar el origen de cada problema detectado para poder remitirlo al responsable, determinar la índole de las modificaciones y qué acciones deben llevarse a cabo para resolverlo de forma satisfactoria e inmediata.
- ✓ Indicar si el plan de pruebas debe volver a realizarse total o parcialmente, y si será necesario contemplar nuevos casos de prueba no considerados anteriormente.

Artefactos

De entrada

- ✓ Resultados de las pruebas del Sistema de los Casos de Estudio del Proyecto CICPC.
- ✓ Plan de Pruebas del Proyecto CICPC.

De salida

- ✓ Evaluación del Resultado de las Pruebas del Sistema de los Casos de Estudio del Proyecto CICPC.

Participantes

- ✓ Analistas y Diseñadores de Pruebas del Proyecto CICPC.
- ✓ Jefe de Proyecto CICPC.
- ✓ Jefes de Subsistemas.
- ✓ Responsables de las pruebas.
- ✓ Responsables de Calidad.

2.4 Conclusiones del capítulo

La definición de un manual de procedimiento y desarrollo de una estrategia de pruebas de sistema para el Proyecto CICPC es de vital importancia, pues sin esta no se contaría con la organización necesaria para lograr un efectivo desempeño durante la ejecución de dichas pruebas con la calidad requerida. Contar con un procedimientos determinado y conciso que guíe secuencialmente para llevar a cabo pruebas de sistema que sean eficientes y capaces de recoger la mayor cantidad de errores y/o fallos que tenga una

aplicación, es de total cumplimiento en un Proyecto cómo es el Proyecto CICPC, poniendo fin a los errores que puedan afectarlo y disgustar al cliente.

La estrategia cuenta con los diferentes tipos de pruebas que miden los objetivos y metas fundamentales; además de los casos de pruebas para efectuar las pruebas y se puedan obtener resultados satisfactorios con mejor precisión., garantizando la calidad del producto del Proyecto CICPC.

CAPÍTULO 3: EVALUACIÓN DE LOS RESULTADOS

3.1 Introducción del capítulo

En este capítulo se mostrarán los resultados obtenidos de las pruebas de sistema realizadas a dos Casos de Estudio implementados por los desarrolladores del Proyecto CICPC. Estos Casos de Estudio se desarrollaron para entrenar a los desarrolladores del mismo y para el buen funcionamiento interno del Proyecto, los Casos de Estudios son: Ficha Técnica y Soporte Técnico.

El Caso de Estudio Ficha Técnica surge como solución a una problemática real del Proyecto CICPC, dada la necesidad de controlar la vida del estudiante, dentro y fuera del mismo. Es una aplicación Web implementada sobre la Plataforma J2EE, y con una Ingeniería desarrollada de acuerdo a la Metodología establecida por RUP. La aplicación consta de un módulo Administrativo, en el cual se controla todo lo referente a la seguridad y accesibilidad del sistema y varios módulos de Gestión en los que se implementaron las funcionalidades referentes al desempeño del estudiante, dígase Desempeño Investigativo, Docente, Cursos Optativos, etc., además de tener registrado los datos personales de los estudiantes. El sistema está implementado para que el usuario final del producto, sea quien maneje la aplicación desde el punto de vista administrable, o sea, los estudiantes de los cuales se tienen registrados los datos, solo podrán acceder a la aplicación para ver su información y a la misma solo tendrán el permiso de solo lectura, no podrán ejercer ningún cambio.

El Caso de Estudio Soporte Técnico es para atender de forma rápida todas las peticiones hechas por el cliente, teniendo solo acceso a revisar estas peticiones los administradores del Proyecto CICPC.

Prestaciones de Soporte Técnico

1. Revisar cada solicitud y darle respuesta a las mismas lo antes posible.
2. Mantener todas las actualizaciones de software necesarios en el Proyecto.
3. Los administradores pueden cuantificar los medios de los laboratorios del Proyecto, y saber el estado actual de cada uno de estos medios.

3.1.1 Organización y planificación de las pruebas

Para validar la estrategia propuesta se llevaron a cabo las pruebas de sistema en el transcurso de una semana, dichas pruebas fueron: prueba funcional, prueba de stress, prueba de seguridad, prueba de

compatibilidad, prueba de fiabilidad y prueba disponibilidad y red. Para ello tuvimos una reunión con la segunda líder del Proyecto una semana antes de la fecha de realización de las pruebas para planificar la cantidad total de probadores que se necesitaban para efectuar las pruebas, la presencia de los administradores de sistema para instalar la herramienta VMWare versión 5.5.1 para montar las máquinas virtuales con los navegadores y sistemas operativos necesarios para ejecutar las pruebas, con un mismo usuario y contraseña de forma tal que cuando entraran los probadores al laboratorio tuvieran las máquinas virtuales preparadas y con la aplicación lista para comenzar a probar, también se necesitaron los administradores de sistema para que garantizaran 10 GB de capacidad libres en la partición (E/D) en cada una de las máquinas del laboratorio. Además se acordó usar solamente un laboratorio ya que para ejecutar las máquinas virtuales se necesita como mínimo 764MB de RAM.

Al finalizar las pruebas se recopiló toda la información obtenida y se graficaron estos resultados que aparecen a continuación:

3.2 Resultados obtenidos del caso de estudio de Ficha Técnica del Proyecto de CICPC

Las pruebas de sistema realizadas a este Caso de Estudio se ejecutaron el 7 de mayo de 2007.

Para la realización de las pruebas de sistema de dicho caso de estudio se contó con 19 probadores, 11 probadores para las pruebas funcionales, 2 para las pruebas de compatibilidad, 3 para las pruebas de seguridad, 1 para la prueba de stress, 2 para la prueba de disponibilidad y red, en la primera hora se realizaron las pruebas funcionales, las de seguridad y disponibilidad y red y en la segunda hora se realizaron las pruebas de stress y compatibilidad.

La selección de los mismos fue hecha por la segunda líder del Proyecto CICPC de acuerdo a la cantidad de probadores que se necesitaban por cada tipo de prueba, y la selección de alguno de ellos estuvo dada según las habilidades que tenían para desarrollar dicha actividad.

Las pruebas se realizaron en dos horas, a las 4:45 PM comenzaron las pruebas y a las 6:45 PM terminaron, se necesitó media hora para preparar el entorno de laboratorio.

Todas las orientaciones pertinentes para esta actividad fueron distribuidas por la segunda líder del Proyecto a los probadores. Las responsables por el equipo de calidad para realizar las pruebas de sistema fueron Yuniet Toll y Yilennis Mendoza que planificaron la distribución de cada una de las pruebas y la cantidad de probadores a cada una de las integrantes del equipo de calidad del Proyecto CICPC.

3.2.1 Resultados obtenidos

3.2.1.1 Pruebas Funcionales

Tabla 10 Resultados de las Pruebas de Sistema al Caso de Estudio Ficha Técnica

Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	José Ernesto Lara Rodríguez
Nº	1
Elementos	Interfaz
CU	Administrar Asignaturas
No conformidad	El botón Cancelar no hace nada...
Localización de la no Conformidad	Pantalla "Eliminar Asignatura"
Sugerencias	Quitar el botón, no cumple ningún objetivo, o ponerle un vínculo a una pantalla apropiada
Atributos asociados	Complejidad (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	María de los Ángeles García M.
Nº	1
Elementos	Interfaz
CU	Administrar Actividades
No conformidad	El botón "Cancelar" lo que hace es dar un paso atrás, o sea, si inserto una actividad determinada y borro un carácter, cuando cancelo, deshace la operación, o sea, inserta el carácter nuevamente y si vuelvo a dar cancelar, no borra la actividad que escribí. En ningún escenario se hace lo que está establecido.
Localización de la no Conformidad	"Cancela el envío de los datos"
Sugerencias	Que cuando oprima el botón cancelar, borre la actividad que escribí.
Atributos asociados	Complejidad (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	María de los Ángeles García M.
Nº	2
Elementos	Interfaz
CU	Administrar Actividades
No conformidad	Reunión Proyecto, Reunión Proyecto Si el que registra una actividad tiene problemas con las tildes, y

	escribe lo que está en negrita y oprime aceptar, el sistema no muestra un mensaje diciendo que esa actividad existe y la inserta
Localización de la no Conformidad	“ Datos incorrectos”
Sugerencias	
Atributos asociados	Compleitud (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	María de los Ángeles García M.
Nº	3
Elementos	Interfaz
CU	Administrar Actividades
No conformidad	
Localización de la no Conformidad	“Modificar datos de una actividad”
Sugerencias	Que muestre un mensaje indicando el cambio. Ver Figura 1 y 2
Atributos asociados	Compleitud (sugerencia)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Isis Margarita Blanco Zamora
Nº	1
Elementos	Interfaz
CU	Gestionar Ayudantía
No conformidad	El caso de uso “Gestionar ayudantía” no existe por ese nombre (En su lugar aparece “Gestionar Información Individual”, y de ahí se puede acceder a “Registrar Ayudantía” o “Modificar Ayudantía”).
Localización de la no Conformidad	Una vez registrado el usuario, se accede a “Administración y Gestión”, de ahí a “Gestionar Información Individual”, y desde esa pantalla se puede acceder a “Registrar Ayudantía” o “Modificar Ayudantía”, que se despliega en el menú “Ayudantía”
Sugerencias	Debe ser equivalente el nombre del caso de prueba (Y del caso de uso) con el nombre de las pantallas que aparecen realmente en la interfaz
Atributos asociados	Compleitud (no conformidad) Corrección (no conformidad) Consistencia (sugerencia) Consistencia (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Isis Margarita Blanco Zamora

Nº	2
Elementos	Interfaz
CU	Gestionar Ayudantía
No conformidad	No me permite introducir un nuevo estudiante: no están activos los campos para entrar sus datos.
Localización de la no Conformidad	Una vez registrado el usuario, se accede a “Administración y Gestión”, de ahí a “Gestionar Información Individual”, y desde esa pantalla se accede a “Modificar Ayudantía”, que se despliega en el menú “Ayudantía”.
Sugerencias	Debe darse la posibilidad de introducir los datos para un nuevo estudiante.
Atributos asociados	Compleitud (sugerencia) Compleitud (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Isis Margarita Blanco Zamora
Nº	3
Elementos	Interfaz
CU	Gestionar Ayudantía
No conformidad	El sistema muestra un error, cuando se va a “Modificar la ayudantía individual”, y no se ha escogido el estudiante.
Localización de la no Conformidad	Una vez registrado el usuario, se accede a “Administración y Gestión”, de ahí a “Gestionar Información Individual”, y desde esa pantalla se accede a “Modificar Ayudantía”, que se despliega en el menú “Ayudantía”. Una vez en esa pantalla se llenan los datos correspondientes.
Sugerencias	El sistema debería mostrar un mensaje para que el usuario se percate de que necesita especificar el estudiante al cual va a modificar la ayudantía.
Atributos asociados	Compleitud (no conformidad) Compleitud (sugerencia)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Liudmila Sánchez Almenares
Nº	1
Elementos	Interfaz
CU	Administrar Cursos Optativos
No conformidad	Al pasar el cursor sobre algún acceso se despliega más de un campo y se queda desplegado aunque no se haga ninguna función con él. Ver Figura 3
Localización de la no	Administrar Cursos Optativos

Conformidad	
Sugerencias	
Atributos asociados	Compleitud (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Maylen Cepero Madruga.
Nº	1
Elementos	Interfaz
CU	Administrar Desempeño Investigativo
No conformidad	Cuando voy a Registrar, Modificar y Eliminar y lo hace correctamente no le notifica al usuario que ya se realizó la operación.
Localización de la no Conformidad	Registrar, Modificar y Eliminar Desempeño Investigativo.
Sugerencias	Que cuando se realice cualquiera de estas operaciones se le notifique al usuario que ya se realizó la actividad satisfactoriamente.
Atributos asociados	Compleitud (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Maylen Cepero Madruga.
Nº	2
Elementos	Interfaz
CU	Administrar Ficha Personal
No conformidad	Que cuando entro a uno de los casos de usos no tengo forma de ir al otro directamente
Localización de la no Conformidad	En las páginas de los casos de usos
Sugerencias	
Atributos asociados	Compleitud (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Liana Toledo Bueno
Nº	1
Elementos	Interfaz
CU	Registrar Nueva Asignatura
No conformidad	Admite nombres con dígitos ejemplo: Mat6 Esto sucede al final del nombre ejemplo: 6MA_t, Ma6_t , con estos ejemplos también se detectan errores. Ver Figura 1
Localización de la no Conformidad	Registrar Datos de una Asignatura
Sugerencias	

Atributos asociados	Complejidad (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Liana Toledo Bueno
Nº	2
Elementos	Interfaz
CU	Registrar Nueva Asignatura
No conformidad	
Localización de la no Conformidad	
Sugerencias	<p>Cuando se vaya a la opción Cancelar, se deben eliminar los mensajes que aparecen debajo reportando los errores:</p> <ul style="list-style-type: none"> -Problemas de integridad de datos - Error de la entrada.
Atributos asociados	Complejidad (sugerencia)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Liana Toledo Bueno
Nº	3
Elementos	Interfaz
CU	Registrar Nueva Asignatura
No conformidad	<p>Cuando tratamos de Insertar una Asignatura con errores ya existente muestra error, y cuando damos muestra Cancelar con otra entrada, sale por defecto el anterior.</p> <p>Ver figura 2 y 3</p>
Localización de la no Conformidad	
Sugerencias	
Atributos asociados	Complejidad (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Liana Toledo Bueno
Nº	4
Elementos	Interfaz
CU	Registrar Nueva Asignatura
No conformidad	
Localización de la no Conformidad	
Sugerencias	Que los sitios tengan un mapa de navegación para mayor entendimiento del usuario.

Atributos asociados	Completitud (sugerencia)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Liana Toledo Bueno
Nº	5
Elementos	Interfaz
CU	Registrar Nueva Asignatura
No conformidad	
Localización de la no Conformidad	
Sugerencias	Que los encargados de preparar las Pruebas, preparen a los probadores con vista a trabajar con el Documento de Prueba
Atributos asociados	Normalización (sugerencia)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Liana Toledo Bueno
Nº	6
Elementos	Interfaz
CU	Registrar Nueva Asignatura
No conformidad	Cuando listamos las asignaturas sale el paginado un número variable de Asignaturas. No se muestra una asignatura salen 26 cuando el listado total es 27. Ver figura 4,5 y 6
Localización de la no Conformidad	
Sugerencias	
Atributos asociados	Completitud (no conformidad) Consistencia (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Liana Toledo Bueno
Nº	7
Elementos	Interfaz
CU	Registrar Nueva Asignatura
No conformidad	
Localización de la no Conformidad	
Sugerencias	Que cuando el usuario cometa un error el sistema le muestre un mensaje notificándole el error cometido.
Atributos asociados	Consistencia (sugerencia)

Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Liana Toledo Bueno
Nº	8
Elementos	Interfaz
CU	Registrar Nueva Asignatura
No conformidad	Salí de la aplicación porque se había bloqueado (demasiado lento) Ahora entré por user: admin. ; pass: admin. No me deja entrar.
Localización de la no Conformidad	
Sugerencias	
Atributos asociados	Consistencia (sugerencia) Complejidad (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Yinimary Ortega Montoya
Nº	1
Elementos	Interfaz
CU	Administrar Desempeño Laboral
No conformidad	No encontré ninguna inconformidad.
Localización de la no Conformidad	
Sugerencias	
Atributos asociados	Normalización (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Yordankis Matos López
Nº	1
Elementos	Interfaz
CU	Gestionar Cursos Optativos
No conformidad	Cuando se quiere iniciar el caso de uso no hay ningún lugar que diga gestionar cursos optativos. Lo que se muestra es en la parte izquierda gestionar información individual y gestionar información colectiva todo separado
Localización de la no Conformidad	
Sugerencias	Pienso que debería decir en la parte izquierda gestionar información y donde dice cursos optativos que se muestren entonces la opciones: "Registrar Curso Optativo Individual" "Registrar Curso Optativo Colectivo"

	“Modificar Datos de Curso Optativo Individual” “Modificar Datos de Curso Optativo Colectivo”.
Atributos asociados	Consistencia (sugerencia) Compleitud (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Yordankis Matos López
Nº	2
Elementos	Interfaz
CU	Gestionar Cursos Optativos
No conformidad	En ocasiones el mensaje que se muestra de error no te dice el por qué del error
Localización de la no Conformidad	Gestionar información individual, cursos optativos, registrar cursos optativos
Sugerencias	
Atributos asociados	Compleitud (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Yordankis Matos López
Nº	3
Elementos	Interfaz
CU	Gestionar Cursos Optativos Colectivo
No conformidad	En este caso de uso el campo “Tutor” admite nombres con símbolos extraños
Localización de la no Conformidad	Gestionar información colectiva, cursos optativos, registrar cursos optativos
Sugerencias	
Atributos asociados	Compleitud (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Yudanis Gago Martínez
Nº	1
Elementos	Interfaz
CU	Registrar Desempeño Investigativo Individual
No conformidad	Muestra los nombres de los estudiantes que para registrarle algún desempeño hay que seleccionarlo y eso no se sabe
Localización de la no Conformidad	Gestionar Desempeño Investigativo
Sugerencias	Debería ser tener algún sistema de búsqueda de nombres y que se

	viera como un campo a llenar para el registro de un desempeño.
Atributos asociados	Consistencia (sugerencia) Compleitud (no conformidad) Compleitud (sugerencia)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Yudanis Gago Martínez
Nº	2
Elementos	Interfaz
CU	Modificar Desempeño Investigativo Individual
No conformidad	Al seleccionar un estudiante para realizar una modificación de una actividad investigativa y seleccionar la actividad investigativo a modificar no aparecen los datos de la actividad.
Localización de la no Conformidad	Gestionar Desempeño Investigativo
Sugerencias	Deberían aparecer los datos de la actividad investigativo para que el usuario pueda modificar los datos del estudiante que estime conveniente.
Atributos asociados	Compleitud (no conformidad) Compleitud (sugerencia)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Yudanis Gago Martínez
Nº	3
Elementos	Interfaz
CU	Registrar Desempeño Investigativo Colectivo
No conformidad	Al tratar de registrar un desempeño investigativo colectivo y dejar vacío (dando espacio) en el campo “nombre trabajo” lo adiciono normalmente
Localización de la no Conformidad	Gestionar Desempeño Investigativo
Sugerencias	No debería dejar registrar una actividad investigativa con esta característica.
Atributos asociados	Compleitud (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Raciel Yera
Nº	1
Elementos	Interfaz

CU	Registrar desempeño laboral individual
No conformidad	A la hora de la validación algunos mensajes los muestra en inglés
Localización de la no Conformidad	Pantalla de registrar desempeño laboral individual.
Sugerencias	Mostrar los mensajes en español, haciendo los cambios pertinentes en los validadores.
Atributos asociados	Corrección (sugerencia) Corrección(no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Raciel Yera
Nº	2
Elementos	Interfaz
CU	Registrar desempeño laboral individual
No conformidad	Al presionar el botón "Cancelar", limpia todos los campos, pero no sale del caso de uso.
Localización de la no Conformidad	Pantalla de registrar desempeño laboral individual.
Sugerencias	Redireccionar hacia otra pantalla tras presionar el botón Cancelar.
Atributos asociados	Compleitud (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Raciel Yera
Nº	3
Elementos	Interfaz
CU	Modificar desempeño laboral
No conformidad	No notifica al usuario en caso de que el envío de los datos sea exitoso
Localización de la no Conformidad	Pantalla de modificar desempeño laboral individual.
Sugerencias	Notificar al usuario que fue exitosa la acción
Atributos asociados	Compleitud (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Raciel Yera
Nº	4
Elementos	Interfaz
CU	Modificar desempeño laboral
No conformidad	No tiene opciones para poner la fecha.
Localización de la no Conformidad	Pantalla de modificar desempeño laboral colectivo

Sugerencias	Debe informarnos en todo momento qué formato debe tener la fecha.
Atributos asociados	Consistencia (sugerencia)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Raciel Yera
Nº	5
Elementos	Interfaz
CU	Modificar desempeño laboral
No conformidad	Los validadores no muestran los mensajes correctamente
Localización de la no Conformidad	Pantalla de modificar desempeño laboral colectivo
Sugerencias	Debe ajustarse toda la parte de la validación.
Atributos asociados	Compleitud (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Raciel Yera
Nº	6
Elementos	Interfaz
CU	Registrar y modificar desempeño laboral colectivo
No conformidad	No corresponden las interfaces asociadas a estos casos de uso con su real implementación
Localización de la no Conformidad	Pantalla de registrar y modificar desempeño laboral colectivo.
Sugerencias	Ajustar las pantallas a la descripción del caso de uso
Atributos asociados	Compleitud (no conformidad) Consistencia (sugerencia)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Serguey Frómeta
Nº	1
Elementos	Interfaz
CU	Registrar Desempeño Docente Individual
No conformidad	El sistema impide el acceso a todas las funcionalidades, excepto con el usuario de administración.
Localización de la no Conformidad	Página Principal
Sugerencias	El nivel de acceso debe ser gestionado por roles.
Atributos asociados	Compleitud (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza

Tipo de Prueba	Funcional
Nombre del Probador	Serguey Frómeta
Nº	2
Elementos	Interfaz
CU	Registrar Desempeño Docente Individual
No conformidad	El sistema muestra la información donde se debe registrar la misma
Localización de la no Conformidad	Gestionar Desempeño Docente
Sugerencias	La inserción de datos debe separarse de los resultados. Los estudiantes deben mostrarse en un campo de selección.
Atributos asociados	Complejidad (no conformidad) Consistencia (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Serguey Frómeta
Nº	3
Elementos	Interfaz
CU	Registrar Desempeño Docente Colectivo
No conformidad	El sistema notifica una Excepción desconocida.
Localización de la no Conformidad	Gestionar Desempeño Docente
Sugerencias	Cuando el usuario trata de insertar datos, el sistema notifica una excepción desconocida.
Atributos asociados	Complejidad (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Yayneris Zambrana Hernández
Nº	1
Elementos	Interfaz
CU	Administrar Actividades
No conformidad	Cuando entro una combinación de caracteres con caracteres extraños y doy en la opción "Aceptar" no me agrega la actividad pero cuando voy a "Cancelar" no me cancela y me deja el nombre que entré con caracteres extraños
Localización de la no Conformidad	En el botón Cancelar
Sugerencias	Debe ajustarse toda la parte de la validación.
Atributos asociados	Complejidad (no conformidad) Integridad (sugerencia)
Fecha:	7 de mayo de 2007

Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Yayneris Zambrana Hernández
Nº	2
Elementos	Interfaz
CU	Administrar Actividades
No conformidad	Cuando pongo el Mouse encima de las opciones en la parte de arriba de la página, y lo quito estas opciones se quedan como si aún el Mouse la estuviera marcando.
Localización de la no Conformidad	En las opciones de la parte de arriba de la página
Sugerencias	Debe ajustarse toda la parte de la validación.
Atributos asociados	Complejidad (no conformidad) Integridad (sugerencia)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Yayneris Zambrana Hernández
Nº	3
Elementos	Interfaz
CU	Modificar datos de la actividades
No conformidad	Cuando entro una actividad y doy en el botón cancelar no me borra lo que tenía escrito anteriormente.
Localización de la no Conformidad	En el botón Cancelar.
Sugerencias	Debe ajustarse toda la parte de la validación.
Atributos asociados	Complejidad (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Yayneris Zambrana Hernández
Nº	4
Elementos	Interfaz
CU	Eliminar Actividad
No conformidad	No elimina ninguna actividad.
Localización de la no Conformidad	
Sugerencias	Pienso que deberían tener una opción de Ninguna Opción...
Atributos asociados	Complejidad (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Funcional
Nombre del Probador	Yeleyne Maure

Nº	1
Elementos	Interfaz
CU	Gestionar Usuario
No conformidad	Cuando se crea un usuario, el único rol que se le asigna es Administrador, no ninguno otro de los que se especifica en el CU.
Localización de la no Conformidad	Gestionar Información Usuario
Sugerencias	Considero que debe quedarse así, pues la aplicación está hecha para los estudiantes, o sea, no para Trabajadores. En los RF, se plantea que el administrador será quien controle los procesos administrativos de la aplicación, por tanto ha de ser el único perfil que se asigne
Atributos asociados	Complejidad (no conformidad)

3.2.1.2 Pruebas de Seguridad

A pesar de que no se contó con un requisito no funcional, que requiriera verificar la seguridad del Caso de Estudio de Ficha Técnica del proyecto CICPC, las responsables de las pruebas de calidad del proyecto CICPC Yilennis Mendoza Torres y Yuniet Toll Palma decidieron desarrollar esta otra prueba para verificar con qué seguridad contaba dicha aplicación.

Tabla 11 Resultados de las Pruebas de Sistema al Caso de Estudio Ficha Técnica

Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Seguridad
Nombre del Probador	José Enrique Altuna
Nº	1
Elementos	Interfaz
Nombre del Requisito No Funcional	
No conformidad	Una vez que hice el proceso de autenticación con el usuario admin. Y pasé al panel de administración copié la dirección de esta funcionalidad y al deslogearme y pegar la dirección anterior me relogea sin tener que introducir usuario y contraseña
Localización de la no Conformidad	
Sugerencias	
Atributos asociados	Tolerancia de errores o fallos (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Seguridad

Nombre del Probador	Yunior Alayo Rondón
Nº	1
Elementos	Interfaz
Nombre del Requisito No Funcional	
No conformidad	Luego de haber entrado al sitio vuelvo a la página principal y trato de autenticarme y me deja sin haber salido de la sesión anterior.
Localización de la no Conformidad	Página principal
Sugerencias	Controlar que un usuario no vea la página de autenticación luego de haberse autenticado. Otra sugerencia sería controlar que nadie se pueda autenticar si ya hay un usuario autenticado
Atributos asociados	Tolerancia de errores o fallos (no conformidad) Tolerancia de errores o fallos (sugerencia)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Seguridad
Nombre del Probador	Yunior Alayo Rondón
Nº	2
Elementos	Interfaz
Nombre del Requisito No Funcional	
No conformidad	El vínculo que permite salir de la sesión actual no está funcionando bien ya que solo redirecciona a una página determinada pero el usuario actual permanece activo
Localización de la no Conformidad	Todas las páginas
Sugerencias	Revisar la implementación del sistema de salida de la aplicación
Atributos asociados	Tolerancia de errores o fallos (no conformidad) Concisión (sugerencia)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Seguridad
Nombre del Probador	Reiniel Herrera Pereda
Nº	1
Elementos	Interfaz
Nombre del Requisito No Funcional	
No conformidad	La sesión permanece abierta cuando en la página que está usándose de la aplicación se abre otra página, después con tan solo poner la dirección de la aplicación que quieras, puedes entra.
Localización de la no Conformidad	

Sugerencias	Esto se cumple con todas las direcciones de la aplicación.
Atributos asociados	Tolerancia de errores o fallos (no conformidad)

3.2.1.3 Pruebas de Stress o Resistencia

A pesar de que no se contó con un requisito no funcional que requiriera verificar el Stress o Resistencia del Caso de Estudio de Ficha Técnica del proyecto CICPC, las responsables de las pruebas de calidad del proyecto CICPC Yilennis Mendoza Torres y Yuniet Toll Palma decidieron desarrollar esta prueba para comprobar el estado de la aplicación en cuanto al siguiente requisito no funcional para poder medir cuantos usuarios pueden conectarse concurrentemente

Tabla 12 Resultados de las Pruebas de Sistema al Caso de Estudio Ficha Técnica

Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Stress o Resistencia
Nombre del Probador	Isis Margarita Blanco Zamora
Nº	1
Elementos	Interfaz
Nombre del Requisito No Funcional	
No conformidad	La aplicación no responde al tener 12 usuarios conectados, siendo las 6:20 p.m.
Localización de la no Conformidad	
Sugerencias	Debe programarse una forma de contrastar o evitar el exceso de "flooding", para que la aplicación no colapse
Atributos asociados	Comportamiento del tiempo (sugerencia)
	Comportamiento del tiempo (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Stress o Resistencia
Nombre del Probador	Isis Margarita Blanco Zamora
Nº	2
Elementos	Interfaz
Nombre del Requisito No Funcional	
No conformidad	El 7 de mayo 2007 a las 4:20 PM con 12 usuarios explota el sistema Ficha Técnica
Localización de la no	

Conformidad	
Sugerencias	
Atributos asociados	Comportamiento del tiempo (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Stress o Resistencia
Nombre del Probador	Isis Margarita Blanco Zamora
Nº	3
Elementos	Interfaz
Nombre del Requisito No Funcional	
No conformidad	El sistema muestra un mensaje de error al tener 150 solicitudes, y a partir de ese momento no responde el servidor donde se encuentra la aplicación Ficha Técnica, siendo las 6:20 p.m.
Localización de la no Conformidad	PC servidor de la aplicación de Ficha Técnica.
Sugerencias	Debe implementarse algún método para que la aplicación funcione aunque la PC servidor cierre la sesión en la que se trabaja en ese momento.
Atributos asociados	Comportamiento del tiempo (no conformidad) Comportamiento del recursos (sugerencia)

3.2.1.4 Pruebas de Disponibilidad y Red

Tabla 13 Resultados de las Pruebas de Sistema al Caso de Estudio Ficha Técnica

Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Disponibilidad y Red
Nombre del Probador	Jorge Amado Soria
Nº	1
Elementos	Interfaz
Nombre del Requisito No Funcional	RA1.Fiabilidad - El sistema deberá estar disponible las 24 horas - El tiempo permitido que el sistema puede estar fuera de operación después de un fallo no debe exceder 2 días.
No conformidad	
Localización de la no Conformidad	
Sugerencias	Cambiar el IP y el puerto por un alias, por ejemplo, http://FichaTécnica.uci.cu
Atributos asociados	Tolerancia de errores o fallos (no conformidad)
Fecha:	7 de mayo de 2007

Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Disponibilidad y Red
Nombre del Probador	Jorge Amado Soria
Nº	2
Elementos	Interfaz
Nombre del Requisito No Funcional	RA2.Fiabilidad - El sistema deberá estar disponible las 24 horas - El tiempo permitido que el sistema puede estar fuera de operación después de un fallo no debe exceder 2 días.
No conformidad	
Localización de la no Conformidad	
Sugerencias	Replicación de la aplicación para distribuir la carga sobre varios servidores, aunque todos actúan sobre la misma base de datos.
Atributos asociados	Tolerancia de errores o fallos (sugerencia)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Disponibilidad y Red
Nombre del Probador	Jorge Amado Soria
Nº	3
Elementos	Interfaz
Nombre del Requisito No Funcional	RA3 Fiabilidad - El sistema deberá estar disponible las 24 horas - El tiempo permitido que el sistema puede estar fuera de operación después de un fallo no debe exceder 2 días.
No conformidad	
Localización de la no Conformidad	
Sugerencias	Revisar la carga del backup periódicamente
Atributos asociados	Tolerancia de errores o fallos (sugerencia)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Disponibilidad y Red
Nombre del Probador	Arniel Serrano
Nº	1
Elementos	Interfaz
Nombre del Requisito No Funcional	RA4.Fiabilidad - El sistema deberá estar disponible las 24 horas - El tiempo permitido que el sistema puede estar fuera de operación después de un fallo no debe exceder 2 días.
No conformidad	
Localización de la no Conformidad	

Sugerencias	El servidor debe estar configurado para auto recuperación en casos de fallos eléctricos. Esto incluye una función de reinicio automático si el servidor se cae.
Atributos asociados	Tolerancia de errores o fallo (sugerencia)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Disponibilidad y Red
Nombre del Probador	Arniel Serrano
Nº	2
Elementos	Interfaz
Nombre del Requisito No Funcional	RA5.Fiabilidad - El sistema deberá estar disponible las 24 horas - El tiempo permitido que el sistema puede estar fuera de operación después de un fallo no debe exceder 2 días.
No conformidad	
Localización de la no Conformidad	
Sugerencias	Deshabilitar o detectar flooding (refrescos numerosos) para evitar que los servlets salgan de funcionamiento.
Atributos asociados	Tolerancia de errores o fallo (sugerencia)

3.2.1.5 Pruebas de Compatibilidad

Tabla 14 Resultados de las Pruebas de Sistema al Caso de Estudio Ficha Técnica

Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Compatibilidad, Sistema Operativo: Windows 2003 Server, Navegador: NetsCape y Opera
Nombre del Probador	David Torres Vázquez
Nº	1
Elementos	Interfaz
Nombre del Requisito No Funcional	RA6.Usabilidad - Para poder ver las páginas adecuadamente necesita utilizar un navegador compatible con estándares Web. - Proporcionar al usuario información relacionada con el estado actual del sistema
No conformidad	En el navegador Opera no sale la fecha correctamente
Localización de la no Conformidad	Al abrir la aplicación
Sugerencias	Tratar que la aplicación sea compatible con estos navegadores Opera, NetsCape.
Atributos asociados	Adaptabilidad (no conformidad)

	Adaptabilidad (sugerencia)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Compatibilidad, Sistema Operativo: Windows 2003 Server, Navegador: NetsCape y Opera
Nombre del Probador	David Torres Vázquez
Nº	2
Elementos	Interfaz
Nombre del Requisito No Funcional	RA7.Usabilidad - Para poder ver las páginas adecuadamente necesita utilizar un navegador compatible con estándares Web. - Proporcionar al usuario información relacionada con el estado actual del sistema
No conformidad	En el navegador NetsCape no sale la fecha correctamente.
Localización de la no Conformidad	
Sugerencias	
Atributos asociados	Adaptabilidad (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Compatibilidad, Sistema Operativo: Windows 2003 Server, Navegador: NetsCape y Opera
Nombre del Probador	David Torres Vázquez
Nº	3
Elementos	Interfaz
Nombre del Requisito No Funcional	RA8.Usabilidad - Para poder ver las páginas adecuadamente necesita utilizar un navegador compatible con estándares Web. - Proporcionar al usuario información relacionada con el estado actual del sistema
No conformidad	Que cuando cierras sesión y das al botón atrás del navegador puede seguir navegando como usuario administrador y no cierra la sesión actual.
Localización de la no Conformidad	Cualquier página que entres por administrador
Sugerencias	
Atributos asociados	Instalabilidad (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Compatibilidad, Sistema Operativo: Windows 2003 Server, Navegador: NetsCape y Opera
Nombre del Probador	David Torres Vázquez

Nº	4
Elementos	Interfaz
Nombre del Requisito No Funcional	RA9.Usabilidad - Para poder ver las páginas adecuadamente necesita utilizar un navegador compatible con estándares Web. - Proporcionar al usuario información relacionada con el estado actual del sistema
No conformidad	Que cuando genera el reporte lo que hace es guardar una página jsf
Localización de la no Conformidad	consultar_datos_colectivos.jsf
Sugerencias	
Atributos asociados	Adaptabilidad (no conformidad) Instalabilidad (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Compatibilidad, Sistema Operativo: Windows 2003 Server, Navegador: NetsCape y Opera
Nombre del Probador	David Torres Vázquez
Nº	5
Elementos	Interfaz
Nombre del Requisito No Funcional	RA10.Usabilidad - Para poder ver las páginas adecuadamente necesita utilizar un navegador compatible con estándares Web. - Proporcionar al usuario información relacionada con el estado actual del sistema
No conformidad	Puedo acceder a cualquier carpeta donde se encuentre el sitio y modificar los datos
Localización de la no Conformidad	Carpeta del sitio
Sugerencias	
Atributos asociados	Adaptabilidad (no conformidad) Instalabilidad (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Compatibilidad, Sistema Operativo: Linux, Navegador: Mozilla
Nombre del Probador	Banier Reyes Saiz
Nº	1
Elementos	Interfaz
Nombre del Requisito No Funcional	RA11.Usabilidad -Para poder ver las páginas adecuadamente necesita utilizar un navegador compatible con estándares Web. - Proporcionar al usuario información relacionada con el estado actual del sistema

No conformidad	Las letras no se definen en que región están.
Localización de la no Conformidad	
Sugerencias	No se pudieron capturar imágenes como prueba de lo planteado anteriormente.
Atributos asociados	Adaptabilidad (no conformidad) Instalabilidad (sugerencia)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Compatibilidad, Sistema Operativo: Linux, Navegador:Mozilla
Nombre del Probador	Banier Reyes Saiz
Nº	2
Elementos	Interfaz
Nombre del Requisito No Funcional	RA12.Usabilidad -Para poder ver las páginas adecuadamente necesita utilizar un navegador compatible con estándares Web. - Proporcionar al usuario información relacionada con el estado actual del sistema
No conformidad	La fecha no sale correctamente.
Localización de la no Conformidad	
Sugerencias	
Atributos asociados	Adaptabilidad (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Compatibilidad, Sistema Operativo: Linux, Navegador:Mozilla
Nombre del Probador	Banier Reyes Saiz
Nº	3
Elementos	Interfaz
Nombre del Requisito No Funcional	RA13.Usabilidad -Para poder ver las páginas adecuadamente necesita utilizar un navegador compatible con estándares Web. - Proporcionar al usuario información relacionada con el estado actual del sistema
No conformidad	La fecha no sale correctamente.
Localización de la no Conformidad	
Sugerencias	
Atributos asociados	Adaptabilidad (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza

Tipo de Prueba	Compatibilidad, Sistema Operativo: Linux, Navegador:Mozilla
Nombre del Probador	Banier Reyes Saiz
Nº	4
Elementos	Interfaz
Nombre del Requisito No Funcional	RA14.Usabilidad -Para poder ver las páginas adecuadamente necesita utilizar un navegador compatible con estándares Web. - Proporcionar al usuario información relacionada con el estado actual del sistema
No conformidad	Las letras muy grandes y sin estética.
Localización de la no Conformidad	
Sugerencias	
Atributos asociados	Adaptabilidad (no conformidad) Instalabilidad (no conformidad)
Fecha:	7 de mayo de 2007
Responsables:	Yuniet Toll y Yilennis Mendoza
Tipo de Prueba	Compatibilidad, Sistema Operativo: Linux, Navegador:Mozilla
Nombre del Probador	Banier Reyes Saiz
Nº	5
Elementos	Interfaz
Nombre del Requisito No Funcional	RA15.Usabilidad -Para poder ver las páginas adecuadamente necesita utilizar un navegador compatible con estándares Web. - Proporcionar al usuario información relacionada con el estado actual del sistema
No conformidad	Los reportes salen como un documento adjunto que se puede descargar, sin embargo, los datos de los usuarios salen en el mismo sitio.
Localización de la no Conformidad	
Sugerencias	
Atributos asociados	Adaptabilidad (no conformidad) Instalabilidad (no conformidad)

3.2.2 Representación Gráfica de los resultados

La representación gráfica de los resultados obtenidos en las pruebas realizadas al Caso de Estudio Ficha Técnica, según los siguientes parámetros: Distribución de Probadores, No Conformidades, Sugerencias, No Conformidades por Probador, Sugerencias por Probador. A continuación se muestra:

Tabla 15 Datos por tipo de prueba

Tipo de Prueba	Probadores	No Conf.	Sug.	No Conf./Probador	Sug./Probador
Funcional	13	33	29	2,54	2,23
Compatibilidad	2	10	2	5,00	1,00
Seguridad	3	4	3	1,33	1,00
Stress	1	3	2	3,00	2,00
Disponibilidad y Red	2	0	5	0,00	2,50
Totales	21	50	41	2,38	1,95

A continuación se muestra gráficamente la distribución de los probadores para cada tipo de prueba aplicada.

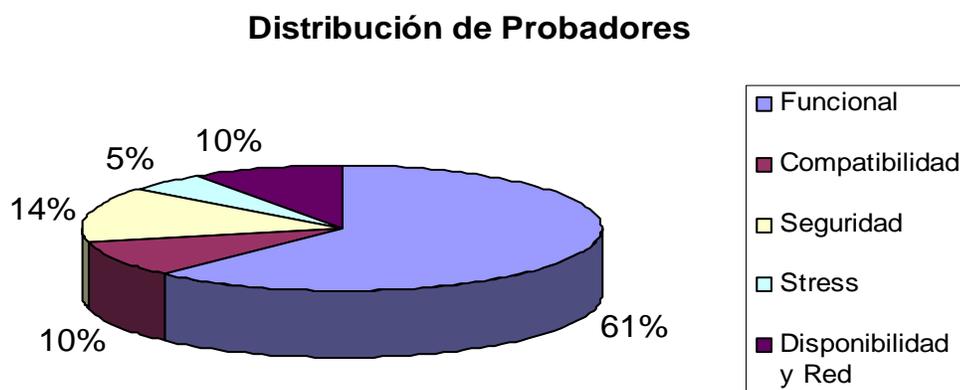


Fig. 5 Distribución de Probadores

A continuación se muestra gráficamente la cantidad de no conformidades encontradas por cada prueba realizada.

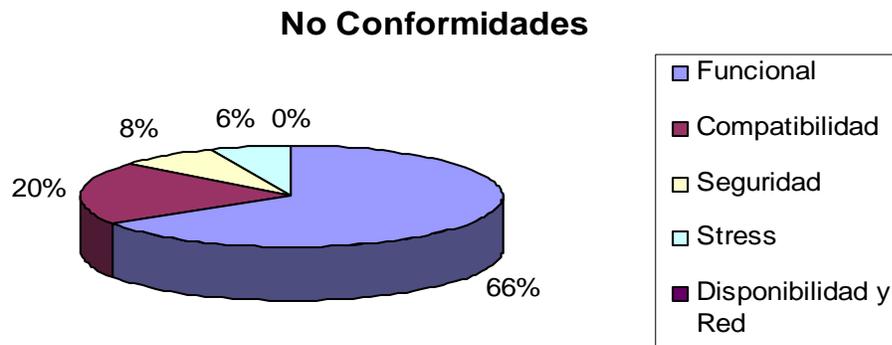


Fig. 6 No Conformidades

A continuación se muestra gráficamente la cantidad de sugerencias por cada prueba realizada.

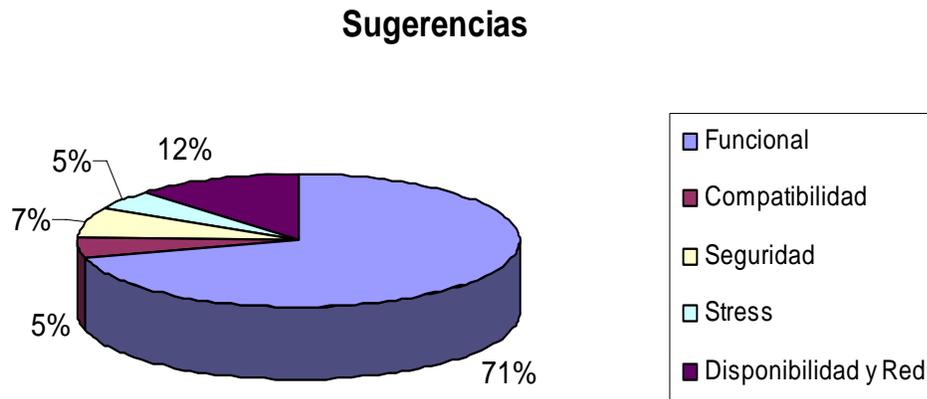


Fig. 7 Sugerencias

A continuación se muestra gráficamente la cantidad de no conformidades encontradas como promedio por cada probador en las pruebas realizadas

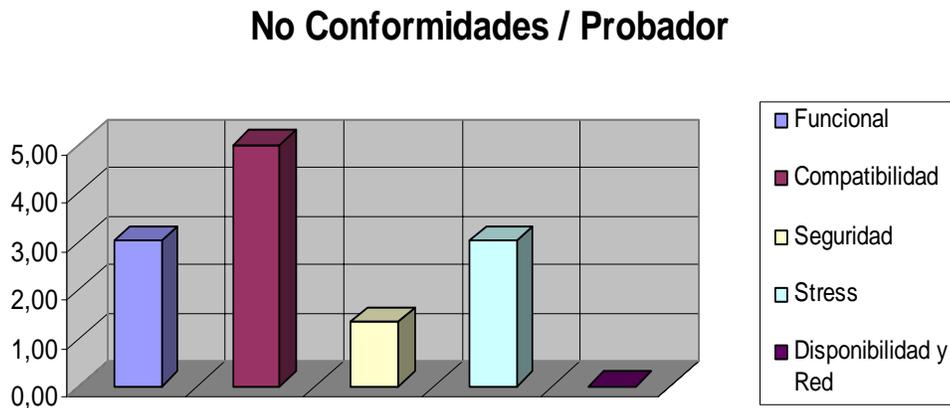


Fig. 8 No Conformidades por Probadores

A continuación se muestra gráficamente la cantidad de sugerencias como promedio por cada probador en las pruebas realizadas.

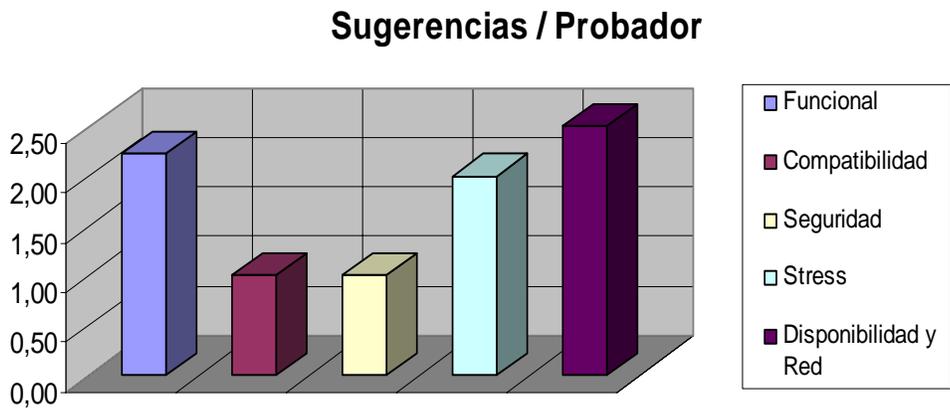


Fig. 9 Sugerencias por Probadores

3.2.2.2 Representación Gráfica según estas subcaracterísticas o atributos de calidad

Representación gráfica de los resultados obtenidos en las pruebas realizadas al Caso de Estudio Ficha Técnica según los siguientes criterios o subcaracterísticas de calidad:

Basadas en la ISO 9126 se definió asociar las características y atributos a las pruebas de sistemas seleccionadas en este trabajo

Tabla 16 Características y atributos de la ISO 9126 asociados a las pruebas de sistema

Características ISO 9126	Atributo	Asociadas a las Prueba de Sistema
Funcionalidad	Complejidad Consistencia Corrección Integridad Normalización	Funcional
Confiabilidad	Recuperabilidad Tolerancia de errores o fallos	Seguridad Disponibilidad y red Fiabilidad
Portabilidad	Adaptabilidad Instalabilidad	Compatibilidad
Eficiencia	Comportamiento del tiempo Comportamiento de los recursos.	Stress Rendimiento
Mantenibilidad	Diagnosticabilidad Consciencia	Seguridad

Tabla 17 Métricas de calidad por tipo de Pruebas de Sistema y atributos asociados según la ISO 9126

Tipo de Prueba	Atributos	No Conf.	Sug.	Puntuación	Escala
Funcionalidad	Complejidad	30	7	0,16	BAJO
	Consistencia	3	6	4,52	ALTO
	Corrección	2	1	4,68	ALTO
	Integridad	0	2	5,00	ALTO
	Normalización	1	1	4,84	ALTO
Confiabilidad	Tolerancia de errores o fallos	8	2	3,71	MEDIO
Eficiencia	Comportamiento del tiempo	3	1	4,52	ALTO
	Comportamiento de los recursos	0	1	5,00	ALTO
Mantenibilidad	Consciencia	0	1	5,00	ALTO
Portabilidad	Adaptabilidad	9	1	3,55	MEDIO
	Instalabilidad	5	1	4,19	MEDIO
Totales		47	21		

La diferencia de cantidad de no conformidades entre ambos resultados mostrados en la tabla de las métricas y la de los datos para medir las pruebas se debe a la existencia de no conformidades que cumplen con más de un atributo de calidad, tales como: en la prueba de compatibilidad la no conformidad del probadores David Vázquez y Banier Reyes cumplen con los atributos adaptabilidad e instalabilidad, la prueba funcional la no conformidad de los probadores Serguey Frómata y Liana Toledo cumplen con los atributos completitud y consistencia, y en la no conformidad del probador Isis Blanco con los atributos completitud, corrección y consistencia.

Escala: es el por ciento que representa la no conformidad según el atributo o tipo de prueba que cumple en una escala del 0 al 5 donde Bajo está en el rango de 0 a 3,5, Medio está en el rango de 3,5 a 4,5 y Alto está en el rango de 4,5 a 5. Si se da el caso de que el valor obtenido está fuera del rango especificado la escala que toma es la que le continúa (ejemplo: el valor es 3,5 toma la escala siguiente, es decir, la escala de Medio)

La métrica “Puntuación”: es una métrica de evaluación de la calidad para las pruebas de sistemas (Castillo 2007). Se obtiene a través de una función basada en la ISO 9126 para medir las no conformidades que están asociados a un tipo de prueba y atributo de calidad, la función es: : al Valor Máximo de la no conformidad encontrada según el tipo de prueba y atributo asociado ($VMNC$) se le resta el valor de la no conformidad encontrada asociada ($VNCE$) a ese tipo de prueba y atributo de calidad, el resultado de la resta se multiplica por 5 y este valor se divide por el Valor Máximo de la no conformidad encontrada según el tipo de prueba y atributo asociado ($VMNC$).

$$MP = \frac{(VMNC - VNCE) * 5}{VMNC}$$

Ejemplo:

$$MP = \frac{(31 - 30) * 5}{31}$$

$$MP = 0,16$$

Por tanto el caso de estudio Ficha Técnica se encuentra en la escala de bajo, por lo que presenta baja calidad.

En dependencia de la cantidad de no conformidades se determina si la calidad del producto es Baja, Media o Alta, ejemplo: en los resultados obtenidos del Caso de Estudio Ficha Técnica la calidad es Baja debido al alto número de no conformidades que cumplen con el atributo Completitud, es decir a mayor número de no conformidades menor calidad según el atributo que esté implicado y el tipo de prueba que se esté probando, esto no indica que la calidad del producto sea totalmente Baja, sino que para ese atributo en ese tipo de prueba es Baja la calidad, para esto hay que tener en cuenta cuanto repercute el atributo en el producto. El atributo Completitud es el que mide que en la aplicación estén implementadas todas las funcionalidades descritas por el cliente, si ocurre lo contrario como es el caso pues entonces la calidad del Caso de Estudio es totalmente Baja.

No Conformidad: es la cantidad de no conformidades encontradas por cada tipo de prueba sistema y atributos asociados a la misma.

Sugerencias: es la cantidad de sugerencias encontradas por cada tipo de prueba sistema y atributos asociados a la misma.

A continuación se muestra gráficamente la cantidad de no conformidades encontradas por tipo de prueba y atributos de calidad.

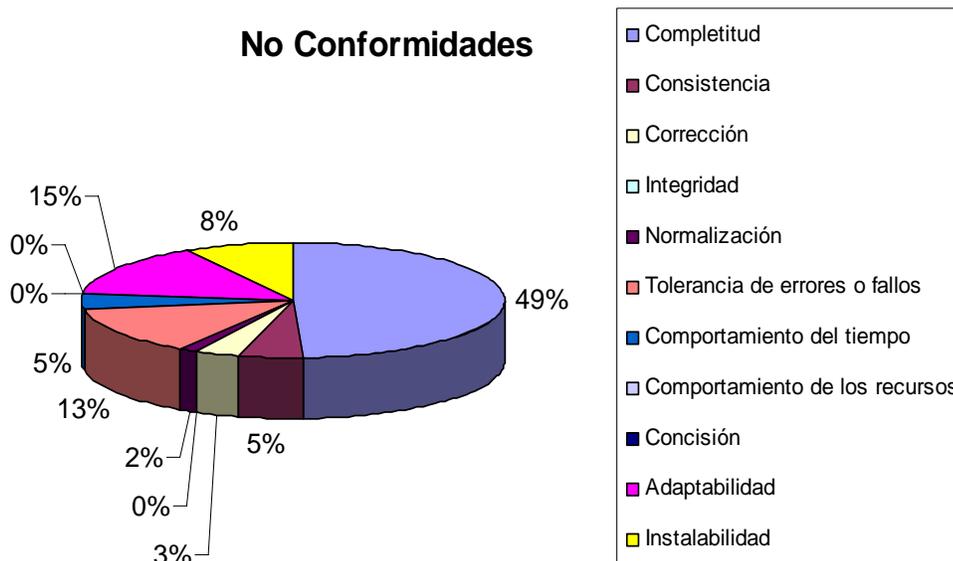


Fig. 10 No Conformidades por tipo de prueba y atributos de calidad

A continuación se muestra gráficamente la cantidad de sugerencias encontradas por tipo de prueba y atributos de calidad.

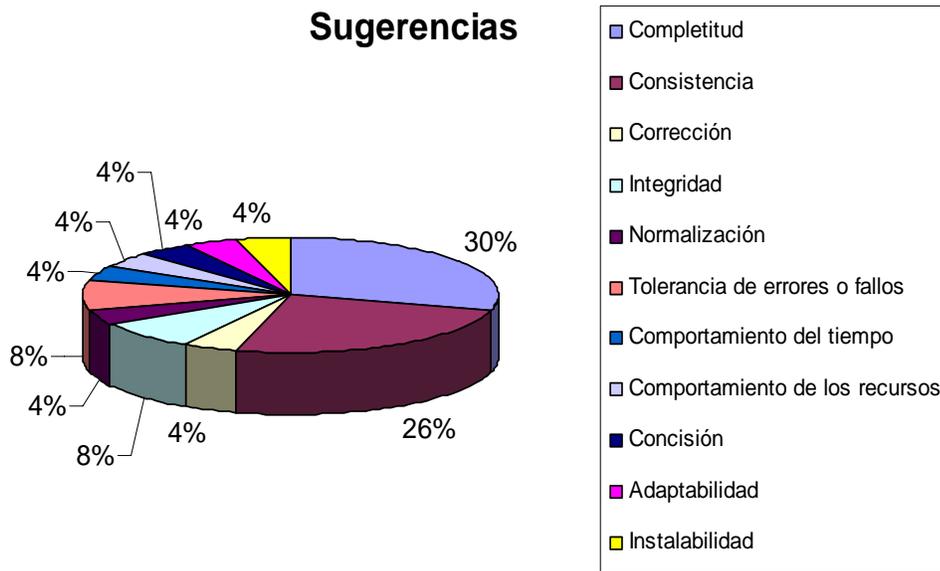


Fig. 11 Sugerencias por tipo de prueba y atributos de calidad

A continuación se muestra gráficamente la puntuación que obtiene dicha subcaracterística o atributo de calidad en una escala del 1 al 5 según con la no conformidad que cumpla.

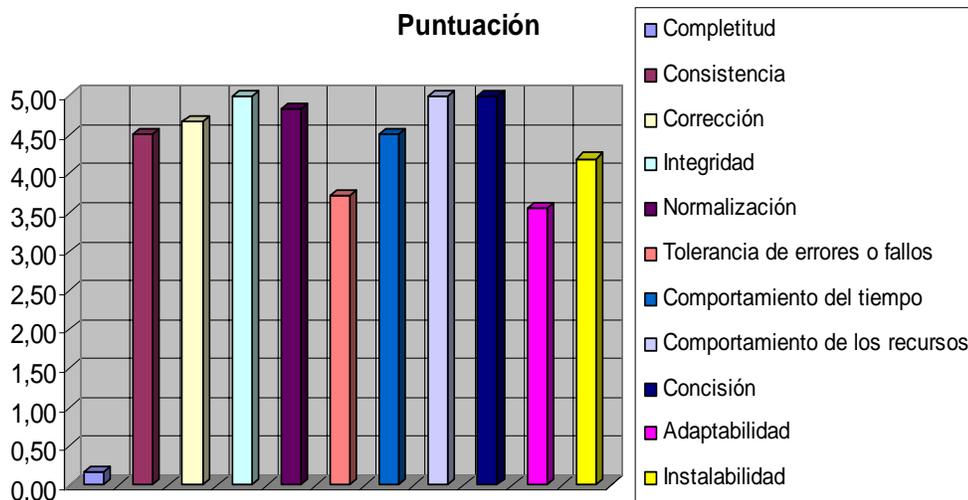


Fig. 12 Puntuación

A continuación se muestra gráficamente el por ciento que representa la no conformidad según el tipo de prueba y atributos que cumplan con la escala de 0 al 5 donde Bajo está en el rango de 0 a 3,5, Medio está en el rango de 3,5 a 4,5 y Alto está en el rango de 4,5 a 5.

Tabla 18 Escala

Escala	Cantidad
Bajo	1
Medio	3
Alto	7

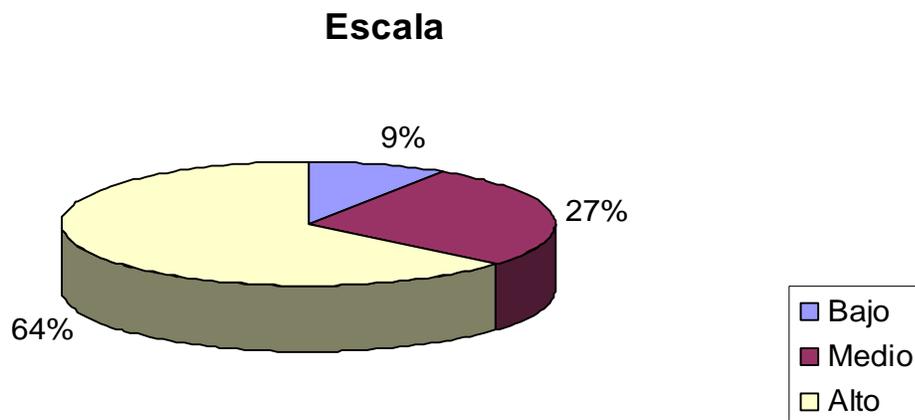


Fig. 13 Escala

3.2.3 Análisis de los resultados de las pruebas realizadas

Haciendo un análisis profundo de los resultados obtenidos en las pruebas realizadas a este Caso de Estudio se detectó que:

- ✓ Hubo que exigir que la aplicación estuviera disponible en el servidor, el cual era un servidor para JAVA, que solo admite 150 peticiones, al llegar a este límite de peticiones colapsaba el servidor mostrando un mensaje de error. Esto ocurría porque no contaban con un servidor propio sino que era común con la estación de trabajo. Esto traía como consecuencia que se perdiera toda la información obtenida hasta el momento teniéndose que empezar desde cero.
- ✓ Las pruebas se retrasaron debido a la no existencia de las analistas del Proyecto CICPC, pues no se contaba con la descripción correcta para llegar al caso de uso que se fuese a probar y tampoco se contaba con un mapa de navegación que explicara la navegabilidad de la aplicación.

- ✓ No se pudieron ejecutar todas pruebas previstas en la estrategia propuesta porque las analistas de sistema no estaban presentes y las responsables de prueba tuvieron que hacer función del rol de analista y diseñadoras de los casos de pruebas, trayendo como consecuencia su retraso.
- ✓ No basta con una hora para preparar el entorno de laboratorio, se necesita más tiempo pues además de levantar las máquinas virtuales y la aplicación que se va a probar es necesario hacer algunas configuraciones a la máquina virtual para que se pueda ver la aplicación.
- ✓ No se pudo contar con el equipo completo de calidad, cuando el laboratorio era de 30 PC para realizar las pruebas, pues solo estaban preparando el entorno del laboratorio un administrador de sistema del Proyecto CICPC y dos responsables de calidad.
- ✓ Ficha Técnica, se encuentra hoy en fase de desarrollo, pues las pruebas pilotos y de calidad a las que fue sometida la aplicación, arrojaron insatisfacciones por parte del cliente que se previeron erradicar en una segunda versión del sistema.

3.3 Resultados Obtenidos del Caso de Estudio Soporte Técnico del Proyecto de CICPC

Las pruebas de sistema realizadas a este Caso de Estudio se ejecutaron el 10 de mayo de 2007.

Para la realización de las pruebas de sistema de dicho caso de estudio se contó con 13 probadores, 8 probadores para las pruebas funcionales, 2 para las pruebas de compatibilidad, 1 para las pruebas de seguridad, 1 para la prueba de stress, 1 para la prueba de fiabilidad, en la primera hora se realizaron las pruebas funcionales, las de seguridad y fiabilidad y en la segunda hora se realizaron las pruebas de stress y compatibilidad.

La selección de los probadores la efectuó la segunda líder del Proyecto CICPC de acuerdo a la cantidad de probadores que se necesitaban por cada tipo de prueba, y la selección de alguno de ellos estuvo dada según las habilidades que tenían para desarrollar dicha actividad.

Las pruebas se realizaron en un marco de dos horas, a las 5:30 PM comenzaron y a las terminaron 7:30 PM. Se necesitó una hora para preparar el entorno de laboratorio.

Todas las orientaciones pertinentes para esta actividad fueron distribuidas por la segunda líder del Proyecto a los probadores. Las responsables por parte del equipo de calidad realizar las pruebas de sistema fueron Yuniet Toll y Yilennis Mendoza que planificaron la distribución de cada una de las pruebas y la cantidad de probadores a cada una de las integrantes del equipo de calidad del Proyecto CICPC.

Al finalizar las pruebas se recopiló toda la información obtenida y se graficaron estos resultados que aparecen a continuación:

3.3.1 Resultados Obtenidos

3.3.1.1 Pruebas Funcionales

Tabla 19 Resultados de las Pruebas de Sistema al Caso de Estudio Soporte Técnico

Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Funcional
Nombre del Probador	Arlan Gálvez Alonso
Nº	1
Elementos	Interfaz
CU	Modificar la cantidad de los medios del LAB.
No conformidad	Se insertaron 459 roturas y las aceptó e incluyó el comentario y la rotura, luego se fue directamente a la página de inicio.
Localización de la no Conformidad	Página de gestionar roturas
Sugerencias	
Atributos asociados	Compleitud)no conformidad)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Funcional
Nombre del Probador	Arlan Gálvez Alonso
Nº	2
Elementos	Interfaz
CU	Modificar la cantidad de los medios del LAB.
No conformidad	Cuando te encuentras en la pantalla de gestionar roturas y la das “agregar nueva actualización”,este muestra una página de error del servidor, (ver Figura1)
Localización de la no Conformidad	Página gestionar Roturas
Sugerencias	
Atributos asociados	Compleitud (no conformidad)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Funcional
Nombre del Probador	Arlan Gálvez Alonso
Nº	3

Elementos	Interfaz
CU	Modificar la cantidad de los medios del LAB.
No conformidad	Al seleccionar un medio para adicionar la nueva rotura, esta te lleva a cualquier parte de la aplicación, o generalmente no hace ningún cambio en la misma o te envía a la página de inicio.
Localización de la no Conformidad	Página gestionar Roturas
Sugerencias	
Atributos asociados	Complejidad (no conformidad)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Prueba Funcional
Nombre del Probador	Arlan Gálvez Alonso
Nº	4
Elementos	Interfaz
CU	Modificar la cantidad de los medios del LAB.
No conformidad	Cuando seleccionas el medio para gestionar las roturas del mismo, este te envía a la página de Gestionar Solicitud (ver imagen2 y 3).
Localización de la no Conformidad	Página de gestionar roturas
Sugerencias	
Atributos asociados	Complejidad (no conformidad)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Funcional
Nombre del Probador	Arlan Gálvez Alonso
Nº	5
Elementos	Interfaz
CU	Modificar la cantidad de los medios del LAB.
No conformidad	Cambio de página y Cambio completamente el diseño.
Localización de la no Conformidad	Página de gestionar Roturas
Sugerencias	
Atributos asociados	Complejidad (no conformidad)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Funcional
Nombre del Probador	Arlan Gálvez Alonso
Nº	6
Elementos	Interfaz

CU	Modificar componentes de la computadora
No conformidad	Cuando seleccionas: el laboratorio o la computadora , esta te envía a la página de Gestionar Solicitud o te muestra la página de error del servidor (ver Figura 4 , 5 y 1)
Localización de la no Conformidad	Página Modificar Componentes de la computadora
Sugerencias	
Atributos asociados	Complejidad (no conformidad) Corrección (no conformidad)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Funcional
Nombre del Probador	Arlan Gálvez Alonso
Nº	7
Elementos	Interfaz
CU	Modificar componentes de la computadora.
No conformidad	Cuando seleccionas la computadora a modificar los componentes , se muestran otros datos que no se refieren a esa computadora, generalmente a la anterior, o la última modificación hecha por el usuario
Localización de la no Conformidad	Página Modificar Componentes de la computadora
Sugerencias	
Atributos asociados	Complejidad (no conformidad) Integridad (no conformidad)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Funcional
Nombre del Probador	Arlan Gálvez Alonso
Nº	8
Elementos	Interfaz
CU	Modificar componentes de la computadora.
No conformidad	Cuando das clic en el botón de agregar nuevo componente, este no realiza la función .(ver Figura 6)
Localización de la no Conformidad	Página Modificar Componentes de la computadora
Sugerencias	Agregar nuevo Componente
Atributos asociados	Complejidad (no conformidad)
Fecha	10 de mayo de 2007

Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Funcional
Nombre del Probador	Arlan Gálvez Alonso
Nº	9
Elementos	Interfaz
CU	Modificar componentes de la computadora.
No conformidad	Insertas datos en un lado y sale en otros lugares de la páginas (ver Figura 7)
Localización de la no Conformidad	
Sugerencias	
Atributos asociados	Complejidad (no conformidad) Consistencia (no conformidad)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Funcional
Nombre del Probador	Yander Cáceres Cruz
Nº	1
Elementos	Interfaz
CU	Gestionar Roles
No conformidad	Acepta dos roles de igual nombre.
Localización de la no Conformidad	Crear nuevo Rol
Sugerencias	Validar dicho problema
Atributos asociados	Complejidad (no conformidad)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Funcional
Nombre del Probador	Yander Cáceres Cruz
Nº	2
Elementos	Interfaz
CU	Gestionar Roles
No conformidad	Acepta dos roles de igual nombre, con la única variación de Mayúsculas y Minúsculas
Localización de la no Conformidad	Crear nuevo Rol
Sugerencias	Validar dicho problema
Atributos asociados	Complejidad (no conformidad)

Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Funcional
Nombre del Probador	Yander Cáceres Cruz
Nº	3
Elementos	Interfaz
CU	Gestionar Roles
No conformidad	Elimina roles asignados a usuarios
Localización de la no Conformidad	Eliminar Rol
Sugerencias	Validar dicho problema
Atributos asociados	Complejidad (no conformidad)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Funcional
Nombre del Probador	Yander Cáceres Cruz
Nº	4
Elementos	Interfaz
CU	Gestionar Roles
No conformidad	Muestra mensaje esperado por eliminar rol asignado, después de borrarlo y haciendo la validación con el próximo en la lista.
Localización de la no Conformidad	Eliminar Rol
Sugerencias	Validar dicho problema
Atributos asociados	Complejidad (no conformidad)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Funcional
Nombre del Probador	Leonardo Santos Ramírez
Nº	1
Elementos	Interfaz
CU	Gestionar solicitudes
No conformidad	Las solicitudes atrasadas encontradas no muestran el nombre del solicitante.
Localización de la no Conformidad	Ver Figura 8.
Sugerencias	

Atributos asociados	Compleitud (no conformidad) Corrección (no conformidad)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Funcional
Nombre del Probador	Leonardo Santos Ramírez
Nº	2
Elementos	Interfaz
CU	Gestionar solicitudes
No conformidad	Al seleccionar una solicitud y cambiarle el estado a rechazada y clicar en tramitar mostró un error inesperado.
Localización de la no Conformidad	Ver anexo 9.
Sugerencias	
Atributos asociados	Compleitud (no conformidad) Corrección (no conformidad)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Funcional
Nombre del Probador	Leonardo Santos Ramírez
Nº	3
Elementos	Interfaz
CU	Gestionar solicitudes
No conformidad	No puedo efectuar una prueba al caso de uso puesto que la aplicación no me deja, o sea no actualiza, esta muy lenta.
Localización de la no Conformidad	
Sugerencias	
Atributos asociados	Compleitud (no conformidad)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Funcional
Nombre del Probador	Leonardo Santos Ramírez
Nº	4
Elementos	Interfaz
CU	Gestionar solicitudes.

No conformidad	La aplicación no responde en ocasiones a lo que se le pide, por ejemplo, al clicar sobre gestionar solicitud la aplicación no cambia.
Localización de la no Conformidad	
Sugerencias	
Atributos asociados	Compleitud (no conformidad)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Funcional
Nombre del Probador	Luís Enrique Hernández Vega
Nº	1
Elementos	Interfaz
CU	Actualizar Software
No conformidad	La descripción no concuerda con la interfaz.
Localización de la no Conformidad	Actualizar Software
Sugerencias	
Atributos asociados	Compleitud (no conformidad) Corrección (no conformidad)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Funcional
Nombre del Probador	Luís Enrique Hernández Vega
Nº	2
Elementos	Interfaz
CU	Actualizar Software
No conformidad	Al acceder a actualizar el software disponible no aparece la opción Quitar Software (Figura 10)
Localización de la no Conformidad	
Sugerencias	
Atributos asociados	Compleitud (no conformidad)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Funcional
Nombre del Probador	Luís Enrique Hernández Vega
Nº	3

Elementos	Interfaz
CU	Actualizar Software
No conformidad	Al acceder al vínculo Actualizar Software y refrescar el sistema apareció lo siguiente (Figura 11)
Localización de la no Conformidad	
Sugerencias	
Atributos asociados	Compleitud (no conformidad)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Funcional
Nombre del Probador	Amed Vázquez Pérez
Nº	1
Elementos	Interfaz
CU	Administrar Computadoras
No conformidad	En ocasiones al pinchar sobre el botón nueva computadora o Seleccionar una nueva Computadora el sistema redirecciona para otras páginas que no tienen nada que ver como por ejemplo Administrar roles, Crear reporte o en otros casos el sistema explota.
Localización de la no Conformidad	Administrar Laboratorios (Ver Figura 12)
Sugerencias	
Atributos asociados	Compleitud (no conformidad)
Fecha	jueves 10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Funcional
Nombre del Probador	Amed Vázquez Pérez
Nº	2
Elementos	Interfaz
CU	Administrar Computadoras
No conformidad	Al pinchar en ocasiones me direcciona para la página Gestionar Solicitud.
Localización de la no Conformidad	Administrar Laboratorios (Ver Figura 13)
Sugerencias	
Atributos asociados	Compleitud (no conformidad)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.

Tipo de Prueba	Funcional
Nombre del Probador	Danier Marante Jacas
Nº	1
Elementos	Interfaz
CU	Administrar los medios del laboratorio
No conformidad	No se encuentra en ninguna parte como lo describe el caso de prueba.
Localización de la no Conformidad	Modificar la cantidad de los medios del LAB
Sugerencias	Que la interfaz sea más sugerente
Atributos asociados	Corrección (no conformidad)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Funcional
Nombre del Probador	Danier Marante Jacas
Nº	2
Elementos	Interfaz
CU	Administrar los medios del laboratorio
No conformidad	No se encuentra de forma implícita, ni con la entrada de datos que se especifica en la prueba
Localización de la no Conformidad	Componentes de la Computadora
Sugerencias	Que se pongan de acuerdo los programadores y los arquitectos
Atributos asociados	Consistencia (no conformidad) Corrección (sugerencia)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Funcional
Nombre del Probador	José Carlos Guevara Turruelles
Nº	1
Elementos	Interfaz
CU	Gestionar estado de la PC
No conformidad	El número de las PC no se cambia, es un campo de solo lectura (Ver Figura 14)
Localización de la no Conformidad	Gestionar estado de la PC
Sugerencias	Que sea este campo editable.
Atributos asociados	Complejidad (no conformidad)

Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Funcional
Nombre del Probador	José Carlos Guevara Turruelles
Nº	2
Elementos	Interfaz
CU	Gestionar estado de la PC
No conformidad	Cuando se varía cualquiera de los valores, excepto el IP, no están validados los otros.
Localización de la no Conformidad	Gestionar estado de la PC
Sugerencias	Si está validado es necesario mostrar un mensaje que avise si la entrada es inválida.
Atributos asociados	Compleitud (no conformidad) Consistencia (sugerencia)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Funcional
Nombre del Probador	Adonys Alea Boffill
Nº	1
Elementos	Interfaz
CU	Realizar Solicitud
No conformidad	Después de seleccionar el laboratorio carga sus máquinas, si se quita la selección siguen aún las máquinas y las deja agregar a la lista de la solicitud, además realiza la solicitud con el laboratorio antes seleccionado... (Ver Figura 15)
Localización de la no Conformidad	Realizar solicitud
Sugerencias	Que cuando desecciones limpie la lista de máquinas y el combobox de estas.
Atributos asociados	Compleitud (sugerencia) Compleitud (no conformidad)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Funcional
Nombre del Probador	Adonys Alea Boffill
Nº	2

Elementos	Interfaz
CU	Realizar Solicitud
No conformidad	Cuando abres el Internet Explorer y entras al sitio te sale sin seleccionar los laboratorios, pero si te deslogueas y vuelves a entrar te sale el último LAB que se había quedado
Localización de la no Conformidad	Realizar solicitud
Sugerencias	Validar mejor el CU.
Atributos asociados	Integridad (no conformidad) Complejidad (no conformidad) Complejidad (sugerencia)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Funcional
Nombre del Probador	Adonys Alea Boffill
Nº	3
Elementos	Interfaz
CU	Realizar Solicitud
No conformidad	Agregas un software que ya existe y no te da error
Localización de la no Conformidad	Caso de Prueba Flujo Alterno
Sugerencias	Validar

3.3.1.2 Pruebas de Stress o Resistencia

Tabla 20 Resultados de las Pruebas de Sistema al Caso de Estudio Soporte Técnico

Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Stress o Resistencia
Nombre del Probador	Isis M. Blanco Zamora
Nº	1
Elementos	Interfaz
Nombre del Requisito No Funcional	RA1. Rendimiento RA1.1. El sistema debe permitir al menos 75 conexiones simultáneas con igual rendimiento para todos.
No conformidad	Siendo las 6:17 p.m. Cuando se trata de refrescar la aplicación, esta no responde y sólo se muestra la página principal sin las opciones y

	empezó a las 5:35 PM.
Localización de la no Conformidad	Página principal de la aplicación
Sugerencias	Esto puede haber ocurrido debido a que los probadores del sistema eliminaron la cuenta de invitado de la aplicación y esto no se podía hacer.
Atributos asociados	Comportamiento del tiempo (no conformidad)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Stress o Resistencia
Nombre del Probador	Isis M. Blanco Zamora
Nº	2
Elementos	Interfaz
Nombre del Requisito No Funcional	RA2. Rendimiento RA2.1. El sistema debe permitir al menos 75 conexiones simultáneas con igual rendimiento para todos.
No conformidad	La aplicación colapsa al tener 12 usuarios conectados, siendo las 6:35 p.m. y empezó a probarse a las 5:35 PM.
Localización de la no Conformidad	
Sugerencias	
Atributos asociados	Comportamiento del tiempo (no conformidad) Comportamiento de los recursos (no conformidad)

3.3.1.3 Pruebas de Seguridad

Tabla 21 Resultados de las Pruebas de Sistema al Caso de Estudio Soporte Técnico

Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Seguridad
Nombre del Probador	Yander Cáceres Cruz
Nº	1
Elementos	Interfaz
Nombre del Requisito No Funcional	RA3. Interfaces RA3.1. El sistema usara el protocolo LDAP para las autenticaciones con el dominio UCI
No conformidad	A pesar de que para mi rol no se asigne la posibilidad de Gestionar Roles, puedo acceder a ella por la URL correspondiente
Localización de la no	

Conformidad	
Sugerencias	Verificar la seguridad
Atributos asociados	Tolerancia de errores o fallos (no conformidad)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Seguridad
Nombre del Probador	Yander Cáceres Cruz
Nº	2
Elementos	Interfaz
Nombre del Requisito No Funcional	
No conformidad	Después de hacer LOGOUT puedo acceder a cualquier página por la URL, esto solo se resuelve reiniciando el navegador.
Localización de la no Conformidad	
Sugerencias	Verificar la seguridad
Atributos asociados	Tolerancia de errores o fallos (no conformidad) Diagnosticabilidad (no conformidad)

3.3.1.4 Pruebas de Fiabilidad

Tabla 22 Resultados de las Pruebas de Sistema al Caso de Estudio Soporte Técnico

Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Fiabilidad
Nombre del Probador	Yilennis Mendoza Torres
Nº	1
Elementos	Interfaz
Nombre del Requisito No Funcional	RA5. Fiabilidad RA5.1.El sistema debe tener un tiempo de respuesta menor de 24h ante un fallo que imposibilite su uso RA5.2.Cualquier fallo que limite las funcionalidades del sistema deberá ser reparada antes de las 12h
No conformidad	Durante el desarrollo de estas pruebas se pudo verificar que los administradores del sistema Soporte Técnico se mantuvieron todo el tiempo arreglando online los problemas de la aplicación así como también los fallos que ocurrían durante el tiempo de prueba.
Localización de la no Conformidad	
Sugerencias	Considero que debe de eliminarse esta no conformidad pues esto

	trae desorden durante el tiempo de prueba ya que los probadores opinan y los administradores se pasan todo el tiempo interactuando con los probadores y el sistema.
Atributos asociados	Recuperabilidad (no conformidad) Recuperabilidad (sugerencia)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Fiabilidad
Nombre del Probador	Yilennis Mendoza Torres
Nº	2
Elementos	Interfaz
Nombre del Requisito No Funcional	RA6.Fiabilidad RA6.1 El sistema estará disponible todo los días en el horario laboral (mañana y tarde) y el horario de tiempo de máquina(noche)
No conformidad	Durante el desarrollo de estas pruebas se pudo verificar que el sistema solo está disponible cuando los administradores ponen a correr la aplicación en la PC servidor.
Localización de la no Conformidad	
Sugerencias	Esto no puede ocurrir y deben de remediarlo lo antes posible porque sino el objetivo para el cual está hecha la aplicación no se cumple.
Atributos asociados	Recuperabilidad (no conformidad) Tolerancia de errores o fallos (no conformidad) Tolerancia de errores o fallos (sugerencia)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Prueba de Fiabilidad
Nombre del Probador	Yilennis Mendoza Torres
Nº	3
Elementos	Interfaz
Nombre del Requisito No Funcional	RA7.Fiabilidad RA7.1. El sistema debe tener un tiempo de respuesta menor de 24h ante un fallo que imposibilite su uso. RA7.2. Cualquier fallo que limite las funcionalidades del sistema deberá ser reparada antes de las 12h
No conformidad	Durante el desarrollo de estas pruebas se pudo verificar que el sistema no muestra al cliente cuando hay un fallo pues los administradores de la aplicación se mantuvieron online para hacer todos los arreglos al instante apenas escuchaban a un probador decir lo que pasaba de momento o cuando existía alguna no conformidad del cliente detectada.

Localización de la no Conformidad	
Sugerencias	Verificar mejor el manejo de los administradores para poder validar los errores que ocurren y mostrarlos tanto al cliente como en el servidor.
Atributos asociados	Tolerancia de errores o fallos (no conformidad)

3.3.1.5 Pruebas de Compatibilidad.

Tabla 23 Resultados de las Pruebas de Sistema al Caso de Estudio Soporte Técnico

Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Compatibilidad, Sistema Operativo: Windows 2003 Server Navegador: Opera
Nombre del Probador	Arniel Serrano Hernández
Nº	1
Elementos	Interfaz
Nombre del Requisito No Funcional	RA8. Usabilidad RA8.1. El sistema debe correr sobre cualquier plataforma (software libre o propietario). RA8.2. El sistema podrá correr en cualquier navegador (IE, FireFox, Opera, NetsCape). RA8.3. El sistema debe permitir una navegabilidad fácil a todas sus páginas. RA8.4. El sistema abrirá los enlaces sobre el propio navegador.
No conformidad	No tiene acceso a intranet.uci.cu y a inter-nos.uci.cu debido a que presenta problemas con el método de autenticación. Ver Figura 1
Localización de la no Conformidad	Menú principal, Vínculos: 1-Intranet.uci.cu 2-Inter-nos.uci.cu
Sugerencias	E
Atributos asociados	Adaptabilidad (no conformidad)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Prueba de Compatibilidad, Sistema Operativo: Windows 2003 Server Navegador: Opera
Nombre del	Arniel Serrano Hernández

Probador	
Nº	2
Elementos	Interfaz
Nombre del Requisito No Funcional	RA9. Usabilidad RA9.1. El sistema debe correr sobre cualquier plataforma (software libre o propietario) RA9.2. El sistema podrá correr en cualquier navegador (IE, FireFox, Opera, NetsCape) RA9.3. El sistema debe permitir una navegabilidad fácil a todas sus páginas RA9.4. El sistema abrirá los enlaces sobre el propio navegador
No conformidad	
Localización de la no Conformidad	
Sugerencias	La configuración de los navegadores tiene que mejorarse para navegar en la red ya que no todos los navegadores no se configuran igual.
Atributos asociados	Instalabilidad (sugerencia)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Prueba de Compatibilidad, Sistema Operativo: Windows 2003 Server Navegador: Opera
Nombre del Probador	Arniel Serrano Hernández
Nº	3
Elementos	Interfaz
Nombre del Requisito No Funcional	RA10. Usabilidad RA10.1. El sistema debe correr sobre cualquier plataforma (software libre o propietario) RA10.2. El sistema podrá correr en cualquier navegador (IE, FireFox, Opera, Netscape) RA10.3. el sistema debe permitir una navegabilidad fácil a todas sus páginas RA10.4. El sistema abrirá los enlaces sobre el propio navegador
No conformidad	Problemas en el vínculo
Localización de la no Conformidad	Vínculo Quienes Somos
Sugerencias	Problema con diferentes link de la Web
Atributos asociados	Adaptabilidad (no conformidad) Adaptabilidad (sugerencia)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Compatibilidad, Sistema Operativo: Linux (Ubuntu Desktop 7.04),

	Navegador: Mozilla Firefox 2.0.0.3
Nombre del Probador	Carlos Manuel Hernández Vega
Nº	4
Elementos	Interfaz
Nombre del Requisito No Funcional	RA11. Usabilidad RA11.1. El sistema debe correr sobre cualquier plataforma (software libre o propietario) RA11.2. El sistema podrá correr en cualquier navegador (IE, FireFox, Opera, Netscape) RA11.3. el sistema debe permitir una navegabilidad fácil a todas sus páginas RA11.4. El sistema abrirá los enlaces sobre el propio navegador
No conformidad	Cuando se va a crear un reporte, si se da crear y no se especifica uno explota la aplicación
Localización de la no Conformidad	En: http://10.35.12.146:8080/SoporteTecnico1.2/pages/Crear%20Reporte.jsf A la hora de escoger un reporte, si se da crear y no se escoge ninguno. Ver Figura 2. Ver Figura 3.
Sugerencias	Validar esto tanto del lado del cliente usando JavaScript compatible con el DOM de Netscape y/o validar del lado del servidor también, para que no importe el cliente usado.
Atributos asociados	Adaptabilidad (sugerencia) Adaptabilidad (no conformidad)

3.3.1.6 Pruebas Rendimiento

Tabla 24 Resultados de las Pruebas de Sistema al Caso de Estudio Soporte Técnico

Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Rendimiento
Nombre del Probador	Yilennis Mendoza Torres
Nº	1
Elementos	Interfaz
Nombre del requisito No Funcional	RA1. Rendimiento RA1.1. Todas las respuestas del sistema a las operaciones serán menores de 5s RA1.2. El sistema debe permitir al menos 75 conexiones simultáneas con igual rendimiento para todos.

No conformidad	Empezó a las 5:35 PM las pruebas a la aplicación y siendo las 6:17 p.m. cuando se trata de refrescar la aplicación, esta no responde y sólo se muestra la página principal sin las opciones y solo hay 11 usuarios conectados simultáneamente, se demoró más del tiempo previsto en dar respuesta funcional a los probadores.
Localización de la no Conformidad	Aplicación
Sugerencias	Validar mejor los requisitos no funcionales sugeridos para la aplicación.
Atributos asociados	Comportamiento del tiempo(no conformidad)
Fecha	10 de mayo de 2007
Responsables	Yuniet Toll y Yilennis Mendoza.
Tipo de Prueba	Rendimiento
Nombre del Probador	Yilennis Mendoza Torres
Nº	2
Elementos	Interfaz
Nombre del Requisito No Funcional	RA2. Rendimiento RA2.1. Todas las respuestas del sistema a las operaciones serán menores de 5s RA2.2. El sistema debe permitir al menos 75 conexiones simultáneas con igual rendimiento para todos.
No conformidad	A mediado de las pruebas la aplicación se desconectó 2 veces, la aplicación solo contaba con 12 usuarios conectados concurrentemente.
Localización de la no Conformidad	Aplicación
Sugerencias	Validar mejor los requisitos no funcionales sugeridos para la aplicación.
Atributos asociados	Comportamiento del tiempo(no conformidad) Tolerancia de errores o fallos(no conformidad)

3.3.2 Representación Gráfica

Representación gráfica de los resultados obtenidos en las pruebas realizadas al Caso de Estudio Soporte Técnico según los siguientes parámetros: Distribución de Probadores, No Conformidades, Sugerencias, No Conformidad por Probador, Sugerencia por Probador.

Tabla 25 Datos para medir las pruebas

Tipo de Prueba	Probadores	No Conf.	Sug.	No Conf./Probadores	Sug/Probador
Funcional	8	29	12	3,63	1,50
Compatibilidad	2	3	4	1,50	2,00
Seguridad	1	2	2	2,00	2,00
Stress	1	2	1	2,00	1,00
Fiabilidad	1	3	3	3,00	3,00
Rendimiento	1	2	2	2,00	2,00
Totales	14	41	24	2,93	1,71

A continuación se muestra gráficamente la cantidad de probadores que participaron por cada tipo de prueba realizada.

Distribución de Probadores

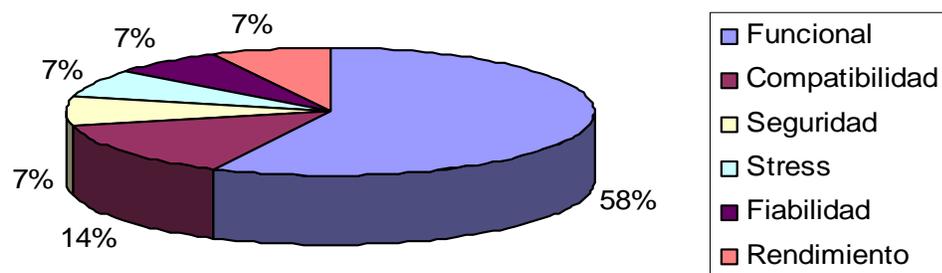


Fig. 14 Distribución de Probadores de Soporte Técnico

A continuación se muestra gráficamente la cantidad de no conformidades encontradas cada tipo de prueba realizada.

No Conformidades

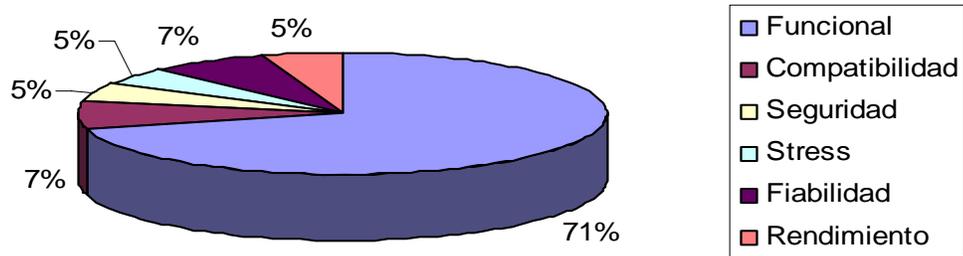


Fig. 15 No Conformidades

A continuación se muestra gráficamente a la cantidad de sugerencias por cada prueba realizada

Sugerencias

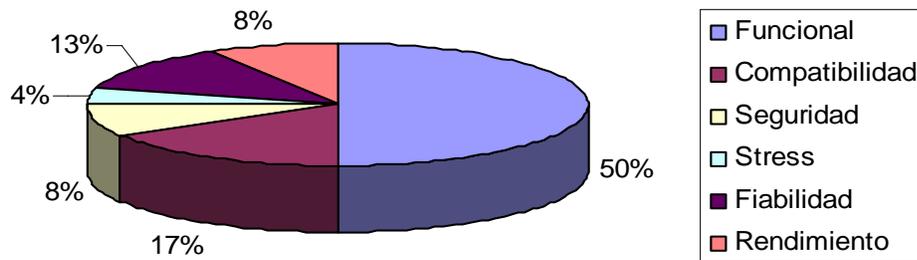


Fig. 16 Sugerencias

A continuación se muestra gráficamente la cantidad de no conformidades como promedio por cada probador de cada prueba realizada.

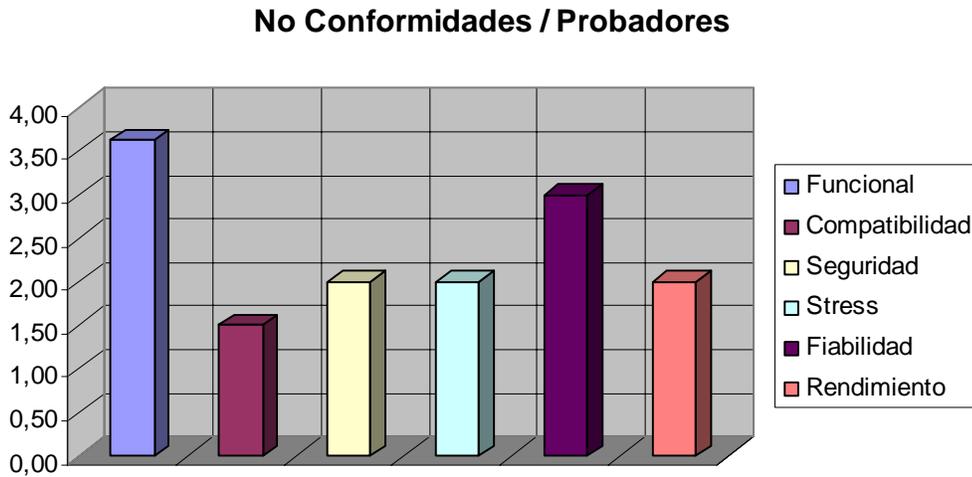


Fig. 17 No Conformidades por Probadores

A continuación se muestra gráficamente la cantidad de sugerencias como promedio por cada prueba realizada

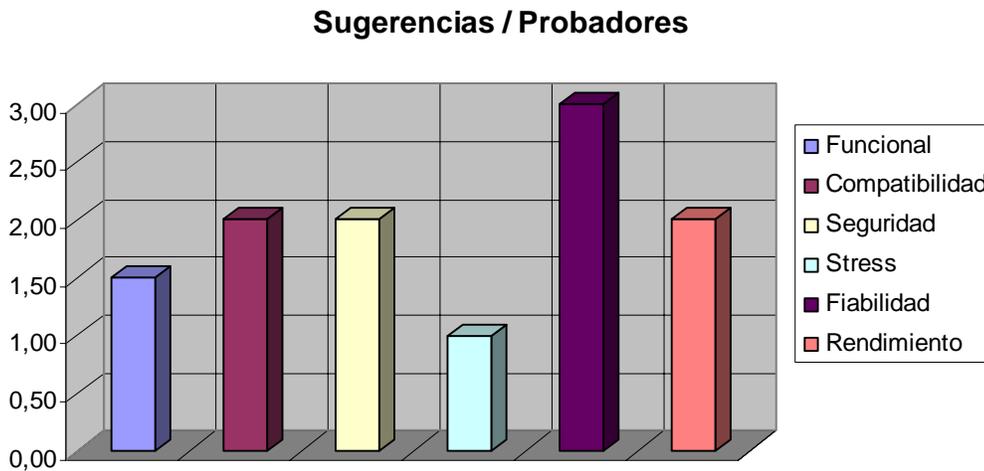


Fig. 18 Sugerencias por Probadores

3.3.3 Representación gráfica según estas subcaracterísticas o atributos de calidad

Representación gráfica de los resultados obtenidos en las pruebas realizadas al Caso de Estudio Ficha Técnica según los siguientes criterios o subcaracterísticas de calidad:

Basadas en la ISO 9126 se definió asociar las características y atributos a las pruebas de sistemas seleccionadas en este trabajo.

Tabla 26 Características y atributos de la ISO 9126 asociados a las pruebas de sistema

Características ISO 9126	Atributo	Asociadas a las Prueba de Sistema
Funcionalidad	Complejidad Consistencia Corrección Integridad Normalización	Funcional
Confiabilidad	Recuperabilidad Tolerancia de errores o fallos	Seguridad Disponibilidad y red Fiabilidad
Portabilidad	Adaptabilidad Instalabilidad	Compatibilidad
Eficiencia	Comportamiento del tiempo Comportamiento de los recursos.	Stress Rendimiento
Mantenibilidad	Diagnosticabilidad Concisión	Seguridad

Tabla 27 Métricas de calidad por tipo de Pruebas de Sistema y atributos asociados según la ISO 9126

Tipo de Prueba	Subcaracterísticas	No Conf.	Sug.	Puntuación	Escala
Funcionalidad	Complejidad	26	3	0,19	BAJO
	Consistencia	3	0	4,44	MEDIO
	Corrección	5	1	4,07	MEDIO
	Integridad	3	0	4,44	MEDIO
	Normalización	0	1	5,00	ALTO
Confiabilidad	Tolerancia de errores o fallos	4	0	4,26	MEDIO
	Recuperabilidad	2	1	4,63	ALTO
Eficiencia	Comportamiento del tiempo	2	0	4,63	ALTO
	Comportamiento de los recursos	1	0	4,81	ALTO
Mantenibilidad	Diagnosticabilidad	1	0	4,81	ALTO
Portabilidad	Adaptabilidad	3	2	4,44	MEDIO
	Instalabilidad	0	1	5,00	ALTO
	Totales	50	9	2,74	BAJO

La diferencia de cantidad de no conformidades entre ambos resultados mostrados en la tabla de las métricas y la de los datos para medir las pruebas se debe a la existencia de no conformidades que

cumplen con más de un atributo de calidad, tales como: en la prueba de rendimiento del probador Yilennis Mendoza la no conformidad cumple con los atributos tolerancia de errores o fallos y con el de comportamiento del tiempo, en la prueba funcional la no conformidad del probador Arlan Gálvez cumple con los atributos completitud, corrección, integridad y consistencia, en la prueba funcional los probadores Leonardo Santos y Luis Enrique Hernández las no conformidades cumplen con los atributos completitud y corrección, el probador Adonis Alea la no conformidad cumple con los atributos completitud e integridad, en la prueba de stress el probador Isis Blanco la no conformidad cumple con los atributos comportamiento del tiempo y comportamiento de los recursos, en la prueba de seguridad el probador Yander Cáceres la no conformidad cumple con los atributos tolerancia de errores o fallos y diagnosticabilidad, en la prueba de fiabilidad el probador Yilennis Mendoza la no conformidad cumple con los atributos recuperabilidad y tolerancia de errores y fallos.

Escala: es el por ciento que representa la no conformidad según el atributo o tipo de prueba que cumple en una escala del 0 al 5 donde Bajo está en el rango de 0 a 3,5, Medio está en el rango de 3,5 a 4,5 y Alto está en el rango de 4,5 a 5. Si se da el caso de que el valor obtenido está fuera del rango especificado la escala que toma es la que le continúa (ejemplo: el valor es 3,5 toma la escala siguiente, es decir, la escala de Medio).

La métrica “Puntuación”: es una métrica de evaluación de la calidad para las pruebas de sistemas (Castillo 2007). Se obtiene a través de una función basada en la ISO 9126 para medir las no conformidades que están asociados a un tipo de prueba y atributo de calidad, la función es: al Valor Máximo de la no conformidad encontrada según el tipo de prueba y atributo asociado ($VMNC$) se le resta el valor de la no conformidad encontrada asociada ($VNCE$) a ese tipo de prueba y atributo de calidad, el resultado de la resta se multiplica por 5 y este valor se divide por el Valor Máximo de la no conformidad encontrada según el tipo de prueba y atributo asociado ($VMNC$).

Ejemplo:

$$MP = \frac{(VMNC - VNCE) * 5}{VMNC}$$

$$MP = \frac{(27 - 26) * 5}{27}$$

$$MP = 0,19$$

Por tanto el caso de estudio Soporte Técnico se encuentra en la escala de bajo, por lo que presenta baja calidad.

En dependencia de la cantidad de no conformidades se determina si la calidad del producto es Baja, Media o Alta, ejemplo: en los resultados obtenidos del Caso de Estudio Soporte Técnico la calidad es Baja debido al alto número de no conformidades que cumplen con el atributo Completitud, es decir a mayor número de no conformidades menor calidad según el atributo que esté implicado y el tipo de prueba que se esté probando, esto no indica que la calidad del producto sea totalmente Baja, sino que para ese atributo en ese tipo de prueba es Baja la calidad, para esto hay que tener en cuenta cuanto repercute el atributo en el producto. El atributo Completitud es el que mide que en la aplicación estén implementadas todas las funcionalidades descritas por el cliente, si ocurre lo contrario como es el caso pues entonces la calidad del Caso de Estudio es totalmente Baja.

No Conformidad: es la cantidad de no conformidades encontradas por cada tipo de prueba sistema y atributos asociados a la misma.

Sugerencias: es la cantidad de sugerencias encontradas por cada tipo de prueba sistema y atributos asociados a la misma.

A continuación se muestra gráficamente la cantidad de no conformidades encontradas por tipo de prueba y atributos de calidad.

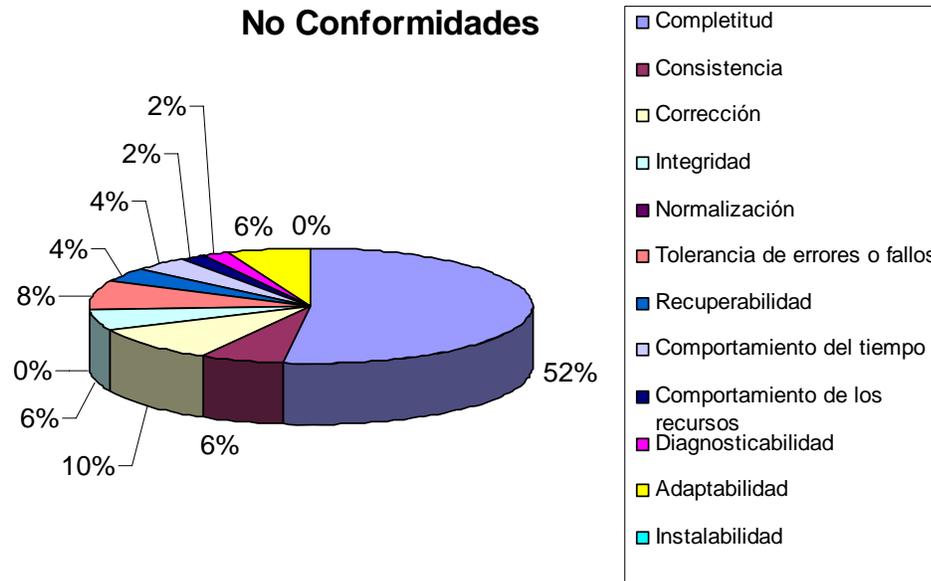


Fig. 19 No Conformidades por tipo de prueba y atributos de calidad

A continuación se muestra gráficamente la cantidad de sugerencias encontradas por tipo de prueba y atributos de calidad.

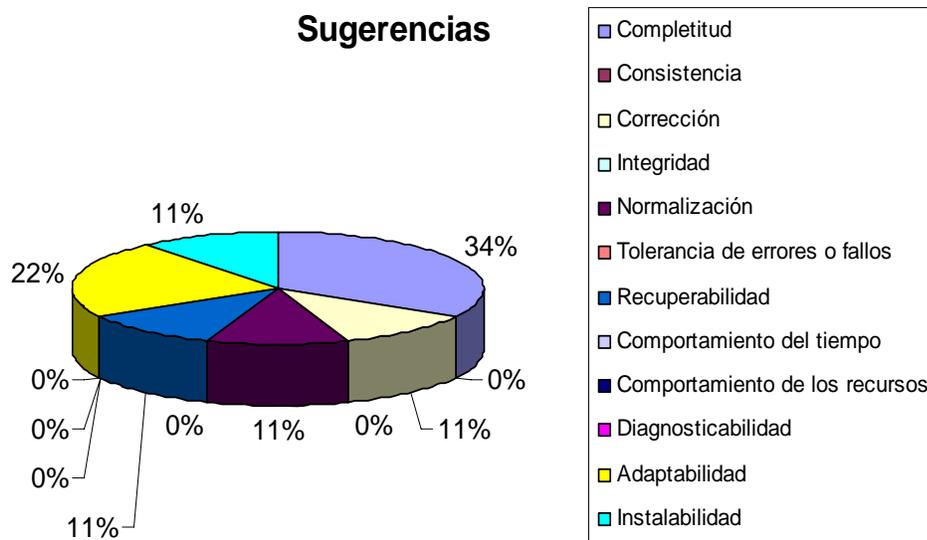


Fig. 20 Sugerencias por tipo de prueba y atributos de calidad

A continuación se muestra gráficamente la puntuación que obtiene dicha subcaracterística o atributo de calidad en una escala del 1 al 5 según con la no conformidad que cumpla.

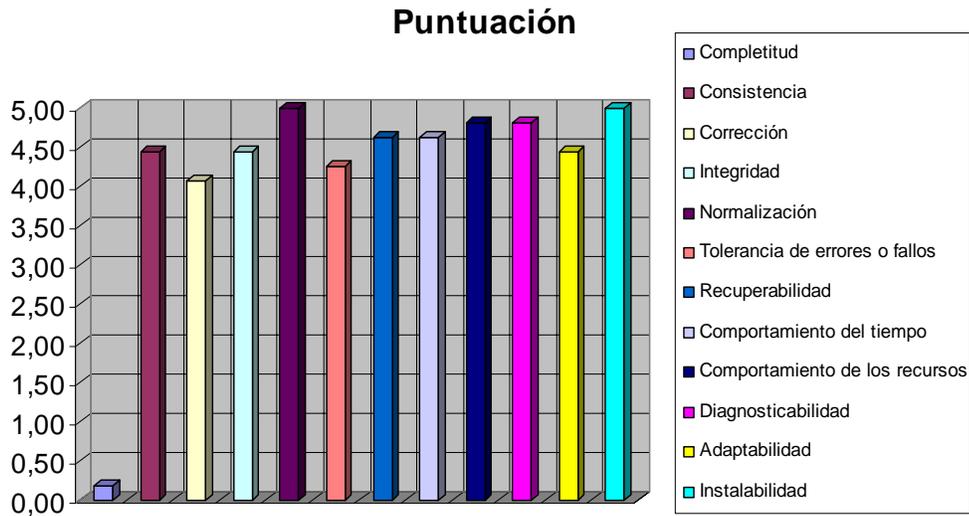


Fig. 21 Puntuación

A continuación se muestra gráficamente el por ciento que representa la no conformidad según el atributo o subcaracterística que cumple en una escala de 0 al 5 donde Bajo está en el rango de 0 a 3,5, Medio está en el rango de 3, a 4,5 y Alto está en el rango de 4,5 a 5.

Tabla 28 Escala

Esca	Cantidad
Bajo	1
Medio	5
Alto	6

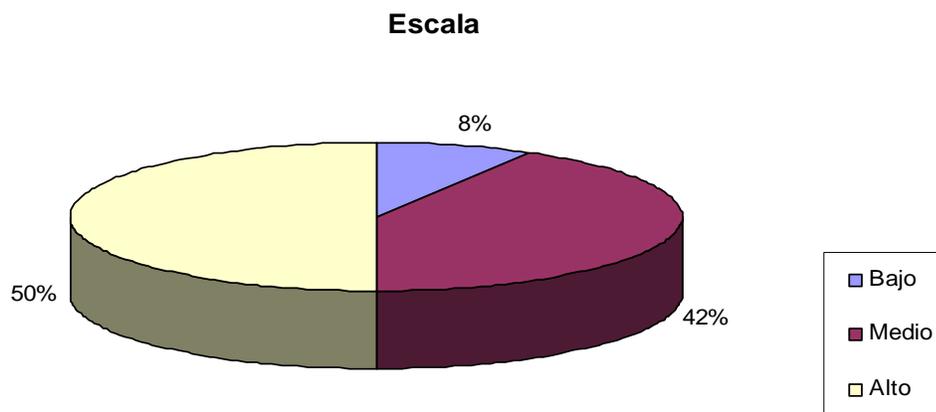


Fig. 22 Escala

3.3.4 Análisis de las pruebas realizadas.

Haciendo un análisis profundo de los resultados obtenidos en las pruebas realizadas a este Caso de Estudio se detectó que:

- ✓ Hubo que exigir que la aplicación estuviera disponible en el servidor, el cual era un servidor para JAVA, que solo admite 150 peticiones, al llegar a este límite de peticiones colapsaba el servidor mostrando un mensaje de error. Esto ocurría porque no contaban con un servidor propio sino que era común con la estación de trabajo. Esto traía como consecuencia que se perdiera toda la información obtenida hasta el momento teniéndose que empezar desde cero.
- ✓ Las pruebas se retrasaron debido a la no existencia de las analistas del Proyecto CICPC, pues no se contaba con la descripción correcta para llegar al caso de uso que se fuese a probar y tampoco se contaba con un mapa de navegación que explicara la navegabilidad de la aplicación.
- ✓ No se pudieron ejecutar todas pruebas previstas en la estrategia propuesta porque las analistas de sistema no estaban presentes y las responsables de pruebas tuvieron que hacer función del rol de analista y diseñadoras de los casos de pruebas, trayendo como consecuencia su retraso.
- ✓ No basta con una hora para preparar el entorno de laboratorio, se necesita más tiempo pues además de levantar las máquinas virtuales y la aplicación que se va a probar es necesario hacer algunas configuraciones a la máquina virtual para que se pueda ver la aplicación.
- ✓ Los administradores de la aplicación estuvieron todo el tiempo arreglando los problemas del sistema online.

- ✓ No se pudo contar con el equipo completo de calidad, cuando el laboratorio era de 30 PC para realizar las pruebas, pues solo estaban preparando el entorno del laboratorio un administrador de sistema del Proyecto CICPC y dos responsables de calidad.

3.4 Análisis de la distribución de los probadores por tipo de prueba en ambos casos de estudio.

Para seleccionar los probadores por tipo de prueba se tuvo en cuenta sus habilidades para desarrollar la actividad asignada. Además fue necesario tener presente que el probador no haya trabajado directamente en la aplicación sometida a prueba.

Para determinar la cantidad de probadores se tuvo en cuenta los objetivos de cada tipo de prueba definido en la propuesta. Algunos tipos de prueba poseen objetivos menos complejos y más específicos que otros tipos de prueba, por ejemplo, las pruebas de Compatibilidad tienen como objetivo verificar la funcionalidad del sistema en diferentes plataformas y navegadores; para este caso se determinó que con sólo 3 probadores bastaba para probar el sistema en los navegadores y en las plataformas requeridas. Sin embargo, para la prueba de Stress que persigue el cumplimiento de los requisitos no funcionales relacionados con las condiciones límites requeridas, fue necesario asignar una mayor cantidad de probadores.

En el caso de las pruebas funcionales se seleccionó la cantidad de probadores de acuerdo a la cantidad de casos de usos que tiene cada aplicación, pues por cada caso de uso se diseña un caso de prueba atendiendo al flujo básico, flujos alternos, escenarios, secciones y pantallas del caso de uso. Esto indica el volumen del contenido que hay que probar, por lo que se determinó asignar un caso de uso a un probador.

Para la prueba de Stress, Rendimiento, Fiabilidad, Disponibilidad y Red se selecciona la cantidad de probadores según la especificación del requisito no funcional para llevar a cabo esta prueba, al igual que se hizo para la prueba de Compatibilidad.

3.5 Análisis de los resultados obtenidos en ambos casos de estudio Ficha Técnica y Soporte

Técnico.

Ficha Técnica

Tabla 29 Métricas de calidad por tipo de Pruebas de Sistema y atributos asociados según la ISO 9126

Tipo de Prueba	Atributos	No Conf.	Sugerencias	Puntuación	Escala
Funcionalidad	Compleitud	30	7	0,16	BAJO
	Consistencia	3	6	4,52	ALTO
	Corrección	2	1	4,68	ALTO
	Integridad	0	2	5,00	ALTO
	Normalización	1	1	4,84	ALTO
Confiabilidad	Tolerancia de errores o fallos	8	2	3,71	MEDIO
Eficiencia	Comportamiento del tiempo	3	1	4,52	ALTO
	Comportamiento de los recursos	0	1	5,00	ALTO
Mantenibilidad	Concisión	0	1	5,00	ALTO
Portabilidad	Adaptabilidad	9	1	3,55	MEDIO
	Instalabilidad	5	1	4,19	MEDIO
	Totales	47	21	2,74	BAJO

Soporte Técnico

Tabla 30 Métricas de calidad por tipo de Pruebas de Sistema y atributos asociados según la ISO 9126

Tipo de Prueba	Subcaracterísticas	No Conf.	Sugerencias	Puntuación	Escala
Funcionalidad	Compleitud	26	3	0,19	BAJO
	Consistencia	3	0	4,44	MEDIO
	Corrección	5	1	4,07	MEDIO
	Integridad	3	0	4,44	MEDIO
	Normalización	0	1	5,00	ALTO
Confiabilidad	Tolerancia de errores o fallos	4	0	4,26	MEDIO
	Recuperabilidad	2	1	4,63	ALTO
Eficiencia	Comportamiento del tiempo	2	0	4,63	ALTO
	Comportamiento de los recursos	1	0	4,81	ALTO
Mantenibilidad	Diagnosticabilidad	1	0	4,81	ALTO
Portabilidad	Adaptabilidad	3	2	4,44	MEDIO
	Instalabilidad	0	1	5,00	ALTO
	Totales	50	9	2,21	BAJO

Métrica “Comparativa”: es una métrica de evaluación de las pruebas de sistemas y atributos. Se obtiene a través de una función basada en la ISO 9126 teniendo en cuenta las no conformidades que están asociados a un atributo de calidad y a un tipo de prueba. La función es: la sumatoria de la multiplicación de las no conformidades () por la métrica Puntuación () dividido por la cantidad total de las no conformidades ().

$$MC = \frac{\left(\sum_{i=1}^i (VNC * MP) \right)}{NC}$$

Resultado de Ficha Técnica

$$MC = \frac{\left(\sum_{i=1}^{11} \left((30 * 0,16) + (3 * 4,52) + (2 * 4,68) + (0 * 5) + (1 * 4,84) + (8 * 3,71) + (3 * 4,52) + (0 * 5) + (0 * 5) + (9 * 3,55) + (5 * 4,19) \right) \right)}{47}$$

$$MC = \frac{128,7}{47}$$

$$MC = 2,738 \approx 2,74$$

Resultado de Soporte Técnico

$$MC = \frac{\left(\sum_{i=1}^{12} \left((26 * 0,19) + (3 * 4,44) + (5 * 4,07) + (3 * 4,44) + (0 * 5) + (4 * 4,26) + (2 * 4,63) + (2 * 4,63) + (1 * 4,81) + (1 * 4,81) + (3 * 4,44) + (0 * 5) \right) \right)}{50}$$

$$MC = \frac{110,43}{50}$$

$$MC = 2,2086 \approx 2,21$$

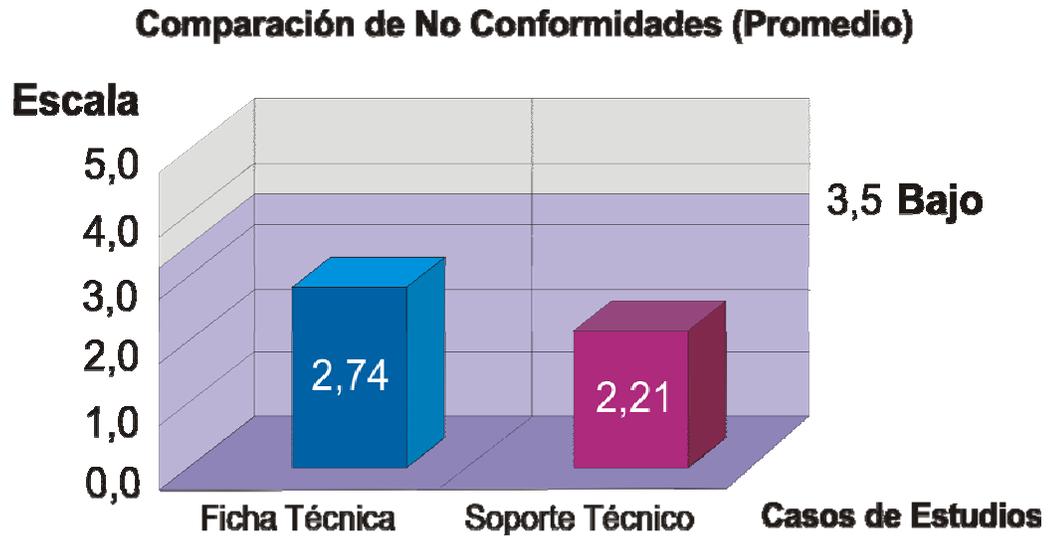


Fig. 23 Resultados generales

3.6 Conclusiones del capítulo

En este capítulo se recopilamos todos los resultados obtenidos de las pruebas realizadas a los dos Casos de Estudio Ficha Técnica y Soporte Técnico. Durante la ejecución de las pruebas las responsables de calidad se percataron de numerosas dificultades que provocaron retraso en la ejecución de las mismas tales como: falta de madurez y experiencia en la tarea, falta de organización, falta de apoyo, entre otras. Los resultados fueron recogidos en las respectivas plantillas acordadas y confeccionadas por las responsables de calidad, los mismos se obtuvieron de manera exitosa, luego fueron analizados, con el fin de entregárselos a los jefes de subsistema de cada uno de los Casos de Estudio del Proyecto CICPC probados. Ambas aplicaciones volvieron a la fase de desarrollo, pues deben cumplir con las normas de calidad establecidas en el Proyecto CICPC.

CONCLUSIONES

Finalizando este trabajo se llegaron a las siguientes conclusiones:

- ✓ Se demostró la importancia de contar con un manual de procedimiento para asegurar la máxima organización en la planificación, diseño, ejecución y evaluación de los resultados de las pruebas de sistema, evitando que el software sea rechazado al final del ciclo de desarrollo
- ✓ Se concluyó que la evaluación de las pruebas es un aspecto determinante para la producción, pues de esto depende medir hasta qué punto el software cumple con las especificaciones del cliente.
- ✓ Se diseñaron los casos de prueba necesarios según los requisitos funcionales y no funcionales de la aplicación, teniendo en cuenta las pruebas establecidas en la propuesta.
- ✓ Los resultados de las pruebas obtenidos en los Casos de Estudio Ficha Técnica y Soporte Técnico del Proyecto CICPC fueron favorables pues se pudo refinar y validar la propuesta.
- ✓ Se estableció un manual que contiene un conjunto de pasos secuenciales que orientan cómo proceder durante la realización de las pruebas de sistema en el Proyecto CICPC.

RECOMENDACIONES

- ✓ Se recomienda el estudio y utilización de herramientas para el control y seguimiento de las pruebas pues favorecen no solo la comprensión y análisis, sino que potencializan la mejora de la calidad producida.
- ✓ Se recomienda el estudio de herramientas de ejecución de pruebas de sistemas para realizarlas de manera automática, tales como: JMeter, Rational Robot, Rational Functional Testing, la Suite Green Volution compuesta por: la Green-Way, la Green --Test, la Green - Notes, y la Green – Metrics.
- ✓ Una vez terminada la Captura de Requisitos del Proyecto CICPC, se recomienda hacer un estudio detallado de los requisitos no funcionales del proyecto para detectar si se necesita incluir otro tipo de prueba no reflejada en esta propuesta.
- ✓ Se recomienda planificar un curso de capacitación a los probadores con el objetivo de prepararlos para la ejecución de las pruebas (cronograma de las pruebas, tipos de prueba, uso de las plantillas de casos de prueba y de la tabla de no conformidades).

REFERENCIAS BIBLIOGRÁFICAS

- Almunia, J. (2007) "Diseño de la interfaz de usuario. Pruebas del software." **Volumen**, DOI:
- Álvarez, J. C. G. (2000) "Grupo de Investigación de Lenguajes y Sistemas Informáticos e Ingeniería del Software de la Universidad de Granada " **Volumen**, DOI:
- Ambler, S. W. (2006) "Método de Pruebas Orientada a Objetos para el Ciclo de Vida Completo (FLOOT)." **Volumen**, DOI:
- Anna. C Grimán, M. P., Luis. E Mendoza (2007). "Estrategia de Pruebas para Software OO que garantiza Requerimientos No Funcionales."
- Antonio, A. d. (1999). GESTIÓN, CONTROL Y GARANTÍA DE LA CALIDAD DEL SOFTWARE.
- AVATAR (2007). "Fase de transición de RUP."
- Calero, D. C. (2003) "Modelos de calidad software Calidad y Medición de Sistemas de Información." **Volumen**, DOI:
- Castillo, L. L. (2007). Casos de Pruebas para Pruebas No Funcionales. Desoft S.A..
- Castillo, L. L. (2007). Métricas de evaluación de las pruebas de sistemas y atributos.
- Collado, M. (2003) "Pruebas de software. Técnicas de prueba del software. Estrategias de prueba del software." **Volumen**, DOI:
- Davis (1995). "Fase de Elaboración. FT Prueba (procedimientos genéricos y aplicación de algunos tipos de pruebas simples)."
- Desoft S.A., M. (2005). Conceptos de Disciplina de Pruebas.
- Edward (1996). "Fase de Elaboración. FT Prueba (procedimientos genéricos y aplicación de algunos tipos de pruebas simples)."
- Fernández, L. (1995) "Grupo de trabajo de Madrid. El grupo de trabajo de Calidad del Software de ATI Madrid." **Volumen**, DOI:
- Gutiérrez, J. J. (2005) "Generación de pruebas de sistema a partir de la especificación funcional." **Volumen**, DOI:
- Gutiérrez, J. J. (2006). "MODELOS DE PRUEBAS PARA PRUEBAS DEL SISTEMA." IEEE (1986). "estándar 1012."
- IEEE, S.-. (1990).
- Informática, I. I. N. d. E. e. (1997) "Pruebas de Sistema." **Volumen**, DOI:
- Ivar Jacobson, G. B., James Rumbaugh (2004). "El proceso unificado de desarrollo de software." **2**.
- Kendall, K. (1997). "Análisis y diseño de sistemas." **3ª edición**: 796- 800.
- López, N. S. (2006). "Solución Tecnológica Integral del Cuerpo de Investigaciones Científicas, Penales y Criminalísticas. Proyecto de Seguridad del CICPC. INFORME TECNICO SOFTWARE."
- Lovelle, J. M. C. (1999). "Calidad del Software Conferencia."
- Mañas, J. A. (1994). "Prueba de Programas."
- Métrica, M. (2007). "Construcción del Sistema de Información." **Versión 3**.
- Montero, Y. H. (2002). No Solo Usabilidad Journal **1**.
- Myers (1979). "Fase de Elaboración. FT Prueba (procedimientos genéricos y aplicación de algunos tipos de pruebas simples)."
- Myers, G. (2005). "Fase de Elaboración. FT Prueba (procedimientos genéricos y aplicación de algunos tipos de pruebas simples)."
- Pressman (2006). "Ingeniería de Software, un enfoque practico. DISEÑO DE CASOS DE PRUEBA."
- Pressman, R. S. (1995). "Ingeniería del software, un enfoque práctico, Mcgraw Hill."
- Pressman., R. S. (1998). "Ingeniería del software. Un enfoque práctico. McGrawHill".

- Pruebas (2007). "Desarrollo de Sistemas."
- Quesada, J. A. L. (2007). "Ingeniería del Software. Pruebas del software."
- Rubio, F. Ó. G. and C. B. Santos (2007) "Tema 7. Pruebas del Software Tema 9. Pruebas del Software." **Volumen**, DOI:
- RUP, P. T. (2003). "Rational Suite ".
- Salazar, J. P. (2004) "Calidad." **Volumen**, DOI:
- Software, C. d. a. D. d. I. y. G. d. (2005). "Fase de Elaboración. FT Prueba (procedimientos genéricos y aplicación de algunos tipos de pruebas simples)."
- Solutions, R.-T. T. (1996) "Security Testing." **Volumen**, DOI:
- Teruel, P. d. (2007). "Tema 5. Calidad de Software. Escuela Universitaria."
- Torres, M. C. (2006) "Modelos de Evaluación de Mejora." **Volumen**, DOI:
- Wikipedia (2006). "Fiabilidad de sistemas."
- Wikipedia, l. e. l. (2007). "Rational Unified Process."

BIBLIOGRAFÍA**1. 1976** (1)

Boehm, B. W., Brown, J.R., Lipow, M. (1976). Quantitative Evaluation of Software Quality, Proceedings 2nd International Conference on Software Engineering: 592-605.

2. 1978 (1)

Boehm, B. W., Brown, J.R., Lipow, M., Macleod, G.J., Merritt, M.J (1978). Characteristics of Software Quality.

3. 1979 (1)

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

4. 1981 (1)

Miller, E., Howden, W. E. (1981). "Tutorial, Software Testing & Validation Techniques, 2a ed., IEEE Computer Society Press".

5. 1986 (1)

IEEE (1986). "estándar 1012."

6. 1987 (1)

De Millo, R. A. e. a. (1987). "Software Testing and Evaluation."

7. 1990 (1)

IEEE, S.-. (1990).

8. 1994 (1)

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

9. 1995 (3)

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

Fernández, L. (1995) "Grupo de trabajo de Madrid. El grupo de trabajo de Calidad del Software de ATI Madrid." **Volumen**, DOI:

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

10. 1996 (4)

Edward (1996). "Fase de Elaboración. FT Prueba (procedimientos genéricos y aplicación de algunos tipos de pruebas simples)."

Solutions, R.-T. T. (1996). Automated Functional Testing.

Solutions, R.-T. T. (1996) "Scalability & Performance Testing Services." **Volumen**, DOI:

Solutions, R.-T. T. (1996) "Security Testing." **Volumen**, DOI:

- 11. 1997** (3)
De Domingo, J. y. A., A (1997) "Calidad y mejora continua." **Volumen**, DOI:
Informática, I. I. N. d. E. e. (1997) "Pruebas de Sistema." **Volumen**, DOI:
Kendall, K. (1997). "Análisis y diseño de sistemas." **3ª edición**: 796- 800.
- 12. 1998** (1)
Pressman., R. S. (1998). "Ingeniería del software. Un enfoque práctico. McGrawHill".
- 13. 1999** (3)
Antonio, A. d. (1999). GESTIÓN, CONTROL Y GARANTÍA DE LA CALIDAD DEL SOFTWARE.
Jiménez, A. d. A. (1999). "Calidad del Software, Apuntes de la Asignatura de Planificación y Gestión del Desarrollo de Sistemas Informáticos, Facultad de Informática, Universidad Politécnica de Madrid."
Lovelley, J. M. C. (1999). "Calidad del Software Conferencia."
- 14. 2000** (1)
Álvarez, J. C. G. (2000) "Grupo de Investigación de Lenguajes y Sistemas Informáticos e Ingeniería del Software de la Universidad de Granada " **Volumen**, DOI:
- 15. 2002** (2)
Dustin, E. (2002). Quality Web Systems. Performance, Security, and Usability. Addison Wesley.
Montero, Y.H., *No Solo Usabilidad journal* 2002.
- 16. 2003** (3)
Calero, D. C. (2003) "Modelos de calidad software Calidad y Medición de Sistemas de Información." **Volumen**, DOI:
Collado, M. (2003) "Pruebas de software. Técnicas de prueba del software. Estrategias de prueba del software." **Volumen**, DOI:
RUP, P. T. (2003). "Rational Suite".
- 17. 2004** (5)
Arenas, L. d. S. G. (2004) "METODOLOGÍA PARA EVALUAR LA CALIDAD DE LOS SISTEMAS DE INFORMACIÓN." **Volumen**, DOI:
Ivar Jacobson, G. B., James Rumbaugh (2004). "El proceso unificado de desarrollo de software." **2**.
Jacobson Ivar, B. G., Rumbaugh James (2004). "El proceso unificado de desarrollo de software "1.
Salazar, J. P. (2004) "Calidad." **Volumen**, DOI:
Telemáticos, D. d. I. d. S. (2004) "JUNIT. Pruebas Unitarias." **Volumen**, DOI:
- 18. 2005** (5)
Desoft S.A., M. (2005). Conceptos de Disciplina de Pruebas.
Gutiérrez, J. J. (2005) "Generación de pruebas de sistema a partir de la especificación funcional." **Volumen**, DOI:
Myers, G. (2005). "Fase de Elaboración. FT Prueba (procedimientos genéricos y aplicación de algunos tipos de pruebas simples)."
Software, C. d. a. D. d. I. y. G. d. (2005). "Fase de Elaboración. FT Prueba (procedimientos genéricos y

aplicación de algunos tipos de pruebas simples)."
 Latina, O. (2005) *Guía de JMeter*. **Volumen**,

19. 2006 (9)

Ambler, S. W. (2006) "Método de Pruebas Orientada a Objetos para el Ciclo de Vida Completo (FLOOT)."
Volumen, DOI:

Escalona, M. J. (2006) "MODELOS DE PRUEBAS PARA PRUEBAS DEL SISTEMA." **Volumen**, DOI:

Gutiérrez, J. J. (2006). "MODELOS DE PRUEBAS PARA PRUEBAS DEL SISTEMA."

Mejías, M. (2006). "MODELOS DE PRUEBAS PARA PRUEBAS DEL SISTEMA."

Reina, A. M. (2006). "MODELOS DE PRUEBAS PARA PRUEBAS DEL SISTEMA."

Torres, M. C. (2006) "Modelos de Evaluación de Mejora." **Volumen**, DOI:

Wikipedia (2006). "Fiabilidad de sistemas."

López, N.S., *Solución Tecnológica Integral del Cuerpo de Investigaciones Científicas, Penales y Criminalísticas. Proyecto de Seguridad del CICPC. INFORME TECNICO SOFTWARE*. 2006.

Pressman, *Ingeniería de Software, un enfoque practico. DISEÑO DE CASOS DE PRUEBA*. 2006.

20. 2007 (12)

Almunia, J. (2007) "Diseño de la interfaz de usuario. Pruebas del software." **Volumen**, DOI:

AVATAR (2007). "Fase de transición de RUP."

Castillo, L. L. (2007). Casos de Pruebas para Pruebas No Funcionales. Desoft S.A.

Métrica, M. (2007). "Construcción del Sistema de Información." **Versión 3**.

Wikipedia (2007) "Prueba unitaria." **Volumen**, DOI:

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

Anna. C Grimán, M.P., Luis. E Mendoza, *Estrategia de Pruebas para Software OO que garantiza Requerimientos No Funcionales*. 2007.

Pruebas, *Desarrollo de Sistemas*. 2007.

Quesada, J.A.L., *Ingeniería del Software. Pruebas del software*. 2007.

Rubio, F.Ó.G. and C.B. Santos (2007) *Tema 7. Pruebas del Software. Tema 9. Pruebas del Software*.

Volumen,

T.Koomen, M.P., *MEJORA DEL PROCESO DE PRUEBAS*. 2007.

Teruel, P.d., *Tema 5. Calidad de Software. Escuela Universitaria*. 2007.

Almunia, Joaquín (1)

Almunia, J. (2007) "Diseño de la interfaz de usuario. Pruebas del software." **Volumen**, DOI:

AVATAR (1)

AVATAR (2007). "Fase de Transición de RUP."

Castillo, Lazaro Luis (1)

Castillo, L. L. (2007). Casos de Pruebas para Pruebas No Funcionales. Desoft S.A.

Métrica, Metodología (1)

Métrica, M. (2007). "Construcción del Sistema de Información." **Versión 3**.

Wikipedia (1)

Wikipedia (2007) "Prueba Unitaria." **Volumen**, DOI:

Wikipedia, la enciclopedia libre (1)

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

Anna. C Grimán, María Pérez, Luis. E Mendoza (1)

Anna. C Grimán, M.P., Luis. E Mendoza, *Estrategia de Pruebas para Software OO que garantiza Requerimientos No Funcionales*. 2007.

Pruebas (1)

Pruebas, *Desarrollo de Sistemas*. 2007.

Quesada, Juan Antonio López (1)

Quesada, J.A.L., *Ingeniería del Software. Pruebas del software*. 2007.

Rubio, Félix Óscar García (1)

Rubio, F.Ó.G. and C.B. Santos (2007) *Tema 7. Pruebas del Software*
Tema 9. Pruebas del Software. Volumen,

Santos, Crescencio Bravo (1)

Rubio, F.Ó.G. and C.B. Santos (2007) *Tema 7. Pruebas del Software*
Tema 9. Pruebas del Software. Volumen,

T.Koomen, M. Pol (1)

T.Koomen, M.P., *MEJORA DEL PROCESO DE PRUEBAS*. 2007.

Teruel, Politécnico de (1)

Teruel, P.d., *Tema 5. Calidad de Software. Escuela Universitaria*. 2007.

1 GLOSARIO DE TÉRMINOS

- 2 1. **Adendas:** apéndice o conjunto de notas añadidas después de terminada una obra escrita para
3 aclarar, completar o rectificar su contenido.
- 4 2. **Calidad de Software:** conjunto de propiedades inherentes a un objeto que le confieren capacidad
5 para satisfacer necesidades implícitas o explícitas.
- 6 3. **Caso de Prueba:** conjunto de condiciones o variables bajo las cuáles el analista determinará si el
7 requisito de una aplicación es parcial o completamente satisfactorio.
- 8 4. **Caso de Uso:** es una técnica para especificar el comportamiento de un sistema: "es una secuencia
9 de interacciones entre un sistema y personas u otros sistemas de hardware y software que usa
10 alguno de sus servicios."
- 11 5. **Caso de Uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se
12 capta cuando se modela el negocio y se representa a través de los requerimientos.
- 13 6. **CICPC:** Cuerpo de Investigaciones Central Policial y Criminalísticas.
- 14 7. **CP:** Casos de Prueba.
- 15 8. **CU:** Casos de Uso
- 16 9. **Escala:** es el por ciento que representa la no conformidad encontrada según el tipo de prueba y
17 atributos de calidad en una escala de 0 al 5 donde Bajo está en el rango de 0 a 3,5, Medio está en
18 el rango de 3,5 a 4,5 y Alto está en el rango de 4,5 a 5.
- 19 10. **Estrategia de Prueba de Sistema:** es el conjunto de pasos que indican cómo desarrollar de forma
20 organizada las pruebas de sistema a un software o aplicación determinada.
- 21 11. **Flooding:** Se usa para designar un comportamiento abusivo de la red de comunicaciones,
22 normalmente por la repetición desmesurada de algún mensaje en un corto espacio de tiempo.
- 23 12. **Hardware:** es el conjunto de elementos materiales que componen una computadora.
- 24 13. **Manual de Procedimiento:** es el conjunto de pasos lógicos ordenados cronológicamente que
25 indican cómo proceder en determinado proceso.
- 26 14. **Métrica:** es el término que describe muchos y muy variados casos de medición.
- 27 15. **Pruebas de Sistema:** son las que se le realizan al software una vez que se termina la fase de
28 construcción.
- 29 16. **Ralentiza:** sinónimo de lentitud.

- 1 17. **Servlets Java:** Los servlets son objetos que corren dentro del contexto de un contenedor de
2 servlets (Ej. Tomcat) y extienden su funcionalidad.
- 3 18. **Sistema de Software:** es el conjunto de programas que administra los recursos de hardware.
- 4 19. **Software:** es el conjunto de programas y procedimientos necesarios para hacer posible la
5 realización de una tarea específica, en contraposición a los componentes físicos del sistema.
- 6 20. **Suite:** Conjunto de pruebas.
- 7 21. **WYSWYG:** significa: “what you see is what you get” - lo que ves es lo que obtienes. Permiten
8 editar o crear página Web en un modo visual.