

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**  
**Facultad 7**



**Propuesta de Procedimiento para la  
Revisión y Evaluación de un  
Producto de Software Terminado**

Trabajo de diploma para optar por el título de  
Ingeniero en Ciencias Informáticas

**Autoras:** Mireilys Martínez Miranda.  
Yelenys Delvoys Marchante

**Tutores:** Ing. Regla María Silva Calderón  
Ing. Katia Hurtado Duvergel

Ciudad de la Habana, Julio de 2007

## **DECLARACION DE AUTORÍA**

---

Declaramos ser autoras 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 firmamos la presente a los \_\_\_\_ días del mes de Junio de 2007.

### **Autoras:**

\_\_\_\_\_  
**Mireilys Martinez Miranda**

\_\_\_\_\_  
**Yelenys Delvoys Marchante**

### **Tutoras:**

\_\_\_\_\_  
**Lic. Regla María Silva Calderón**

\_\_\_\_\_  
**Ing. Katia Hurtado Duvergel**

## *Agradecimientos.*

---

*Deseamos al concluir este trabajo de tesis agradecer a las personas que han contribuido a nuestra formación profesional y a aquellos que nunca han dejado de darnos su incondicional apoyo y confianza.*

*A nuestra tutora **Lic. Regla María Silva Calderón** por acogernos y confiar en nosotras, por darnos el aliento necesario para continuar, por su apoyo, sus criterios y su incalculable ayuda, por ser ejemplo a seguir y meta a alcanzar.*

*A la **Ing. Katia Hurtado Duvergel** por dedicarnos parte de su tiempo a encontrar las palabras que aun faltaban.*

*A la **Ing. Lourdes Escalona Peral** por estar ahí justo cuando la necesitamos.*

*A **Yaimi Márquez** por preguntar y entender, por ser la mejor compañera de estudio que hemos tenido y por todos los desvelos que hemos pasado en estos años.*

*A **Eraclides Vicet** y **Anabel Díaz** por su apoyo incondicional y estar siempre presentes.*

*A **Maritza Ortiz** por su profesionalidad y aporte de conocimientos muy valiosos en la revisión de este trabajo de diploma.*

*A todos los **Especialistas de SOFTEL** que colaboraron con el desarrollo de esta tesis con sus aportes de conocimiento, visión crítica y objetividad.*

*A todos **nuestros amigos**, por preocuparse, por preguntar, por reír juntos, por las charlas.*

*A todos aquellos que de una forma u otra han hecho posible que hayamos llegado hasta aquí.*

*A todos los que se acercaron a preguntar, a indagar por el estado de la tesis.*

**Gracias a todos**

A mi **Mamá** por haberme querido tanto y estar siempre a mi lado, por apoyarme durante todos estos años, por escucharme y comprenderme, por ser un ejemplo a seguir.

A mi **Hermano** por ser mi alegría y mi orgullo.

A mis **Abuelos** por estar siempre al tanto.

A **Manolito** por estar pendiente y presente, en las buenas y en las malas, por amarme y quererme.

A mi **papá** por quererme en silencio.

A **Fran** por su apoyo siempre oportuno.

A **Josefa** por su cariño y sus consejos en el momento preciso.

A **Olga** por el amor que me ha brindado, por sus buenos consejos y por estar junto a mí.

**Mireifys**

A mi **Mamá** por haberme impulsado a seguir cuando me faltaba el aliento y convencerme de que era posible, por el cariño, el amor, la confianza, la honestidad, el ejemplo, que me has transmitido durante toda la vida, principalmente en los momentos más difíciles.

A **Pepe** por darme su amor y apoyarme en TODO para alcanzar este sueño.

A mi **Papá** que se sentiría muy orgulloso de verme en este momento.

A mi **Hermano** por estar siempre en mi corazón.

A mis **Abuelos** por la confianza, la preocupación, el apoyo, los desvelos.

A mi **Tía** por su optimismo y confianza, por entenderme y quererme y por su constante preocupación.

A **Yorlan** por ser compañero, amigo, novio, por todo el amor y el apoyo que me ha brindado siempre.

A mi **familia** en general por estar siempre atentos a mis resultados en la escuela.

**Yelenys**

## *Dedicatoria.*

---

*A mi mamá y a mi hermano.*

*A mis abuelos.*

*A Manolito.*

*Mireilys*

*A mi mamá y a mi hermano.*

*A Pepe.*

*A mis abuelos.*

*A Yorlan.*

*Yelenys*

*A nosotras.*

## Resumen

---

En el proceso de desarrollo de software se han evidenciado significativas dificultades en la calidad de los productos. La prueba del software constituye una garantía de calidad del mismo y representa una revisión final de las especificaciones, diseño y codificación. Por lo que deben realizarse pruebas minuciosas y bien planificadas.

Actualmente en el Laboratorio de Calidad de la Facultad 7 de la Universidad de las Ciencias Informáticas las revisiones a los software que les son entregados no se efectúa de forma eficiente. Esta investigación propone un procedimiento a seguir en el mismo que permite garantizar la calidad del proceso de revisión y evaluación de los productos de software, lo que posibilitará que el producto final, una vez revisado, posea un mejor índice de calidad en relación a las pruebas realizadas hasta la actualidad.

Se decidió tomar la norma cubana ISO/IEC 9126-1:2005, debido a que es la que más se ajusta a la situación abordada, define las características a evaluar en un sistema que ya esté terminado y ofrece la flexibilidad al que la usa, de definir sus propios métodos evaluativos. Utilizándola, se conformaron los Cuestionarios de Evaluación del software para ser usados por los miembros del equipo evaluador y de esta forma mejorar la situación actual existente en el laboratorio.

Con la aplicación de este procedimiento se espera obtener una mejor estructura organizativa del personal del Laboratorio de Calidad de la Facultad 7 y mejorar la seguridad y confiabilidad de los productos revisados en el mismo.

# Índice

---

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1 . FUNDAMENTOS TEÓRICOS</b> .....	<b>7</b>
1.1 CALIDAD Y DESARROLLO DE SOFTWARE .....	8
1.2 MARCO CONCEPTUAL.....	10
1.2.1 <i>Calidad</i> .....	10
1.2.2 <i>Calidad del software</i> .....	10
1.2.3 <i>Gestión de la calidad del software</i> .....	10
1.2.4 <i>Sistema de calidad</i> .....	10
1.2.5 <i>Prueba</i> .....	10
1.2.6 <i>Prueba del Software</i> .....	11
1.2.7 <i>Laboratorio de calidad</i> .....	11
1.2.8 <i>Normas ISO</i> .....	11
1.3 OBJETO DE ESTUDIO.....	12
1.3.1 <i>Objetivos estratégicos de la organización</i> .....	12
1.3.2 <i>Flujo actual de los procesos</i> .....	13
1.3.3 <i>Análisis crítico de la ejecución de los procesos</i> .....	14
1.4 FUNDAMENTACIÓN DE LOS OBJETIVOS.....	14
1.5 ANTECEDENTES DEL TEMA.....	15
1.6 TENDENCIAS Y TECNOLOGÍAS ACTUALES.....	19
1.6.1 <i>Estándares de calidad</i> .....	19
1.6.2 <i>Normas ISO 9000</i> .....	19
1.6.3 <i>La calidad del Software dentro de la serie ISO/IEC 9001:2000</i> .....	20
1.6.4 <i>NC ISO/IEC 9126</i> .....	22
1.6.5 <i>Marco actual de las pruebas de software</i> .....	27
1.6.6 <i>Principios de las pruebas de software</i> .....	27
1.6.7 <i>Métodos de prueba</i> .....	28
1.6.8 <i>Tipos de prueba</i> .....	31
1.6.9 <i>Otros tipos de pruebas</i> .....	32
1.6.10 <i>Depuración (debugging)</i> .....	32
1.6.11 <i>Plan de Pruebas</i> .....	33
1.7 ANÁLISIS CRÍTICO DE LAS FUENTES Y BIBLIOGRAFÍAS UTILIZADAS .....	34

1.8	CONCLUSIONES PARCIALES .....	35
<b>CAPÍTULO 2 . PROCEDIMIENTO PROPUESTO.....</b>		<b>36</b>
2.1	PROPÓSITO.....	37
2.2	ALCANCE.....	37
2.3	DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS.....	37
2.4	REQUERIMIENTOS DE ORGANIZACIÓN DEL LABORATORIO .....	38
2.4.1	<i>Estructura Organizativa .....</i>	<i>38</i>
2.4.2	<i>Equipamiento del Laboratorio (Software y Software).....</i>	<i>42</i>
2.4.3	<i>Condiciones de Seguridad del Laboratorio para el Producto.....</i>	<i>43</i>
2.5	ACCIONES A LLEVAR A CABO EN EL PROCESO .....	44
2.5.1	<i>Planificación y Preparación de la Revisión Técnica.....</i>	<i>44</i>
2.5.2	<i>Estrategia de Verificación.....</i>	<i>48</i>
2.5.3	<i>Evaluación del Software .....</i>	<i>57</i>
2.6	CONCLUSIONES PARCIALES .....	66
<b>CAPÍTULO 3 . ANÁLISIS COMPARATIVO ENTRE EL PROCESO DE REVISIÓN ACTUAL Y LO QUE SE PROPONE. ....</b>		<b>67</b>
3.1	ESTRUCTURA ORGANIZATIVA.....	68
3.2	DOCUMENTACIÓN.....	68
3.2.1	<i>Documentos generados y no generados durante el proceso de revisión del producto.....</i>	<i>69</i>
3.3	ENTRADA DEL PRODUCTO AL LABORATORIO.....	71
3.4	SEGURIDAD DEL PRODUCTO DENTRO DEL LABORATORIO .....	71
3.5	HARDWARE Y SOFTWARE.....	72
3.6	ESTRATEGIA DE VERIFICACIÓN.....	72
3.7	EVALUACIÓN DE LAS PRUEBAS.....	77
3.8	CONCLUSIONES PARCIALES.....	79
<b>CONCLUSIONES .....</b>		<b>80</b>
<b>RECOMENDACIONES .....</b>		<b>81</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>		<b>82</b>
<b>BIBLIOGRAFÍA .....</b>		<b>83</b>
<b>ANEXOS .....</b>		<b>84</b>
<b>GLOSARIO DE TÉRMINOS Y ACRÓNIMOS .....</b>		<b>107</b>

## *Índice de Figuras.*

---

Figura 1. Evolución Histórica de la Calidad .....	8
Figura 2. Gestión de la calidad.....	9
Figura 3. Características de la Calidad durante el Uso .....	23
Figura 4. Características y subcaracterísticas de la Calidad Interna y externa .....	24
Figura 5. Relación entre Calidad externa, Calidad Interna y Calidad durante el Uso.....	26
Figura 6. Pruebas de Caja Negra.....	28

## Introducción

Desde el surgimiento de las Tecnologías de la Informática y de las Comunicaciones (TICs) son muchos los países que han optado por explotar sus potencialidades en todas las esferas de la vida, tanto sociales, económicas como políticas, tomando como base el desarrollo que estas traen consigo, así como las facilidades de trabajo, ahorro de personal, tiempo y recursos. Son estos factores los que persiguen todas aquellas personas, instituciones e incluso naciones que tienen como objetivo desarrollar su economía teniendo como base la informatización de todos los procesos que en ellas se desarrollen.

Cuba, es un ejemplo fehaciente de país que pone en práctica esta doctrina, caracterizándose por sus esfuerzos para lograr productos con calidad. El proceso de informatización en sus inicio fue lento debido a la situación económica por la que atravesaba el país y abarcó aquellas esferas de mayor auge y productividad, que le permitiera al país un mayor desarrollo a la vez de contribuir a la informatización de otras esferas.

Hoy por hoy se ha erguido ante esta situación y avanza a la cabeza de los países tercermundistas, dando pasos agigantados en cuanto al avance tecnológico e informático en esferas tan importantes como la medicina, la educación, la economía, el turismo y otros sectores que también aportan al desarrollo económico del país.

Se pueden mencionar algunos ejemplos concretos que demuestran los resultados obtenidos producto de este avance, entre ellos se destacan el Centro de Inmunológica Molecular que produce vacunas y antídotos a enfermedades nunca antes contrarrestadas, para consumo nacional y la exportación; la creación de un red de informática médica (INFOMED) donde se puede encontrar diversa información relacionada con temas de salud nacional e internacional; y la más reciente creación de la Revolución Cubana que es la Universidad de las Ciencias Informáticas (UCI), centro especializado en la formación de ingenieros informáticos con un alto nivel de preparación capaces de producir software con calidad.

Todo este desarrollo ha favorecido la producción y comercialización de productos así como la prestación de servicios, requiriendo los mismos una calidad que le permita su continuo comercio y explotación, además que aporta una opinión favorable a los productos nacionales. Es por esta razón, que el tema calidad está siendo muy difundido en todas las esferas, ya que la situación actual que se está viviendo en el mercado es el abarrotamiento de productos, favoreciéndose en la venta solo aquellos productores que exponen productos de alta calidad.

Garantizar la calidad del software en todas las esferas de la sociedad es una meta internacional, donde muchas empresas actualmente están enfrascadas en esta tarea, con el fin de lograr productos eficientes o capaces de satisfacer las necesidades del cliente.

En el caso de la producción de software la calidad no se reduce a garantizar la calidad del producto sino a garantizar también la calidad del proceso tanto de desarrollo como del mantenimiento del producto.

Normalmente en muchas empresas del mundo, incluyendo las de Cuba se hacen las pruebas que garantizan la calidad de los productos terminados y no precisamente durante el ciclo de vida del software, por lo que se debe ser exigente con las evaluaciones y revisiones de los productos terminados sobre todo cuando dichos software se evalúan en un laboratorio

En Cuba gracias a “los ingentes esfuerzos del Estado Socialista por transitar aceleradamente hacia la Informatización de la Sociedad Cubana como vía para aumentar la calidad de vida, la eficiencia y la competitividad del país, garantizando la estabilidad, confiabilidad, vitalidad, seguridad e inviolabilidad de las nuevas tecnologías”[1], se cuenta con una importación y exportación de productos y servicios con CALIDAD.

La UCI tiene un incremento considerable en la producción de software, tanto para el consumo nacional como para la exportación. Por tanto se ha visto en la necesidad de formar pequeños grupos de trabajo con el objetivo de revisar detalladamente los software que se produzcan dentro de cada facultad, para de esta forma facilitar el trabajo en el Laboratorio de Certificación de la Universidad (CALISOFT).

Estos grupos actualmente están llevando cada uno un proceso de calidad diferente en algunos casos normalizados y en otros no, por lo que no se están rigiendo por una norma, estándar o general que les permitan realizar un trabajo eficiente.

CALISOFT recibe software ya terminados y previamente revisados por el grupo de calidad que tiene cada una de las diez facultades con que cuenta el centro de altos estudios. Cada Facultad produce software regidos por un perfil determinado tales como: salud, procesamiento digital de imágenes y señales, televisión, software libre, software educativo, entre otros.

Actualmente en la Facultad 7 existe un Laboratorio de Calidad que tiene como objetivo: *revisar productos de software terminados desarrollados en la misma*, sin embargo se ha detectado que:

- ✓ Este laboratorio no tiene una organización estructural óptima para la revisión de software.
- ✓ En el laboratorio no existe un procedimiento que rija todo el proceso de evaluación de un producto de software terminado.
- ✓ No se aplican los métodos de seguridad informática y protección del software.
- ✓ No existe una norma que garantice la revisión y evaluación del software con la calidad requerida.
- ✓ No cuenta con los requisitos necesarios para lograr implantar el procedimiento que se desea proponer.
- ✓ Desde el punto de vista del equipo de trabajo no están definidos los roles que facilitan el desempeño satisfactorio de la tarea que se quiere realizar.
- ✓ No hay criterio de organización para repartir las responsabilidades entre los integrantes del equipo de calidad y por tanto se hace imposible el control y evaluación del cumplimiento de dichas tareas.
- ✓ Existe poca comunicación entre los desarrolladores y el equipo de trabajo.

La no existencia de un procedimiento basado en una norma que permita facilitar y avalar la revisión y evaluación del software dentro del laboratorio es una cuestión a la que se le debe dar solución para lograr ordenar y hacer más viable el trabajo dentro del mismo.

Producto de toda esta situación se parte desde un principio por implementar un laboratorio que responda a las exigencias que plantea el procedimiento.

Luego de analizar las dificultades que existen en el Laboratorio de Calidad de la Facultad 7 se identifica como **problema científico**: ¿Cómo garantizar la calidad del proceso de revisión y evaluación de los productos de software en el laboratorio de la Facultad 7 de la Universidad de las Ciencias Informáticas?

Este problema se enmarca en el **objeto de estudio**: Proceso de revisión y evaluación de los productos de software en la Universidad de las Ciencias Informáticas.

El **campo de acción** abarcado es: Proceso de revisión y evaluación de los productos informáticos en el Laboratorio de Calidad de la Facultad 7 de la Universidad de las Ciencias Informáticas.

### **Objetivo general de la investigación:**

Desarrollar un procedimiento que permita garantizar la calidad del proceso de revisión y evaluación de los productos de software en el Laboratorio de Calidad de la Facultad 7 de la Universidad de las Ciencias Informáticas.

Para encaminar la investigación en vista a resolver el problema planteado se propone la siguiente **idea a defender**: El desarrollo de un procedimiento estándar y flexible, basado en la Norma Cubana ISO/IEC 9126-1: 2005, garantizará la calidad en la revisión y evaluación de los productos de software en el Laboratorio de Calidad de la Facultad 7 de la Universidad de las Ciencias Informáticas.

Las **tareas** que se llevarán a cabo para darle cumplimiento al objetivo trazado son:

1. Analizar el estado del arte relacionado con las pruebas que se le realizan a los productos informáticos en los Laboratorios de Calidad.
2. Investigar cómo se lleva a cabo el proceso de trabajo en el Laboratorio de Calidad de la Facultad 7.
3. Analizar las diferentes normas de calidad del software que existen.
4. Evaluar la norma ISO/IEC 9126-1: 2005 para aplicarla en el proceso de evaluación.
5. Establecer los principios de trabajo del Laboratorio de Calidad, para la ejecución de las pruebas al producto de software.

6. Diseñar un procedimiento que garantice estructural y funcionalmente el proceso de revisión y evaluación de los productos de software en el Laboratorio de Calidad de la Facultad 7.
7. Realizar un análisis comparativo entre el flujo actual de los procesos y lo propuesto en el procedimiento.

Cuando se lleva a cabo una investigación se deben de tener en cuenta las herramientas y los métodos para llevarla a cabo.

En esta investigación se hace uso de los métodos teóricos tales como el **analítico – sintético**, con el objetivo de analizar las teorías, documentos, e investigaciones anteriores relacionadas con el tema a tratar, permitiendo la extracción de los elementos más importantes que se relacionan con el objeto de estudio; y el **histórico – lógico**, mediante el cual se obtuvo toda la información relacionada con las normas de calidad, estándares internacionales, la certificación del software, las normas de calidad, así como los antecedentes del tema que se trató.

Dentro de los métodos empíricos se utilizó la **observación**, que posibilitó la visualización de la situación real de lo que ocurre en estos momentos en el Laboratorio de Calidad de la Facultad 7, permitiendo investigar el fenómeno en su manifestación externa; y la **entrevista** con el objetivo de obtener información de algunas personas especialidades en el tema de calidad en la Universidad.

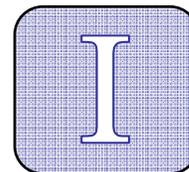
Este trabajo consta de Introducción, 3 capítulos, Conclusiones, Recomendaciones, Referencias bibliográficas, Glosario de términos y Anexos.

En el Capítulo 1 **Fundamentación Teórica** se abordan los aspectos relacionados con el objeto de estudio y campo de acción, los conceptos principales asociados al dominio del problema, y procedimientos propuestos anteriormente. Además se muestra un estudio detallado sobre la puesta en práctica a nivel mundial de normas de calidad relacionadas con la producción y comercialización del software, cuyo resultado será la base del desarrollo de la presente investigación.

En el Capítulo 2 **Procedimiento Propuesto** se muestra un procedimiento que servirá de patrón para implantar las condiciones necesarias del Laboratorio y para que el grupo de trabajo de Calidad de la

Facultad 7 pueda guiar su labor, para que la misma se realice con unanimidad, organización, eficiencia, facilitando así el trabajo del Grupo Central de Calidad (CALISOFT).

En el Capítulo 3 **Análisis comparativo entre el proceso de revisión actual y el que se propone** se realiza una comparación entre el proceso llevado a cabo actualmente en el Laboratorio de Calidad de la Facultad 7 de la Universidad de las Ciencias Informáticas y lo que realmente se debe hacer según el procedimiento propuesto en este trabajo de diploma para que esta revisión tuviera la calidad necesaria.



---

## Capítulo 1. Fundamentos Teóricos.

En este capítulo se realiza un análisis profundo de algunos temas relacionados con la calidad del software y de los métodos de prueba de caja blanca y caja negra que se utilizan para la revisión de un sistema informático.

Se investiga o evalúa todo el proceso de trabajo en el Laboratorio de Calidad de la Facultad 7 de la Universidad de las Ciencias Informáticas así como las tendencias y tecnologías actuales de estas temáticas en Cuba y el resto del mundo profundizando en las Normas ISO establecidas para el control de la calidad, sus características y especificidades con el objetivo de encaminar la investigación en la más apropiada de estas para dar solución al problema planteado.

Para concluir se hace un análisis crítico de todas las fuentes bibliográficas utilizadas durante la realización de este trabajo investigativo.

## 1.1 Calidad y desarrollo de Software

“La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia” [2]. Un software elaborado para ser explotado durante un largo período, necesita ser confiable, mantenible y flexible para disminuir los costos de mantenimiento y perfeccionamiento durante el tiempo de explotación.

A lo largo de la historia el concepto de Calidad ha estado estrechamente vinculado e indisolublemente con el de Desarrollo de Software, es por ello que cuando se habla de uno se habla del otro o se hace referencias a otro y a la evolución histórica de todo el proceso relacionado con la mejora de los sistemas informáticos y la calidad de los mismos. (Ver figura 1)

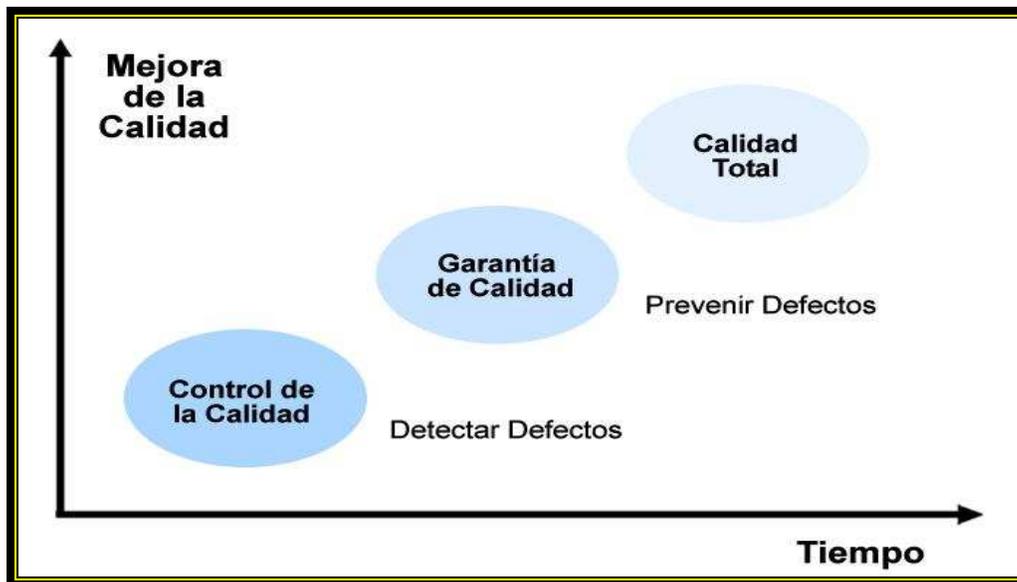


Figura 1. Evolución Histórica de la Calidad

Cuando se va a producir un software es muy importante tener en cuenta los costes que se van a generar como parte de todo el proceso de calidad asociado al mismo. La gestión de la calidad es el único instrumento adecuado para evitar un exceso de los gastos por falta de calidad del producto y de los procesos de desarrollo y de mantenimiento, y para poder decidir de forma responsable si un

sistema es apto para su uso o no. La gestión, y como parte de ella el control de la calidad, no es exclusivamente la responsabilidad de los que desarrollan un sistema software. Sus clientes, es decir, los usuarios finales, tienen un papel incluso más importante en este proceso. En la situación óptima el equipo de calidad está compuesto por representantes de ambos partidos. (Ver figura 2)

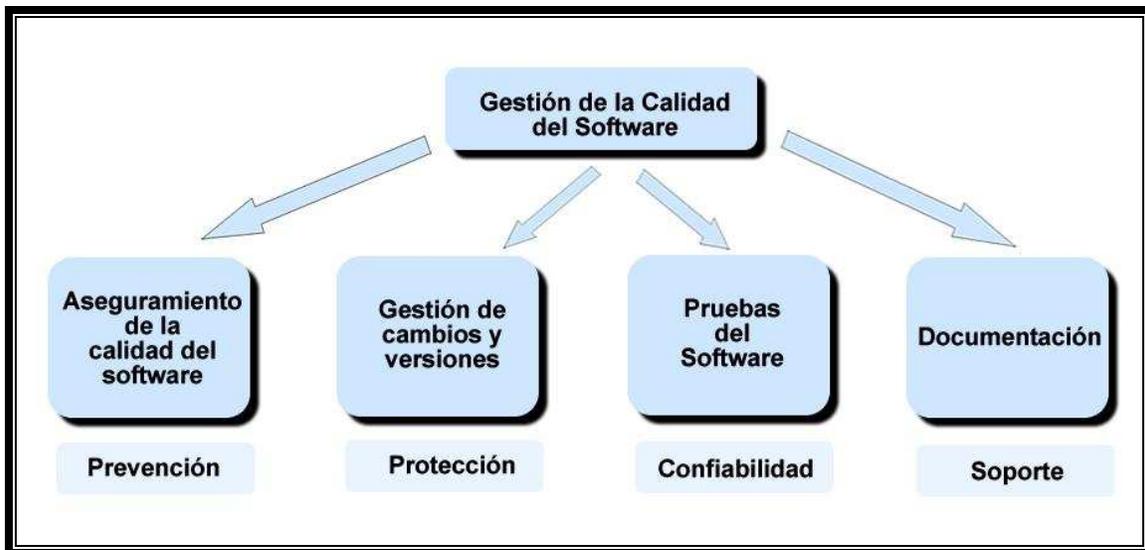


Figura 2. Gestión de la calidad

La adopción de una buena política contribuye en gran medida a lograr la calidad del software, pero no la asegura. Para el aseguramiento de la calidad es necesario su control, evaluación y auditoría.

Desde que aparecieron los primeros grandes sistemas informáticos se incluyó a nivel metódico e imprescindible la relación calidad – pruebas, términos muy dependientes entre sí que garantizan el éxito de los mismos.

Los software al igual que cualquier producto que se elabora en la vida requiere de un gran número de actividades para su realización, en el desarrollo de cualquier actividad se pueden cometer errores ya que ellas son efectuadas por seres humanos y ninguno es perfecto, para garantizar la calidad de los productos software se creó una actividad muy importante en el ciclo de vida del producto, las pruebas

## **1.2 Marco Conceptual.**

### **1.2.1 Calidad**

Conjunto de propiedades de un producto o servicio que le confieren la aptitud de satisfacer las necesidades expresadas o implícitas de un cliente.[3]

### **1.2.2 Calidad del software**

“Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente” [4]

### **1.2.3 Gestión de la calidad del software**

Conjunto de actividades de función general de la dirección que determina la calidad, los objetivos y las responsabilidades y se implanta por medios tales como la planificación, el control, el aseguramiento (garantía) y la mejora de la calidad, en el marco del sistema de calidad establecido.

Teniendo un Sistema de Gestión de Calidad se demuestra previsibilidad en las operaciones internas, así como capacidad para satisfacer los requisitos del cliente. La gestión de la calidad permite una visión general que hace más fácil manejar, medir y mejorar los procesos internos.[5]

### **1.2.4 Sistema de calidad**

Conjunto de la estructura, responsabilidades, actividades, recursos y procedimientos de la organización de una empresa que ésta establece para llevar a cabo la gestión de la Calidad.

### **1.2.5 Prueba**

Según Diccionario de la Lengua Española:

- Examen o experimentación para comprobar el buen funcionamiento de alguna cosa o su adecuación a un determinado fin.
- Demostración de la verdad de alguna cosa, o de su existencia.
- Razón o argumento con que se demuestra la verdad o falsedad de una cosa.
- Parte pequeña o muestra de un todo que se analiza para comprobar su calidad.

### **1.2.6 Prueba del Software**

Es un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación. Sus objetivos son:

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.[4]

### **1.2.7 Laboratorio de calidad**

Se encarga de evaluar para cada versión independiente si cumple con un conjunto de requisitos de calidad y certifica basándose en los resultados de las pruebas y la conformidad con las normas y estándares definidos. Para establecer el nivel de calidad existen tres factores fundamentales:

- Las exigencias del mercado, (expresado en normas y estándares a cumplir).
- Las exigencias internas, (descritas en las políticas de calidad y resoluciones regionales).
- Los impuestos por contrato (detalles pactados entre consumidor y vendedor).

Dependiendo de estos factores el laboratorio debe organizarse y ser flexible, no siempre se evalúan los productos de la misma manera, se toma como base las exigencias internas que pueden estar por encima de las del mercado o el consumidor, pero esto no le molestará a nadie si se establecen normas y rigurosidades.[6]

### **1.2.8 Normas ISO**

Las normas son un modelo, un patrón, ejemplo o criterio a seguir. Una norma es una fórmula que tiene valor de regla y tiene por finalidad definir las características que debe poseer un objeto y los productos que han de tener una compatibilidad para ser usados a nivel internacional. La finalidad principal de las normas ISO es orientar, coordinar, simplificar y unificar los usos para conseguir menores costos y efectividad además tiene valor indicativo y de guía.[7]

### **1.3 Objeto de estudio**

#### **1.3.1 Objetivos estratégicos de la organización**

La producción en la Industria Cubana del Software, compuesta por los productos de la UCI y del resto de las empresas de software radicadas o no en la Universidad, ha crecido paulatinamente y se trabaja en aras de mejorar la calidad de sus productos.

Como apoyo a esta idea y teniendo en cuenta el tamaño colosal del proyecto UCI se necesita de una entidad que de confianza a los clientes y controle los niveles de calidad deseados para esta incipiente industria, por tanto se hace necesaria la concepción y construcción de un órgano capaz de medir, evaluar y certificar la calidad de la producción nacional de software, también instituir un Sello y Premio a la Calidad de los Productos que incluya no solo a los productos nacionales sino también a los destinados a la exportación.

Forman una parte importante de este órgano los laboratorios de calidad de cada una de las diez facultades con que cuenta la UCI. Como es lógico la Facultad 7 está insertada dentro de este mecanismo. Para su funcionamiento interno son muchos los elementos que lo componen: normas, procedimientos, procesos, servicios, trabajos de ingeniería e investigaciones. El Laboratorio de Calidad de la Facultad 7 evalúa uno o más proyectos al mismo tiempo, con un personal con conocimiento sobre el tema dedicados a tiempo parcial o completo, además requiriendo de recursos materiales puestos a su disposición lo que implica un mejor resultado asociado a la calidad del producto.

El funcionamiento del Laboratorio de Calidad de la Facultad 7 permite, primeramente, la evaluación de los productos software elaborados en los diferentes grupos de proyecto productivos con que cuenta la misma, además de cumplir con uno de sus objetivos: agilizar el trabajo del Laboratorio de Calidad Central. Con la adquisición de experiencia y madurez, contaría este equipo con un gran prestigio en la universidad.

Los productos que salen del Laboratorio de Calidad de la Facultad 7 demuestran un alto nivel cualitativo para su futura comercialización en cotizados mercados, incrementando la reputación de los

productores y la protección a las conformidades de los usuarios, al poseer un respaldo oficial. De esta forma dicho laboratorio se convertiría en un eslabón del proceso de certificación del software dentro de la Universidad de las Ciencias Informáticas.

### **1.3.2 Flujo actual de los procesos**

Llevar la calidad interna en un proyecto productivo es una tarea que requiere de varios factores y solo es conformada bajo el cumplimiento de diversos pasos fundamentales.

Para esto se determinan algunos roles en el grupo de calidad de la Facultad 7 y el trabajo de los integrantes dentro del desarrollo del proyecto, dando cumplimiento a todas las tareas definidas.

El flujo de trabajo que sigue este Grupo de Calidad actualmente está enfocado en el cumplimiento de las tareas que inician con la aparición de los entregables de cada área para la aplicación de pruebas. Las que se están aplicando de manera general son las pruebas exploratorias y las pruebas a la documentación. Estas pruebas a la documentación incluyen las pruebas al Manual de usuario, al manual de instalación (en caso de que el producto en cuestión lo requiera), a las especificaciones de los Casos de Usos y a los requerimientos funcionales y no funcionales, reflejándose estas últimas en el diseño de caso de pruebas.

Luego de haber realizado estas pruebas al software los resultados no favorables son plasmados en la Plantilla de No Conformidades; plantilla que se le es entregada al desarrollador del producto para la corrección de los errores encontrados. Los desarrolladores teniendo en cuenta estas no conformidades les realizan los cambios pertinentes a la aplicación y la entregan nuevamente para que el grupo de calidad realice una nueva revisión enfocándose solamente en aquellos errores encontrados anteriormente.

Finalizado este proceso, si el líder de proyecto considera que el software está listo para ser comercializado y tiene la calidad precisa se libera para que el Grupo de Calidad Central realice su trabajo que finaliza con un sello de calidad que avala la calidad del Software.

### **1.3.3 Análisis crítico de la ejecución de los procesos**

El análisis que se lleva a cabo en estos momentos con los productos que llegan al Laboratorio de Calidad, a pesar de que garantiza la confiabilidad del trabajo que se ha realizado, no es el óptimo, producto de que no se tienen en cuenta una serie de pasos a seguir que permita la orientación y organización del trabajo dentro y fuera del proyecto, que por ende influye en el proceso de revisión del producto. Además en el Laboratorio de Calidad se hace necesaria la utilización de una serie de documentos oficiales que permitan la eficiencia, factibilidad y confianza en el trabajo que se lleva a cabo.

Algunos de los factores que se deben tener en cuenta al analizar un determinado producto son:

- Deben estar definidos los espacios en que los miembros del equipo ventilen dudas e inquietudes, podrían definirse talleres de discusión de soluciones técnicas.
- Deben establecerse puntos de chequeo que permitan verificar el estado del proceso.
- Las tareas deben de estar definidas y bien delimitado su tamaño para poder precisar su avance.
- Se debe calcular y guardar todos los resultados, aplicándose métricas y manteniendo un estricto control de cada área. Esto se realizará de forma interna por parte de la persona o equipo designado a esta actividad.
- Se debe guardar en una especie de bitácora, todo el trabajo realizado, dígame documentación oficial del proyecto, plantillas, casos de uso de pruebas, copias de correos de trabajo, notas, etc.

Cada cierto tiempo, no muy largo, se debe realizar una actualización del Plan de Calidad.

### **1.4 Fundamentación de los objetivos**

El proceso de desarrollo de software a lo largo de los años en nuestro país ha adquirido un grado de profesionalidad digno y característico de la ingeniería. Gran parte de este profesionalismo se debe a la madurez adquirida por la Ingeniería de Software y la creciente complejidad de los sistemas de software que se vieron involucrados como elementos activos en tareas críticas de la industria.

Este desarrollo del software ha crecido paulatinamente desde la creación de la UCI. Consecuentemente con este aumento de la producción de software ha crecido su complejidad y se manifiesta ahora en entornos y clientes cada vez más exigentes en materia de calidad. Esta exigencia por parte de los usuarios hace que el proceso utilizado para el desarrollo del mismo deba ser minuciosamente analizado, y rigurosamente estandarizado a los efectos de asegurar y dar un marco seguro y confiable que garanticen la calidad del producto final.

Este criterio define la necesidad que hay de que el grupo de calidad de la Facultad 7 guíe su trabajo por un procedimiento estándar para que exista una uniformidad en cuanto a la calidad del software que se produzca, además para facilitarle el trabajo al grupo de CALISOFT.

Lograr el éxito en la producción de software es hacerlo con calidad. Esto es posible con la implantación de un Proceso de Revisión y Evaluación del Software basado en la norma ISO/IEC 9126 directamente relacionado con la política establecida para su elaboración y que esté en correspondencia con la definición internacional ISO de calidad, ampliamente aceptada.

### ***1.5 Antecedentes del tema***

Se ha evolucionado mucho en los enfoques de la calidad. A finales de los sesenta, cada vez que se quería afrontar la mejora del software todo se dirigía a la mejora del código dejando olvidado todo lo que le rodeaba. Ya en los años ochenta se comenzaron a tener en cuenta los aspectos de especificaciones, diseño, evaluación y gestión del software. Pese a estos cambios, persiste la problemática ya que el nivel de complejidad va aumentando y no se alcanzan los niveles de calidad deseados.

En el año 1986 el Departamento de Defensa de los EE.UU., encargó una guía para la selección de los contratistas del software. De aquí partió la creación de un método que permite valorar el proceso de software en función de un modelo de madurez de la capacidad, el Capability Maturity Model (CMM.). Surge así la solución a los problemas de calidad consistiendo en una mejora de los procesos con los que la organización construye, mantiene y gestiona el software.

La Internacional Organization for Standardization (OSI), constituye el otro foco de aportaciones. En 1987 publicó toda la serie ISO 9000: 9001, 9002 y 9003 que indican los elementos del sistema de calidad, y la 9004 aporta indicaciones y los elementos del sistema de calidad. Una tercera vertiente de aportaciones importantes procede de Australia.

Desde la óptica de la Calidad del Software, los aspectos de reutilización del software están siendo campo de estudio cara a las nuevas tecnologías. Autores como Frakes W., Terry C.[8], plantean estudios de referencia con vista a futuros trabajos: la medición del mantenimiento del software en proyectos específicos con nuevos perfiles o la evaluación de la calidad del software, en el ámbito de las bases de datos, en los esquemas conceptuales o los estudios en orientación a objetos y la aplicación de métricas a nuevas necesidades.

En el II Congreso Nacional de Auditoría y control de Sistemas de Información CASI-99, celebrado en Valencia en Octubre de 1999 [9], se planteó la necesidad de profundizar en el ámbito de la Auditoría de Calidad del Software. Dentro de los eventos que pueden ser referentes en el ámbito internacional se pueden mencionar las celebraciones periódicas de los congresos que dentro del ámbito de la Ingeniería del Software tienen un especial apartado en muchas de sus celebraciones a este campo.

Debido a los cambios rápidos e inexorables que se están produciendo en las tecnologías de la información y a las consecuencias irreversibles de quedarse atrás, las empresas se ven obligadas a asimilar la tecnología o morir. Las empresas tendrán que trabajar cada vez más para estar actualizadas.

Las nuevas tecnologías van a plantear la adecuación de las técnicas convencionales a los retos futuros en materias muy variadas. El interrogante puede estar en si se sabrán satisfacer las necesidades que las nuevas tecnologías demandan.

Es de suma importancia mencionar en el desarrollo de este trabajo investigativo que a pesar de ser el tema de calidad muy antiguo, el término de calidad del Software es un poco más reciente aunque ya existen algunas tesis y trabajos de autores tanto nacionales como internacionales que han tratado sobre el tema.

Es el caso de Juan Manuel Cueva Lovelle de la Universidad de Oviedo en España que hace un resumen detallado sobre los aspectos más importantes relacionados con la calidad del software, definiendo desde su punto de vista conceptos como: calidad del software, aseguramiento de la calidad del software, gestión de la calidad del software, control de la calidad del software, certificación de la calidad, factores que determinan la calidad del software y métricas de la calidad del software. En su trabajo enfatiza que " La dirección de la empresa es la responsable de fijar la política de calidad y las decisiones relativas a iniciar, desarrollar, implantar y actualizar el sistema de calidad" [10]

Además plantea que la calidad del software implica:

- La concordancia entre el software diseñado y los requisitos.
- Seguir un estándar o metodología en el proceso de desarrollo de software.
- Tener en cuenta los requisitos implícitos (no expresados por los usuarios).

Por su parte la Industria Cubana del Software (ICSW) está llamada a convertirse en una significativa fuente de ingresos nacional, como resultado del correcto aprovechamiento de las ventajas del considerable capital humano disponible. La Universidad de las Ciencias Informáticas y el sistema de empresas cubanas vinculadas a este trabajo jugarán un papel importante en el desarrollo de la Industria Cubana del Software, y en la materialización de los proyectos asociados al programa cubano de informatización.

La promoción de la industria cubana del software en el ámbito internacional ha tenido como línea estratégica aprovechar la enorme credibilidad que tiene Cuba en sectores tales como la salud, la educación y el deporte. Continuar la producción sostenida de software de alta calidad en prestaciones, imagen y soporte, para satisfacer las necesidades nacionales en estos sectores, tiene ya una positiva repercusión en el incremento de la exportación.

Es importante señalar que en nuestro país se avanza vertiginosamente en el desarrollo de software educativo para todos los niveles de enseñanza. Se desarrollan, además de los más de 100 productos que ya se usan en las escuelas y que son parte de varias colecciones, productos que cuentan con una alta calidad lo que le ha permitido al país su comercialización, por ejemplo:

- **Colección A jugar:** Jugar y Aprender, para la enseñanza preescolar.

- **8 productos de la Colección Multisaber:** El enigma de las plantas, Aprendiendo a convivir, Saber y hacer, El reino del ajedrez, Los libros mágicos, English 4U, English and you, Soy el maestro, PAEME 1ro y 2do ciclo, para la enseñanza primaria.
- **Otros productos:** Rumiantes, Termotecnia, Lenguaje y Técnicas de Programación, Operaciones Unitarias, Zootecnia, Mecánica del Transporte, Microbiología, Soldadura I, para la enseñanza técnico profesional.[11]

Está claro que las necesidades en Calidad del Software en los próximos años superan el ámbito convencional. Siempre es considerado un entorno, hoy casi superado, compuesto por las técnicas de medición y dictámenes producidos, el cumplimiento de los estándares de creación de software, control de cambios, estrategias de prueba, las revisiones formales, la aplicación de métodos y herramientas y los medios de gestión de la Calidad del Software.

Las nuevas perspectivas en la orientación a objetos, mejora de procesos de software, el estudio de componentes, la medición del software, el desarrollo y mantenimiento de software, la gestión y sistemas de calidad de software, normativa y certificación, Internet, Intranet, plantean nuevos retos futuros y una constante actualización de los medios actuales. En cualquier caso siempre se tiene que luchar con la evaluación de los niveles de concordancia del software, lo que plantea la búsqueda constante de nuevos medios de medición de las especificaciones.

Una evolución se puede apreciar a través de los Congresos de Calidad de Software, organizados por el Comité de Software de la European Organization for Quality (EOQ). El primero, celebrado en Bruselas en 1988, se centró en los elementos de control, tales como las pruebas y su método, herramientas y técnicas aplicables a las pruebas unitarias, de integración, de sistemas y de aceptación. En el celebrado en Oslo en 1990, las pruebas y las métricas de producto fueron el eje central. En 1992, en Madrid, resaltó la reciente publicación de la directiva comunitaria relativa a la certificación ISO 9001. En Basilea, en 1994, se dejó sentir la etapa de transición del sistema ISO 9001 a los modelos de madurez y mejora continua. El pasado 1996 se celebró en Dublín y quizás habrá que resaltar la consolidación de los modelos de mejora continua y la aportación de experiencias cada vez más frecuentes en la utilización de las tecnologías orientadas a los objetos. En este congreso se dedicó una sesión a este apartado.

## **1.6 Tendencias y tecnologías actuales**

### **1.6.1 Estándares de calidad**

La Calidad no es un tema nuevo para la formación profesional como sí lo es el uso de estándares o normas internacionales en el ámbito de las Instituciones con el objeto de generar una nueva cultura institucional.

En el marco de estrategias de implementación de una gestión de calidad total, las Instituciones de Formación Profesional de América Latina y el Caribe se interesan, cada vez más, en utilizar estándares internacionales para certificar la calidad de su proceso formativo

La aplicación de estándares de calidad en el marco de las acciones de modernización institucional y como señal de un trabajo bien hecho, se ha extendido en los últimos años dentro de las instituciones de formación de la región.

La aplicación de los estándares de calidad para la gestión ha evolucionado entre la versión 1994 de la norma ISO 9000, que se orientaba hacia el aseguramiento de la calidad y la versión 2000 de la misma norma; esta última ha incluido nuevas características que favorecen la calidad a partir del compromiso de la institución con un proceso de mejora continua.

### **1.6.2 Normas ISO 9000**

Es un conjunto de normas de calidad establecidas por la Organización Internacional para la Estandarización (ISO) que se pueden aplicar en cualquier tipo de organización.

Su implantación en estas organizaciones, un duro trabajo, supone una gran cantidad de ventajas para sus empresas. Los principales beneficios son:

- Reducción de rechazos e incidencias en la producción o prestación del servicio.
- Aumento de la productividad
- Mayor compromiso con los requisitos del cliente.
- Mejora continua.

La familia de normas apareció por primera vez en 1987 teniendo como base una norma estándar británica, y se extendió principalmente a partir de su versión de 1994, estando actualmente en su versión 2000.

Para verificar que se cumple con los requisitos de la norma, existen entidades de certificación que dan sus propios certificados y permiten el sello. Estas entidades están vigiladas por organismos nacionales que les dan su acreditación.

Para la implantación, es muy conveniente que apoye a la organización una empresa de consultoría, que tenga buenas referencias, y el firme compromiso de la Dirección de que quiere implantar el Sistema, ya que es necesario dedicar tiempo del personal de la empresa para implantar el sistema de calidad.

### **1.6.3 La calidad del Software dentro de la serie ISO/IEC 9001:2000**

El sistema de gestión de la calidad con base en ISO 9000 es genérico por naturaleza y aplicable a todas las empresas, independientemente de su tipo y del tamaño del negocio, incluidas las pequeñas y medianas empresas, y a todas las categorías de productos, ya sean hardware, software, materiales procesados o servicios.

La ISO 9001:2000 especifica **lo que** se requiere que haga una organización, pero no indica **cómo** se debería hacer, con lo que otorga a la empresa una gran flexibilidad para manejar su negocio. Su uso es sencillo, y su lenguaje claro y de fácil comprensión.

Dentro de la ISO/IEC 9001:2000 se pueden encontrar otras normas que están relacionadas con la calidad del software, como son los estándares:

- **Estándar ISO/IEC 9001:2000 TR 15504**

El Instituto de Organizaciones y Estandarización en conjunto con la Comisión Internacional de Electrotecnia, crearon un estándar de certificación y estandarización denominado ISO/IEC Reporte Técnico 15504 o TR 15504, que provee un modelo conceptual y marco para la evaluación, validación, optimización y certificación del proceso de desarrollo o construcción de software. Este marco puede ser utilizado por organizaciones que se vean involucradas en las diferentes etapas del proceso de

construcción y/o selección de software o proveedores del mismo, así como el planeamiento, gerenciamiento, monitoreo, control y mejoras en la adquisición, desarrollo, operación y soporte[12]

“La gestión y administración de la calidad se ha convertido en algo absolutamente importante para dejarlo librado al azar”[13] . Con esta premisa TR 15504 intenta ser el elemento diferenciador que por medio de la definición de un modelo y marco de referencia permiten evaluar el proceso de desarrollo de software de acuerdo a niveles definidos por la norma. Al mismo tiempo, permite asegurar la gestión de calidad en el proceso de desarrollo de software, identificando áreas de mejora y potenciales riesgos.

- **Estándar ISO/IEC 17025**

La nueva norma ISO 17025 sustituye a la Guía ISO/CEI 25 y a la norma europea EN45000. Contiene todos los requisitos que los laboratorios de ensayo tienen que cumplir si desean demostrar que disponen de un sistema de gestión de la calidad, que son técnicamente competentes y que son capaces de producir resultados técnicamente válidos.

La Entidad Nacional de Acreditación ha tomado a partir de Junio de 2001 dicha norma como referencia única para la acreditación de los laboratorios de ensayo.

Aunque ISO 17025 incluye muchas de las características y requerimientos ISO 9001, su enfoque es específico en competencia técnica para verificación. Existen requerimientos para la trazabilidad de las medidas y conocimiento de la incertidumbre de dicha medida, para la estructura y organización de actividades de laboratorio, para la calificación y competencia del personal identificando aquel personal que es clave, y para el informe de resultados.

ISO 17025 requiere de un mayor grado de competencia técnica que los requisitos impuestos por ISO 9001. Las aportaciones de ISO 17025 que la diferencian de ISO 9001 son:

- Factores que promuevan independencia en la medida
- Designar personal técnico y gerencia competente en temas de calidad
- Aspectos de confidencia y protección de propiedad intelectual

- Requisitos con mayor alcance específico para evaluar Identificar y definir metodología para asegurar consistencia de la calibración
- Metodología consistente para pruebas y ensayos
- Controles estrictos sobre procesos y actividades incluido cuando se contraten las mismas
- Registros de los aspectos previamente indicados

#### **1.6.4 NC ISO/IEC 9126**

La ISO/IEC 9126 (1991) “Software product evaluation – Quality characteristics and guidelines for their use”, la que fue elaborada para sustentar este propósito, definió seis características de calidad y describió un modelo de evaluación de los productos de software. Esta consta de las siguientes partes:

Parte 1: Modelo de calidad

Parte 3: Métricas Internas

Parte 2: Métricas Externas

Parte 4: Calidad en el uso

Solo la parte 1: Modelo de Calidad de esta norma americana fue adaptada a los intereses del mercado cubano (Ver Anexo 1) manteniendo su misma estructura y se le agregaron nuevas subcaracterísticas. Esta norma propuesta por Cuba describe un modelo en dos partes para calidad de los productos de software:

- Calidad interna y externa
- Calidad durante el uso (Ver figura 3).



**Figura 3. Características de la Calidad durante el Uso**

La primera parte del modelo especifica seis características para la calidad interna y externa, que son además divididas en subcaracterísticas que se manifiestan externamente cuando el software se usa como una parte del sistema computarizado, y son un resultado de los atributos internos del software. Esta parte de la NC ISO/ IEC 9126 no elabora el modelo para la calidad externa e interna a un nivel inferior al de las subcaracterísticas.

La segunda parte del modelo especifica cuatro características de calidad durante el uso, pero no elabora dicho modelo a un nivel inferior al de las características. La calidad durante el uso es el efecto combinado para el usuario de las seis características de calidad del producto de software.

Esta parte de la NC ISO/IEC 9126 permite especificar y evaluar la calidad del producto de software desde las perspectivas de aquellos asociados con la adquisición, regulación, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoria del software. Por ejemplo puede ser utilizada por los programadores, los clientes, el personal de aseguramiento de la calidad y los evaluadores independientes, particularmente los responsables de especificar y evaluar la calidad de los productos de software. El modelo de calidad definido en esta parte de la NC ISO/IEC 9126 puede usarse para:

- validar la integridad de la definición de los requisitos;
- identificar los requisitos del software;
- identificar los objetivos del diseño del software;
- identificar los objetivos de ensayo del software;
- identificar los criterios de aseguramiento de la calidad;
- Identificar los criterios de aceptación para un producto de software terminado

El modelo de la calidad establecido en la primera parte del estándar, ISO 9126-1, clasifica la calidad del software en un sistema estructurado de características y subcaracterísticas como sigue (Ver figura 4):



Figura 4. Características y subcaracterísticas de la Calidad Interna y externa

**1. Funcionalidad:** un sistema de las cualidades que refieren la existencia de un sistema de funciones y de sus características especificadas. Las funciones son las que satisfacen necesidades indicadas o implicadas.

- Conveniencia
- Exactitud
- Interoperabilidad
- Conformidad
- Seguridad

**2. Confiabilidad:** un sistema de las cualidades que refieren la capacidad del software para mantener su nivel del funcionamiento bajo condiciones indicadas por un período del tiempo indicado.

- Madurez
- Recuperabilidad
- Tolerancia de avería

**3. Utilidad:** un sistema de las cualidades que refieren el esfuerzo necesitó para el uso, y en el gravamen individual de tal uso, por un sistema indicado o implicado de usuarios.

- Instructibilidad
- Comprensibilidad
- Operabilidad

**4. Eficacia:** un sistema de las cualidades que refieren la relación entre el nivel del funcionamiento del software y la cantidad de recursos usados, bajo condiciones indicadas.

- Comportamiento de Tiempo
- Comportamiento del recurso

**5. Capacidad de mantenimiento:** un sistema de las cualidades que refieren el esfuerzo necesitó hacer modificaciones especificadas.

- Estabilidad
- Analizabilidad
- Cambiabilidad
- Ensayabilidad

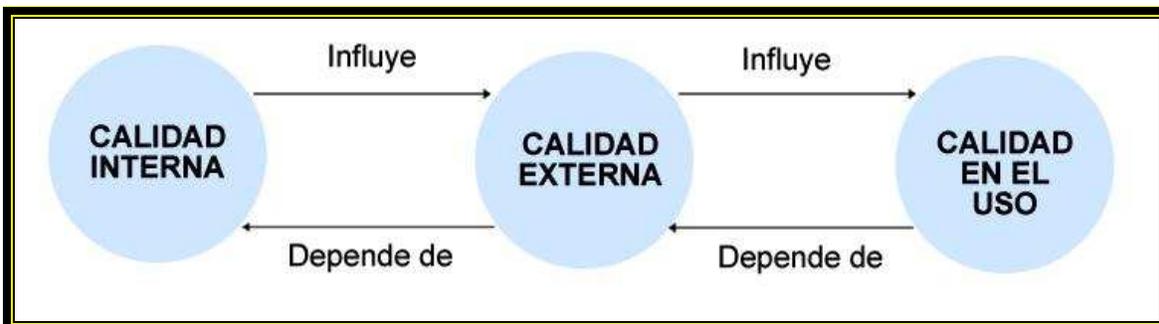
**6. Portabilidad:** un sistema de las cualidades que refieren la capacidad del software de ser transferido a partir de un ambiente a otro.

- Inestabilidad
- Reemplazabilidad
- Adaptabilidad

El producto de software abarca los ejecutables, código de fuente, descripciones de la arquitectura, y así sucesivamente. Consecuentemente, la noción del usuario extiende a los operadores así como a los programadores, que son usuarios de componentes como bibliotecas del software.

El estándar proporciona un marco para las organizaciones para definir un modelo de la calidad para un producto de software, sin embargo, deja a cada organización la tarea de especificar exacto su propio modelo.

Las métricas internas son las que no confían en la ejecución del software (medidas estáticas). Las métricas externas son aplicables al software corriente. Las métricas de funcionamiento de la calidad están solamente disponibles cuando el producto final se utiliza en condiciones verdaderas. Igualmente, la calidad interna determina la calidad externa y viceversa. (Ver figura 5).



**Figura 5. Relación entre Calidad externa, Calidad Interna y Calidad durante el Uso**

Este estándar proviene del modelo establecido en 1977 por McCall y sus colegas, que propusieron un modelo para especificar calidad del software. El modelo de la calidad de McCall se organiza alrededor de tres tipos de características de la calidad:

- Factores (especificar): Describen la vista externa del software, según lo visto por los usuarios.
- Criterios (construir): Describen la vista interna del software, según lo visto por el revelador.

- Métrica (al control): Se definen y se utilizan para proporcionar una escala y un método para la medida.

La ISO 9126 distingue entre un defecto y una inconformidad, un **defecto** que es el incumplimiento de los requisitos previstos del uso, mientras que una **inconformidad** es el incumplimiento de requisitos especificados. Una distinción similar se hace entre la validación y la verificación, sabidas como V&V en el comercio de prueba.

### **1.6.5 Marco actual de las pruebas de software**

Una de las mayores preocupaciones de las organizaciones que desarrollan software, es el nivel de calidad de los productos desarrollados. Los procesos de Ingeniería del Software usados para este efecto, se entremezclan con procesos de ingeniería de requisitos, análisis funcional, gestión de las peticiones, gestión de la configuración del software, análisis, diseño y ejecución de pruebas, etc...., de forma que cada una de las actividades que se realizan tengan un nivel de calidad mínimo y tangible.

La fase de pruebas absorbe una buena porción de los costos de desarrollo de software. Además, se muestra renuente a un tratamiento matemático o, simplemente, automatizado. Su ejecución se basa en metodología (reglas que se les dan a los encargados de probar) que se va desarrollando con la experiencia. Es tediosa, es un arte, es un trabajo que requiere una buena dosis de mala intención, y provoca difíciles reacciones humanas.

Aunque se han desarrollado miles de herramientas de soporte de esta fase, todas han limitado su éxito a entornos muy concretos, frecuentemente sólo sirviendo para el producto para el que se desarrollaron. Sólo herramientas muy generales como analizadores de complejidad, sistemas de ejecución simbólica y medidores de cobertura han mostrado su utilidad en un marco más amplio. Pero al final sigue siendo imprescindible un artista humano que sepa manejarlas.

### **1.6.6 Principios de las pruebas de software**

Para poder realizar las pruebas de una manera más efectiva se han planteado diferentes principios entre los más importantes están:

- A las pruebas se les debe hacer un seguimiento hasta los requisitos del cliente. Para los clientes los errores más graves son los que impiden que se cumplan los requerimientos.
- Las pruebas deben planificarse mucho antes de que comiencen.
- Las pruebas deberían empezar por lo pequeño y progresar hacia lo grande.
- No es posible realizar pruebas exhaustivas.
- Para ser más eficaces las pruebas deberían ser realizadas por un equipo independiente.
- La prueba es usada para demostrar la presencia de errores pero no la ausencia de ellos.
- Los casos de prueba tienen que ser descritos no solo para condiciones válidas de entradas sino también para condiciones no válidas e inesperadas.
- El número de errores sin descubrir es directamente proporcional al número de errores descubiertos.

### 1.6.7 Métodos de prueba

Son dos fundamentales: el método de la caja negra y de la caja blanca.

La prueba de **caja negra** se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene (Ver figura 6).

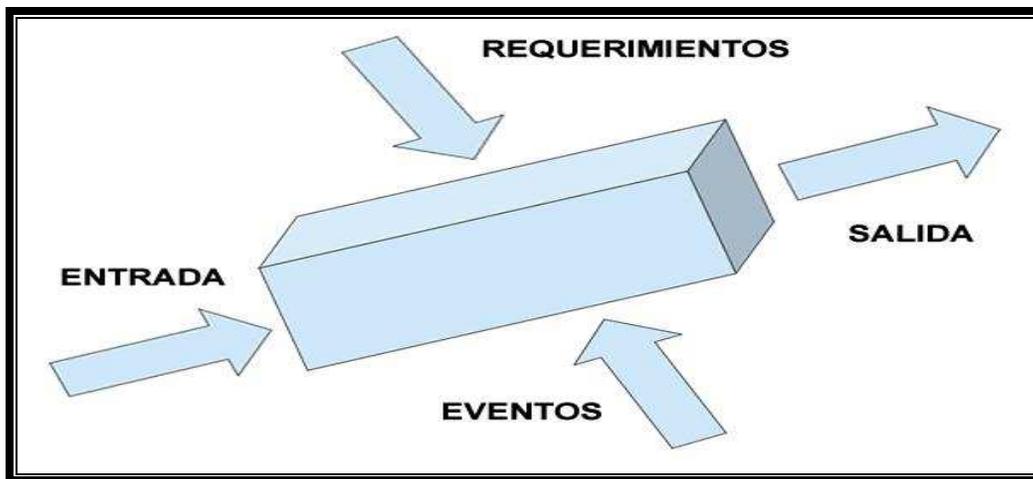


Figura 6. Pruebas de Caja Negra

La prueba de **caja blanca** del software se comprueba los caminos lógicos del software proponiendo casos de prueba que se ejerciten conjuntos específicos de condiciones y/o bucles. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coinciden con el esperado o mencionado.

### **Pruebas de Caja Negra**

- Verifican las especificaciones funcionales y no consideran la estructura interna del programa.
- Es hecha sin el conocimiento interno del producto.
- No validan funciones ocultas (por ejemplo funciones implementadas pero no descritas en las especificaciones funcionales del diseño) por tanto los errores asociados a ellas no serán encontrados.

En otras palabras, la prueba de la caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. O sea los casos de prueba pretenden:

1. Demostrar las funciones del software son operativas.
2. Que las entradas se aceptan de la forma adecuada y que se produce el resultado correcto.
3. Así como la integrada de la información externa (por ejemplo archivos de datos) se mantiene.

### **Técnicas para desarrollar Pruebas de caja negra**

1. **Técnica de la Partición de Equivalencia:** esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
2. **Técnica del Análisis de Valores Límites:** esta técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
3. **Técnica de Grafos de Causa-Efecto:** es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

Dentro del método de Caja Negra la técnica de la Partición de Equivalencia es una de las más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el software, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la

definición de casos de pruebas que descubran clases de errores, reduciendo así en número de clases de prueba que hay que desarrollar.

### **Pruebas de caja blanca:**

Requieren del conocimiento de la estructura interna del programa y son derivadas a partir de las especificaciones internas de diseño o el código.

Mediante los métodos de prueba de la caja blanca, el ingeniero de software puede obtener casos de prueba que garanticen que:

1. Se ejerciten por lo menos una vez todos los caminos independientes para cada módulo.
2. Se ejerciten todas las decisiones lógicas en sus vertientes verdaderas y falsa.
3. Ejecuten todos los bucles en sus límites y con sus límites operacionales.
4. Se ejerciten las estructuras internas de datos para asegurar su validez.

Dado las alternativas de la caja negra y la caja blanca; y de la necesidad de que las pruebas abarquen los requerimientos, las funciones y la lógica interna del programa entonces:

- Las pruebas para los requerimientos deben emplear la estrategia de la caja negra.
- Las pruebas para las funciones deben emplear la estrategia de la caja negra.
- Las pruebas para la lógica interna tiene que necesariamente emplear la estrategia de la caja blanca.

La prueba de la caja blanca, a primera vista, podría parecer impracticable puesto que no es posible aplicarla exhaustivamente para grandes sistemas, sin embargo no se debe desechar; ya que se puede elegir y ejercitar una serie de caminos lógicos importantes, que invoquen además las estructuras de datos más importantes para comprobar su validez.

*Se pueden combinar ambos métodos para llegar a un método que valide la interfaz del software y asegure selectivamente que el funcionamiento interno del software es correcto.*

### **Técnicas para desarrollar Pruebas de Caja Negra**

1. **La prueba del camino básico:** Esta prueba permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. Los casos de prueba obtenidos del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa.
2. **La prueba de condición:** Es un método de diseño de casos de prueba que ejercita las condiciones lógicas contenidas en el módulo de un programa.
3. **La prueba de flujo de datos:** Se selecciona caminos de prueba de un programa de acuerdo con la ubicación de las definiciones y los usos de las variables del programa.
4. **La prueba de bucles:** Es una técnica de prueba de caja blanca que se centra exclusivamente en la validez de las construcciones de bucles.

Los métodos de caja blanca pueden aplicarse a las operaciones que se definen para una clase. Sin embargo, la concisa estructura de muchas operaciones de la clase provoca que algunos argumenten que el esfuerzo aplicado en la prueba de tipo caja blanca pudiera redirigirse mejor hacia pruebas a nivel de clase.

#### **1.6.8 Tipos de prueba**

Las pruebas se desarrollan durante todo el ciclo de vida del proyecto para ir corrigiendo los errores en la medida que se cometen y no dejarlos todo para el final, por tal motivo se han definido diferentes niveles de pruebas:

1. **Prueba de unidad** es la prueba enfocada a los elementos testeables del software, comienzan una vez se ha desarrollado, revisado y verificado en su sintaxis el código fuente.
2. **Prueba de integración:** Es ejecutada para asegurar que los componentes en el modelo de implementación operen correctamente cuando son combinados para ejecutar un caso de uso. Es el proceso de combinar y probar múltiples componentes juntos.
3. **Prueba de Sistema:** Son las pruebas que se hacen cuando el software está funcionando como un todo. Es la actividad de prueba dirigida a verificar el programa final, después que todos los componentes de software y hardware han sido integrados.

4. **Pruebas de aceptación:** Es la prueba final antes del despliegue del sistema, generalmente es realizada por el usuario final. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido. Cuenta con 3 tipos o tres clasificaciones que son: las pruebas de aceptación formales, las pruebas de aceptación informales o alfa, y las pruebas betas.

El nivel de la prueba va avanzando con el incremento del producto. Si los errores no son detectados en etapas tempranas del desarrollo del software requerirán un gran esfuerzo para su detección en etapas finales y será muy costoso para las organizaciones pero si bien los errores no son detectados en los niveles iniciales si tienen que ser detectados en su gran mayoría antes de que el producto sea entregado al usuario final.

### **1.6.9 Otros tipos de pruebas**

- Recorridos (walkthroughs)
- Aleatorias (random testing)
- Solidez (robustness testing)
- Aguante (stress testing)
- Prestaciones (performance testing)
- Conformidad u Homologación (conformance testing)
- Interoperabilidad (interoperability testing)
- Regresión (regression testing)
- Mutación (mutation testing)

### **1.6.10 Depuración (debugging)**

Casi todos los compiladores suelen llevar asociada la posibilidad de ejecutar un programa paso a paso, permitiéndole al operador conocer dónde está en cada momento, y cuánto valen las variables.

Los depuradores pueden usarse para realizar inspecciones rigurosas sobre el comportamiento dinámico de los programas. La práctica demuestra, no obstante, que su uso es tedioso y que sólo son eficaces si se persigue un objetivo muy claro. El objetivo habitual es utilizarlo como consecuencia de la detección de un error. Si el programa se comporta mal en un cierto punto, hay que averiguar la

causa precisa para poder repararlo. La causa a veces es inmediata (por ejemplo, un operador booleano equivocado); pero a veces depende del valor concreto de los datos en un cierto punto y hay que buscar la causa en otra zona del programa.

En general es mala idea "correr al depurador", tanto por el tiempo que se pierde buceando sin una meta clara, como por el riesgo de corregir defectos intermedios sin llegar a la raíz del problema. Antes de entrar en el depurador hay que delimitar el error y sus posibles causas. Ante una prueba que falla, hay que identificar el dominio del fallo, averiguar las características de los datos que provoca el fallo (y comprobar experimentalmente que todos los datos con esas características provocan ese fallo, y los que no las tienen no lo provocan).

El depurador es el último paso para convencernos de nuestro análisis y afrontar la reparación con conocimiento de causa.

### **1.6.11 Plan de Pruebas**

Un plan de pruebas está constituido por un conjunto de pruebas. Cada prueba debe:

- dejar claro qué tipo de propiedades se quieren probar (corrección, robustez, fiabilidad, amigabilidad, ...)
- dejar claro cómo se mide el resultado.
- especificar en qué consiste la prueba (hasta el último detalle de cómo se ejecuta).
- definir cuál es el resultado que se espera (identificación, tolerancia,...) ¿Cómo se decide que el resultado es acorde con lo esperado?

Las pruebas angelicales carecen de utilidad, tanto si no se sabe exactamente lo que se quiere probar, o si no está claro cómo se prueba, o si el análisis del resultado se hace "a ojo".

Estas mismas ideas se suelen agrupar diciendo que un caso de prueba consta de 3 bloques de información:

1. El propósito de la prueba
2. Los pasos de ejecución de la prueba
3. El resultado que se espera

Y todos y cada uno de esos puntos debe quedar perfectamente documentado. Las pruebas de usar y tirar más vale que se tiren directamente, aún antes de usarlas.

Cubrir estos puntos es muy laborioso y, con frecuencia, tedioso, lo que hace desagradable (o al menos muy aburrida) la fase de pruebas. Es mucho más divertido codificar que probar. Tremendo error en el que, no obstante, es fácil incurrir.

Respecto al orden de pruebas, una práctica frecuente es la siguiente:

1. Pasar pruebas de caja negra analizando valores límite. Recuerde que hay que analizar condiciones límite de entrada y de salida.
2. Identificar clases de equivalencia de datos (entrada y salida) y añadir más pruebas de caja negra para contemplar valores normales (en las clases de equivalencia en que estos sean diferentes de los valores límite; es decir, en rangos amplios de valores).
3. Añadir pruebas basadas en "presunción de error". A partir de la experiencia y el sentido común, se aventuran situaciones que parecen proclives a padecer defectos, y se buscan errores en esos puntos. Son pruebas del tipo "¡Me lo temía!"
4. Medir la cobertura de caja blanca que se ha logrado con las fases previas y añadir más pruebas de caja blanca hasta lograr la cobertura deseada. Normalmente se busca una buena cobertura de ramas (revise los comentarios expuestos al hablar de caja blanca).

### ***1.7 Análisis crítico de las fuentes y bibliografías utilizadas***

Para el desarrollo del presente trabajo se consultó fundamentalmente bibliografía de Internet. A pesar de que muchas personas consideran que la información que se encuentre en la red de redes no es fiable, las autoras de este trabajo se han cerciorado de que los autores de los artículos consultados sean personas con una preparación adecuada sobre el tema que se trata, la totalidad de ellos son profesionales en la materia que cuentan en su currículum con destacados títulos en maestrías y postgrados.

La veracidad de la información estudiada se ratifica teniendo en cuenta una serie de elementos presentes en los artículos. Se puede mencionar que en el caso de los artículos la gran mayoría

tenían referencias del autor como es su correo tanto electrónico como postal y dirección de centro de trabajo. En el caso de las páginas Web todas contaban con un logotipo que las identificaba así como medios de contactos. Es válido destacar también que estas páginas Web representan a empresas y entidades reconocidas.

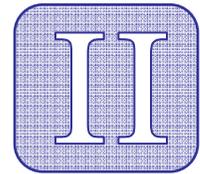
La calidad de la información que se consultó resultó ser la óptima ya que en todos los artículos se mostraba una información clara, precisa, detallada, que compensaba las necesidades de las personas que los examinaron. Es importante resaltar que esta información recolectada poseía elementos tan importantes e imprescindibles a la hora de consultar como son la fecha de creación y actualización, lo que le permite a los usuarios una mayor confianza en la fuente bibliográfica de la que están asesorándose.

### **1.8 Conclusiones Parciales**

Se puede decir entonces que la calidad de software es una de las problemáticas más abordadas en la actualidad y que todo el proceso de revisión de un sistema informático está estrechamente relacionado con la calidad que puede tener.

En el Laboratorio de Calidad de la Facultad 7 de la Universidad de las Ciencias informáticas el proceso de revisión de software no se lleva a cabo con la precisión requerida ya que se revisan solamente aquellos sistemas que están terminados y se omiten pasos durante esta revisión. Además, una problemática existente es que los miembros de este equipo no poseen vastos conocimientos prácticos.

En este capítulo se decidió tomar la norma cubana ISO/IEC 9126-1:2005, debido a que además de ser la que más se ajusta a la situación anteriormente abordada, define qué características se deben evaluar en un sistema que ya este terminado y ofrece la flexibilidad al que la usa de definir sus propios métodos evaluativos.



---

## Capítulo 2. Procedimiento Propuesto.

Llevar la calidad interna en un proyecto productivo, es una tarea que requiere de varios factores y solo es conformada bajo el cumplimiento de diversos pasos fundamentales. Para el desarrollo de este capítulo se propone un procedimiento para la revisión y evaluación de productos de software terminados dentro del Laboratorio de Calidad de la Facultad<sup>7</sup>. Este pretende proporcionar la secuencia de pasos mínimos necesarios, para lograr el cumplimiento de nuestro propósito fundamental: la calidad en la producción. Dicho procedimiento incluye en su propuesta, los requisitos de implantación, los pasos a seguir, así como las planillas a cumplimentar durante la aplicación del mismo.

## 2.1 Propósito

Verificar la calidad de productos de software tanto para la salud como para el procesamiento digital de imágenes y señales, mediante:

- La realización de pruebas.
- La aplicación de la norma NC ISO/IEC 9126-1: 2005 para la evaluación del software.

## 2.2 Alcance

Este procedimiento abarca a todos los productos de software terminados y elaborados por la Facultad 7., y a concierne a todo el personal probador que va a participar en la revisión del software en el Laboratorio de Calidad.

## 2.3 Definiciones, Acrónimos y Abreviaturas

Palabras Claves	Definiciones
Calidad	Grado en que un conjunto de características inherentes al producto cumple con los requisitos.
Casos de Pruebas	Instrucciones documentadas para la persona que realiza las pruebas que especifican cómo se tiene o tendría que probar una función o una combinación de funciones.
Documentación del Producto	Documentos imprescindibles para que el producto entre en el Laboratorio de Calidad, esto incluye: Manual de Usuario, Manual de Instalación, Expediente del Producto, Instalación del Producto de Software.

<b>Defectos del Producto</b>	Defectos existentes en el producto que hacen que el mismo no cumpla con los requisitos especificados por el cliente para su uso o exista un incumplimiento de los estándares definidos por el Grupo de Calidad.
<b>Cliente</b>	Representante de un proyecto productivo que solicite los servicios de revisión de un Software al Laboratorio de Calidad.
<b>Normas y Estándares</b>	Son acuerdos documentados que contienen especificaciones técnicas u otros criterios precisos para ser usados consistentemente como reglas, guías, o definiciones de características, para asegurar que los materiales, productos, procesos y servicios se ajusten a su propósito.
<b>Informe final</b>	Documento que presenta resultados obtenidos o proporciona evidencia de actividades desempeñadas.
<b>Prueba de Caja Negra</b>	Tipo de prueba en la que se verifica la funcionalidad del sistema o parte sin tener en cuenta la estructura o código fuente que garantiza esta funcionalidad.

## 2.4 Requerimientos de Organización del Laboratorio

### 2.4.1 Estructura Organizativa

En el laboratorio se definen diferentes roles, cuyas responsabilidades se describen a continuación:

**El Líder del Proyecto de Calidad (LP)** es responsable de:

1. Recibe la documentación del producto terminado, así como el producto en sí en la versión que va a ser revisada
2. Controla, actualiza y brinda en todo momento la información necesaria sobre:
  - a. Planificación del proceso de pruebas.
  - b. Versiones del producto entregado y el control de tareas.
  - c. Documentación derivada de la herramienta de ingeniería de software usada, debidamente modelada con la metodología adecuada y gestionada a través de la herramienta utilizada.
3. Entrega al Laboratorio de Liberación los artefactos necesarios, establecidos en los correspondientes procedimientos referenciados en este documento

**Jefe del Laboratorio de Calidad** es responsable de:

1. Estar al tanto del funcionamiento del laboratorio y la confiabilidad del proceso.
2. Planifica y supervisa todo el proceso de revisión.
3. Conformar el equipo evaluador y entrega al mismo de la documentación del producto y los resultados obtenidos por los Ingenieros de Pruebas de cada producto.
4. Es el responsable de la elaboración del informe final y el autorizado para que este tenga validez legal.
5. Archiva estos resultados y la copia de la versión del producto con la documentación entregada.

**Equipo de Calidad Interno al Proyecto:**

1. Realiza un proceso interno de revisiones de calidad al producto entregado por un determinado equipo de desarrollo.
2. Sigue las instrucciones dictadas por la Dirección de Calidad (líder del proyecto y jefe del laboratorio), desarrollando los procesos dictados por la misma.
3. Genera la documentación con los resultados obtenidos de sus procesos de prueba y revisiones técnicas, utilizando las plantillas normadas por la misma.

✓ **Diseñador de Pruebas:**

1. Planifica y diseña los tipos de prueba por proyecto y en dependencia de estos tipos de pruebas, asigna a los ingenieros de pruebas para realizar las mismas.
- ✓ **Desarrollador de Pruebas:** Personal especializado en realizar uno o más tipos de pruebas. Para llevar a cabo los procesos de pruebas éste no los diseña, sino que se basa en los casos de prueba ya diseñados.
    - ❖ **Desarrollador de pruebas de integración:** Realizan las pruebas de integración.
    - ❖ **Desarrollador de pruebas de sistema:** Realizan las pruebas del sistema.

### Equipo Evaluador:

1. Recepciona y evalúa los resultados de las pruebas que les fueron entregados por el Jefe del Laboratorio de Calidad.
2. Son los encargados de emitir el criterio final junto al informe que será emitido al Jefe del Laboratorio.
  - ✓ **Evaluador Principal:**
    1. Encargado de la planificación de los encuentros.
    2. Organiza y distribuye las tareas entre los especialistas evaluadores
    3. Responde ante el jefe del laboratorio por la calidad del proceso de evaluación y del informe final.
  - ✓ **Registrador:**
    1. Encargado de elaborar el informe final de cada versión.
  - ✓ **Especialistas Evaluadores:**
    1. Encargados de evaluar los resultados, son por lo general especialistas de prestigio o asesores externos con experiencia y criterios sobre el tipo de producto en proceso.

**Personal Experto:** Es un funcionario o experto en el tema que trata el producto que se desea revisar.

1. Confirma las salidas de los diseños de caso de pruebas.
2. Este personal puede tener o no conocimientos informáticos así como ser o no trabajador de la UCI.

### **2.4.1.1 Habilidades del personal de Calidad**

Estas son algunas de las habilidades con que deben contar los miembros del Laboratorio de Calidad.

### **Líder del Proyecto de Calidad y Jefe del Laboratorio de Calidad:**

1. Experiencia en el proceso de desarrollo de software, el dominio de la aplicación y la plataforma.
2. Asignación y planeación de recursos.
3. Habilidades para el análisis de riesgos, dependencias y toma de decisiones.
4. Habilidades para presentar negociar y comunicar.
5. Experiencia en la administración de proyectos.
6. Debe tener capacidad para atacar y resolver los problemas que se presenten.

### **Equipo de calidad interno al proyecto:**

1. Experiencia escribiendo casos de prueba.
2. Experiencia en usar técnicas de diseño de pruebas.
3. Habilidad e interés para aprender nuevas herramientas y tecnologías.
4. Conocer los tipos de pruebas
5. Desarrollo de casos de pruebas.
6. Uso de herramientas para ejecutar pruebas
7. Determinar requerimientos funcionales y no funcionales
8. Realizar revisiones al diseño.
9. Uso de los estándares de codificación definidos
10. Definir defecto, tipo de defectos (lógicos y de sintaxis), registrarlos y reportarlos
11. Reportes de prueba con registro de defectos

### **Equipo Evaluador:**

1. Registrar defectos y elaborar reporte.
2. Habilidad para documentar los errores con claridad y precisión.
3. Pueda trabajar eficientemente ante plazos cortos de entrega.

Para que los miembros del equipo que van a revisar un determinado software puedan tener estas habilidades, no solo lo logran con la dedicación y el esfuerzo personal, sino que por parte de la Dirección del Proyecto se brindan cursos de capacitación y nivelación incrementando de esta forma sus conocimientos sobre el tema de calidad del software.

Esta superación puede ser recibida antes o durante del proceso que les permita enfrentarse al problema en cuestión con una visión amplia y con un número mayor de herramientas.

### **2.4.2 Equipamiento del Laboratorio (Software y Software)**

Debido a que la Universidad cuenta con tecnologías de desarrollo de software avanzadas todos los productos desarrollados en ella se soportan sobre software libre y en su mayoría utilizan tecnología Web.

Es por esta razón que para equipar el Laboratorio de Calidad se hace necesario analizar dos factores imprescindibles. Primero hay que tener en cuenta las necesidades básicas para trabajar y como segundo punto las necesidades que requiere el producto que se desea revisar.

Si se analiza el primer factor, se concluye que el laboratorio necesita:

- Al menos una PC por cada dos miembros.
- Que sean Pentium IV CPU 2.40Ghz (aunque aún se hacen software con generaciones anteriores de PCs no influye para el trabajo en el Laboratorio, puesto que los software que se pretenden revisar son de la Facultad 7 y todos los laboratorios de la UCI están implementados con Pentium IV o superiores).
- Las PCs deben tener instalados como sistemas operativo: Microsoft Windows 2000 o superior y Linux Debian.
- Lectores de CD y DVD de 24x.
- Monitores VGA 32 bits de colores a 800 x 600.
- Mouses.
- Tarjetas de red 100 Mb.
- Un pequeño switch de 8 puertos para conectarlas a la red en el laboratorio.
- Un Servidor, con amplia capacidad de almacenamiento.
- Quemador.
- Impresora.
- Scanner.

Estas características son las mínimas que puede tener el Laboratorio para lograr un trabajo eficiente aunque se tienen que tener en cuenta las necesidades de pruebas y el entorno en que se quiere que se pruebe un determinado producto. Esto está en dependencia de los requerimientos de software y hardware que necesite el producto en cuestión especificados en el expediente del mismo, de acuerdo a la infraestructura del producto a revisar se debe simular lo mejor posible el ambiente del cliente.

### 2.4.3 Condiciones de Seguridad del Laboratorio para el Producto

La seguridad es fundamental a la hora de afrontar tareas que se realizan en sistemas informáticos ya que son las únicas medidas que pueden garantizar que éstas se realicen con una serie de garantías que se dan por sentado en el mundo físico.

Para que un programa informático sea seguro no basta con utilizarlo correctamente. Hace falta que esté libre de fallos, que no tenga puertas traseras y que no posea ninguna funcionalidad "no documentada".

Es por esta razón que se hace necesario que unos de las primeras cosas que se deben tener en cuenta a la hora de aceptar el producto de software a revisar es que el mismo cuente con un sistema de seguridad óptima, para evitar posteriormente pérdidas de información, contaminación por virus o accesos no autorizados a información confidencial.

Desde la hora en punto en que se recibe el producto, el laboratorio de calidad se compromete a:

1. Proteger el producto durante el proceso de revisión.
2. Ninguna persona no autorizada tendrá acceso al producto de software.
3. No se divulgará información alguna sobre el mismo.
4. No se le realizarán bajo ninguna circunstancia cambio alguno al software que se está revisando.

Es necesario también llegar a un acuerdo importante entre el grupo de calidad y los desarrolladores del producto en cuanto las restricciones de seguridad, como por ejemplo, en ocasiones el software durante el proceso de revisión transita continuamente por la red, por lo que es imprescindible disponer de algún medio de transporte de información.

Además, en el Laboratorio de Calidad deben existir muebles con cierta seguridad para almacenar tanto los documentos firmados, como las versiones involucradas en cada revisión.

### **2.5 Acciones a llevar a cabo en el proceso**

#### **2.5.1 Planificación y Preparación de la Revisión Técnica**

##### **2.5.1.1 Planificación de Actividades**

- El cliente debe solicitar la revisión y evaluación de su producto de software, a través del Formulario de Solicitud de Revisión de Producto de Software (Ver Anexo 2).
- Dada esta solicitud se debe planificar la fecha de entrada del producto al Laboratorio.
- Todas las solicitudes que se le hacen al Laboratorio de Calidad son analizadas y quedan registradas en un cronograma de trabajo teniendo en cuenta las prioridades de evaluación del producto. Estas prioridades las definen en conjunto el Laboratorio de Calidad y la Dirección de la Facultad 7.
- Se realiza una reunión con los propietarios del producto a evaluar. Esta reunión tiene como objetivo informar y explicar el proceso de evaluación, entregar a los desarrolladores de las pruebas lo necesario para realizar la revisión, firma del Acuerdo de Confidencialidad (Ver Anexo 3) y definir la persona que atenderá al equipo de revisión en caso que sea necesario.
- En el laboratorio se realiza la reunión de inicio donde participan todo el personal que intervendrá en el proceso de revisión y evaluación, con el objetivo de analizar cómo se le van a realizar las pruebas a este producto, aclarar todas las dudas que puedan existir sobre estas pruebas y definir fecha de reunión con los propietarios del producto a revisar.
- Las revisiones del producto serán discutidas específicamente con el desarrollador del producto.
- Una vez revisado el software el equipo de evaluación debe realizar una reunión final para elaborar la evaluación definitiva del producto teniendo en cuenta los resultados de las pruebas aplicadas y emitir una evaluación cualitativa donde se comenten aspectos que crean

importantes entre los que fueron evaluados, así como sugerencias que se puedan hacer con vistas a mejorar el producto o servicio.

- Finalmente realizar una reunión de información de los resultados de la revisión y evaluación al desarrollador del producto.
- Si el producto de software necesita una segunda revisión el líder del proyecto del Laboratorio de Calidad se lo comunica al desarrollador del producto en cuestión.
- Los productos de software que necesiten una segunda revisión tienen prioridad por encima de los que se están revisando y de los que están planificados para comenzar a revisar, teniendo en cuenta el tiempo de solución de los problemas planteados.

### **2.5.1.2 Cronograma de Actividades**

1. Deben realizarse reuniones de equipo de forma sistemática como parte del proceso normal de revisión dirigidas por el Líder del Proyecto.
2. Deben definirse espacios en que los miembros del equipo ventilen dudas e inquietudes, podrían definirse talleres de discusión de soluciones técnicas aprobados por el Líder del Proyecto y el Jefe del Laboratorio.
3. Planificadas las fases y las iteraciones del proceso de revisión del producto de software. Encargado: Líder del Proyecto.
4. Se establecerán puntos de chequeo que permitan verificar el estado del proceso de revisión. Encargado: Líder del Proyecto.
5. Delimitado el tamaño de las tareas para poder precisar su avance. Encargado: Líder del Proyecto.
6. Registrarse de forma individual los defectos encontrados. Encargado: Desarrolladores de las pruebas.

### **2.5.1.3 Análisis del Producto**

1. En el proyecto debe estar claramente definido el tipo de software que será producido: Software para la Salud.

2. Debe existir un concepto claro acerca de la complejidad del proyecto, evaluable en correspondencia con el esfuerzo investigativo que deba desplegarse para llevar a cabo la realización del mismo.
3. Debe realizarse una estimación del tiempo y tamaño del proyecto que tribute a la cantidad de casos de uso y a una clasificación de estos que considere variables tales como: complejidad del mismo, cantidad de requisitos, experiencia que se tenga en su solución, relevancia en la arquitectura, jerarquía dentro de esta, etc.
4. Debe realizarse una estimación del tamaño del proyecto que tribute a la cantidad de personas que se necesitan para revisar el proyecto.
5. Teniendo en cuenta el tipo de proyecto que se va a revisar se determinará el objetivo de las pruebas y un estimado de las que se les va a aplicar.

### **2.5.1.4 Requisitos de Aceptación de Producto**

La entrega se realizará en una PC acordada con anterioridad por ambos líderes, donde solo tengan acceso el Líder del Proyecto a revisar y el Líder del Proyecto de Calidad.

El Líder del Proyecto de Calidad recibe el paquete del producto a revisar en el Laboratorio de Calidad de la Facultad 7. Chequea que estén todos los elementos que compone dicho paquete y procede a aceptar el producto, en dependencia de la valoración de lo recibido.

De decidirse, por parte del Líder del Proyecto de Calidad que no se puede proceder a probar a partir de la entrega realizada, se le comunica al Líder del Proyecto (el desarrollador del producto) que hizo la solicitud y se esperaría por una próxima entrega.

En caso de que se acepte el producto es porque cumple con todas las exigencias de entrega del producto al Laboratorio de Calidad y el paquete del mismo contiene:

- Expediente del Producto de Software (Ver Anexo 4).
  - Descripción del Producto: Requisitos de software y hardware, Requisitos de documentación, Requisitos de personal y Otros.
  - Programas y datos necesarios.

- Descripción de los Casos de Uso
- Especificación de Requisitos (Funcionales y no Funcionales)
- Manual de Instalación.
- Manuel de Usuario o Ayuda On-Line.
- Instalación del Software.
- Ayuda.
- Luego de aceptado el Paquete se prosigue con la firma del Acta de Entrega (Ver Anexo 5).

### **2.5.1.5 Características de Calidad que debe tener el Producto Final**

Deben ser definidas por parte del Laboratorio de Calidad y se resumen en:

- Características operativas.
- Capacidad para soportar cambios.
- Adaptabilidad a nuevos entornos.

De modo tal que responda a los siguientes factores:

- Corrección: grado en que una aplicación satisface sus especificaciones y consigue los objetivos encomendados por el cliente.
- Fiabilidad: grado que se puede esperar de una aplicación lleve a cabo las operaciones especificadas y con la precisión requerida.
- Eficiencia: cantidad de recursos que necesita una aplicación para realizar las operaciones con los tiempos de respuesta adecuados.
- Integridad: grado con que puede controlarse el acceso al software o a los datos a personal no autorizado.
- Facilidad de uso: esfuerzo requerido para aprender el manejo de una aplicación, trabajar con ella, introducir datos y conseguir resultados.
- Portabilidad: esfuerzo requerido para transferir la aplicación a otro hardware o sistema operativo.
- Reusabilidad: grado en que partes de una aplicación pueden utilizarse en otras aplicaciones.
- Interoperabilidad: esfuerzo necesario para comunicar la aplicación con otras aplicaciones o sistemas informáticos.

### 2.5.2 Estrategia de Verificación

Una vez aceptado el producto se procede a la revisión del mismo. Primero se definen los tipos de pruebas que se llevarán a cabo, y luego determinar las técnicas, metodologías y herramientas a utilizar a la hora de realizar las pruebas.

Posteriormente se procede al Diseño de los Casos de Pruebas (Ver Anexo 6), utilizando los casos de uso definidos en el negocio del proyecto.

Definido esto se le hace entrega a cada miembro del equipo de calidad una copia del software que se va a revisar, comenzando de esta forma con la aplicación de las pruebas determinadas así como con las Listas de Chequeo (Ver Anexo 7), estas son:

- ✓ Lista de Chequeo de Seguridad (Ver Anexo 7.1)
- ✓ Lista de Chequeo para la Interfaz de Usuario (Ver Anexo 7.2)
- ✓ Lista de Chequeo para Plataformas Interactivas (Ver Anexo 7.3)
- ✓ Lista de Chequeo para Productos Exportables (Ver Anexo 7.4)

Los resultados no favorables detectados durante el proceso de revisión deben ser plasmados en el Registro de No Conformidades (Ver Anexo 8) y debe conformarse uno de estos registros para cada tipo de prueba aplicada.

Luego de esto se hace un informe con los resultados de las pruebas realizadas y se le entrega al equipo desarrollador del producto en cuestión, para que lo analicen y le realicen los cambios pertinentes.

Finalizado este proceso se recibe por parte de los desarrolladores el producto con los cambios realizados comenzando a partir de este momento un proceso cíclico, puesto que mientras el producto tenga errores se seguirá revisando. Es importante tener en cuenta en este punto la planificación de una nueva revisión, debido a que puede existir la posibilidad de que se estén revisando otros productos. En este caso, se realiza una reunión con los desarrolladores del producto de software en cuestión y se analizan los errores encontrados, se especifica el tiempo para la corrección de los errores y se estima un tiempo para recibir la nueva versión sin que el desarrollador se vea en la

necesidad de solicitar nuevamente una revisión de su producto y se exige para esta segunda iteración el documento Respuesta a las No Conformidades (Ver Anexo 9).

Cuando el software esté libre de errores se procederá a conformar un acta de liberación del producto para su posterior revisión por parte del Grupo de Calidad Central.

Es importante que la dirección del Laboratorio de Calidad guarde en una especie de bitácora, todo el trabajo realizado, dígame documentación oficial de los proyectos revisados, así como, plantillas, casos de uso de pruebas, copias de correos de trabajo, notas, etc.

### **2.5.2.1 Pruebas a Realizar**

La creciente inclusión del software como un elemento más de muchos sistemas y la importancia de los costos asociados a un fallo del mismo, han motivado la creación de pruebas más minuciosas y bien planificadas. Estas constituyen 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.

Las pruebas son el proceso de ejercitar un programa con la intención específica de encontrar errores previos a la entrega al usuario final. Es una actividad en la cual un sistema de componentes es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados y una evaluación es hecha de algún aspecto del sistema o componente.

#### **✓ Prueba de integración de los datos y la base de datos**

Asegura que los métodos y procesos de acceso a la base de datos funcionen correctamente y sin corromper datos.

Se debe invocar a cada método o proceso de acceso a la base de datos con datos válidos y no válidos e inspeccionar la base de datos para asegurarse de que se han guardado los datos correctos, que todos los eventos de la base de datos ocurrieron correctamente, o repase los datos devueltos para asegurar que se recuperaron datos correctos por la vía correcta.

Se deben usar bases de datos pequeñas para aumentar la facilidad de inspección de los datos para verificar que no sucedan eventos no aceptables.

### ✓ **Prueba de Funcionalidad**

Se enfoca en requerimientos para verificar que se corresponden directamente a casos de usos o funciones y reglas del negocio. Los objetivos de estas pruebas son verificar la aceptación de los datos, el proceso, la recuperación y la implementación correcta de las reglas del negocio. En resumen, debe asegurar la funcionalidad apropiada del objeto de prueba, incluyendo la navegación, entrada de datos, proceso y recuperación.

Durante la ejecución de esta prueba se deben usar datos válidos y no válidos, para verificar que:

- Se obtienen los resultados esperados cuando se usan datos válidos.
- Cuando se usan datos no válidos se despliegan los mensajes de error o advertencia apropiados.
- Se aplica apropiadamente cada regla del negocio.

Esta prueba se les aplica a los Instaladores, a la Aplicación y al Diseño de Casos de Pruebas.

### ✓ **Prueba de Ciclo del Negocio**

Esta prueba asegura que la aplicación funciona de acuerdo a los requerimientos del negocio. Debe simular ciclos de negocios realizando lo siguiente:

- Las pruebas de funcionalidad se deben modificar para aumentar la cantidad de veces que se ejecuta cada función, simulando varios usuarios diferentes en un período determinado.
- Todas las funciones sensibles a la fecha se deben ejecutar con fechas válidas y no válidas o períodos de tiempos válidos y no válidos.

### ✓ **Prueba de Interfaz de Usuario**

Con esta prueba se verifica que la interfaz de usuario proporcione al usuario el acceso y navegación a través de las funciones apropiada, es decir, que la navegación a través de los elementos que se

están probando reflejen las funciones del negocio y los requerimientos, incluyendo manejo de ventanas, campos y métodos de acceso; los objetos de las ventanas y características, como menús, tamaño, posición y estado, funcionen de acuerdo a los estándares.

Además asegura que los objetos presentes en la interfaz de usuario se muestren como se espera y conforme a los estándares establecidos.

El resultado de la ejecución de esta prueba será recogido en las listas de chequeo.

### ✓ **Prueba de Performance (Tiempo de Respuesta)**

En esta prueba se miden y evalúan los tiempos de respuesta, los tiempos de transacción y otros requerimientos sensitivos al tiempo. El objetivo de la prueba es verificar que se logren los requerimientos de performance.

La prueba de performance es implementada y ejecutada para poner a punto los destinos de pruebas de performance como función de condiciones de trabajo o configuraciones de hardware.

Para ello se debe:

- Usar procedimientos de prueba desarrollados para verificar funciones o ciclos de negocio.
- Modificar archivos de datos para aumentar el número de transacciones o los procedimientos de prueba para aumentar el número de iteraciones de ocurrencia de transacciones.
- Las pruebas se deben ejecutar en una máquina y se debe repetir con múltiples usuarios (virtuales o reales).

Las pruebas de performance deben incluir un trabajo de fondo en el servidor. Esto se puede realizar de distintas formas:

- Enviar transacciones directamente al servidor, generalmente en la forma de consultas (SQL).
- Crear usuarios virtuales para simular muchos clientes, generalmente varios cientos. Se pueden usar herramientas de Emulación de Terminar Remota para lograr este objetivo. Esta técnica también se usa para cargar la red con "tráfico".
- Usar muchos clientes físicos, cada uno corriendo procedimientos de prueba.

La prueba de performance se debe realizar en una máquina dedicada para permitir control total y medición exacta.

Las bases de datos usadas para las pruebas de performance deben tener un tamaño similar a las reales.

### ✓ **Prueba de Carga**

La prueba de carga somete los objetos a verificar a diferentes cargas de trabajo para medir y evaluar los comportamientos de performance y la habilidad de los objetos de continuar funcionando apropiadamente bajo diferentes cargas de trabajo.

El objetivo es determinar y asegurar que el sistema funciona apropiadamente en circunstancias de máxima carga de trabajo esperada. Además evaluar las características de performance, como tiempos de respuesta, tiempos de transacciones y otros elementos sensitivos al tiempo.

Es importante señalar que la prueba de carga debe realizarse en una máquina dedicada para tener control total y exactitud de mediciones, y las bases de datos usadas para la prueba deben tener un tamaño similar a las reales.

### ✓ **Prueba de Esfuerzo (stress, competencia por recursos, bajos recursos)**

La prueba de esfuerzo es un tipo de prueba de performance implementada y ejecutada para encontrar errores cuando hay pocos recursos o cuando hay competencia por recursos. Poca memoria o poco espacio de disco pueden revelar fallas en el software que no aparecen bajo condiciones normales de cantidad de recursos. Otras fallas pueden resultar al competir por recursos compartidos como bloqueos de bases de datos o ancho de banda de red. La prueba de esfuerzo también puede usarse para identificar el trabajo máximo que el software puede manejar.

El objetivo de la prueba de esfuerzo es también identificar y documentar las condiciones bajo las cuales el sistema falla y no continua funcionando apropiadamente.

Para probar recursos limitados, las pruebas se deben ejecutar en una sola máquina, y se debe reducir o limitar la memoria en el servidor. Para las pruebas de esfuerzo restantes, deber usarse múltiples clientes, cualquiera que ejecute las mismas pruebas o pruebas complementarias para producir el peor caso de volumen de operaciones.

Las pruebas de esfuerzo de red pueden requerir herramientas de red para cargar la red con mensajes o paquetes. La cantidad de disco del servidor usada por el sistema debe ser reducida temporalmente para restringir el espacio disponible para crecimiento de la base de datos. Sincronizar el acceso simultáneo de varios clientes accediendo a los mismos datos.

### ✓ Prueba de Volumen

Somete el software a grandes cantidades de datos para determinar si se alcanzan límites que causen la falla del software.

- Máximo (real o físicamente posible) número de clientes conectados, o simulados, todos realizando la misma operación (peor caso de operación) por un período de tiempo extenso.
- Máximo tamaño de base de datos y múltiples consultas ejecutadas simultáneamente.

La Prueba de Volumen identifica la carga máxima continua que puede manejar el software a prueba en un período dado.

Se deben usar múltiples clientes, ejecutando las mismas pruebas o pruebas complementarias para producir el peor caso de volumen de operaciones o mezcla en un período de tiempo extenso.

Se debe crear el tamaño máximo de base de datos (real, escalado o con datos representativos) y múltiples clientes ejecutando consultas simultáneamente por un período de tiempo extenso.

### ✓ Prueba de Seguridad y Control de Acceso

Esta prueba se enfoca en dos áreas de seguridad:

- Seguridad en el ámbito de aplicación, incluyendo el acceso a los datos y a las funciones de negocios: verifica que un actor pueda acceder solo a las funciones o datos para los cuales su

tipo de usuario tiene permiso. Para ello se debe identificar y hacer una lista de cada tipo de usuario y las funciones y datos sobre las que cada tipo tiene permiso; crear pruebas para cada tipo de usuario y verificar cada permiso creando operaciones específicas para cada tipo de usuario y modificar el tipo de usuario y volver a ejecutar las pruebas para los mismos usuarios. En cada caso, verificar que las funciones o datos adicionales están correctamente disponibles o son denegados.

- Seguridad en el ámbito de sistema, incluyendo conexión, o acceso remoto al sistema: verificar que solo los actores con acceso al sistema y a las aplicaciones, puedan acceder a ellos. Para esto el acceso al sistema debe ser discutido con el administrador del sistema o la red. Esta prueba no puede requerirse como tal, es una función del administrador del sistema o de la red.

### ✓ Prueba de Fallas y Recuperación

Estas pruebas aseguran que el software puede recuperarse de fallas de hardware, software o mal funcionamiento de la red sin pérdida de datos o de integridad de los datos.

La Prueba de Recuperación es un proceso en el cual la aplicación o sistema se expone a condiciones extremas, o condiciones simuladas, para causar falla, como fallas en dispositivos de Entrada/Salida o punteros a la base de datos inválidos. Los procedimientos de recuperación se invocan y la aplicación o sistema es monitoreado e inspeccionado para verificar que se recupera apropiadamente la aplicación o sistema y se logre la recuperación de datos.

En la prueba se incluyen los siguientes tipos de condiciones:

- Interrupción de energía al cliente.
- Interrupción de energía al servidor.
- Interrupción de comunicaciones mediante los servidores de la red.
- Interrupción de comunicación o pérdida de energía de los discos del servidor o con los controladores.
- Ciclos incompletos (procesos de filtro de datos interrumpidos, procesos de sincronización de datos interrumpidos).

- Punteros a la base de datos o claves inválidos.
- Elementos de datos en la base de datos inválidos o corruptos.

Es de suma importancia para la realización de estas pruebas simular las siguientes acciones, individualmente:

- Interrumpir la energía del cliente: apagar el PC.
- Interrumpir la energía del servidor: simular o iniciar el proceso de apagado del servidor.
- Interrupción por medio de los servidores de red: simular o iniciar la pérdida de comunicación con la red (desconectar físicamente la comunicación o apagar el servidor de red o router.
- Interrumpir la comunicación o quitar la energía de los discos del servidor o sus controladores: simular o eliminar físicamente la comunicación con uno o más controladores de disco o los discos.
- Una vez que se lograron o simularon estas condiciones, se deben invocar los procedimientos de recuperación.
- Las pruebas de ciclos incompletos utilizan la misma técnica excepto que los procesos de bases de datos deben ser abortados a sí mismos o terminados prematuramente.
- Las últimas dos pruebas requieren que se logre un estado conocido de la base de datos. Se deben corromper manualmente campos de la base de datos, punteros y claves trabajando directamente sobre la base de datos (utilizando herramientas para la base de datos). Se deben ejecutar las pruebas de Funcionalidad y Ciclo de negocio y verificar que los ciclos se completen.

Los procedimientos para desconectar cables (simulando falta de energía o pérdida de comunicación) no son deseables o factibles. Se pueden requerir métodos alternativos, como software de diagnóstico. Se requieren los grupos de recursos de Sistemas, Bases de datos y Red.

Estas pruebas deben ejecutarse fuera del horario de trabajo normal o en una máquina aislada.

### ✓ **Prueba de Configuración**

La Prueba de Configuración verifica el funcionamiento del software con diferentes configuraciones de software y hardware.

Para la realización de esta prueba se debe:

- Abrir y cerrar varias sesiones de software que no son objeto de prueba, como parte de la prueba o antes de comenzar la prueba.
- Ejecutar operaciones seleccionadas para simular la interacción del actor con el software objeto de prueba y con el software que no es objeto de prueba.
- Repetir los procedimientos anteriores minimizando la memoria convencional disponible en la máquina cliente.

Los sistemas, red, servidores de red, bases de datos, etc., deben ser documentados como parte de esta prueba.

### ✓ **Prueba de Instalación**

Tiene dos propósitos:

- Asegurar que el software puede ser instalado en diferentes condiciones (como una nueva instalación, una actualización, y una instalación completa o personalizada) bajo condiciones normales y anormales. Condiciones anormales pueden ser insuficiente espacio en disco, falta de privilegios para crear directorios, etc.
- Verificar que, una vez instalado, el software opera correctamente. Esto significa normalmente ejecutar un conjunto de pruebas que fueron desarrolladas para Prueba de Funcionalidad.

### ✓ **Prueba de Documentos**

Esta prueba debe asegurar que los documentos relacionados al software que se generen en el proceso sean correctos, consistentes y entendibles. Se incluyen como documentos los Materiales para Soporte al Usuario, Documentación Técnica, Ayuda en Línea y todo tipo de documento que forme parte del paquete de software.

Para ello se hace necesario que el documento objeto de prueba sea:

- Correcto, esto es, que cumpla con el formato y organización para el documento establecido en el proyecto.

- Consistente, esto es, que el contenido del documento sea fiel a lo que hace referencia. Si el documento es Documentación de Usuario, que la explicación de un procedimiento sea exactamente como se realiza el procedimiento en el software, si se muestran pantallas que sean las correctas.
- Entendible, esto es, que al leer el documento se entienda correctamente lo que expresa y sin ambigüedades, además que sea fácil de leer.

Para verificar que el documento es correcto se debe comparar con el estándar definido si existe o con las pautas de documentación y ver que el documento cumple con ellas.

Para verificar que el documento es consistente se debe ejecutar el programa siguiendo el documento Materiales de Soporte al Usuario y comprobar que lo que se explica en estos documentos es exactamente lo que se ejecuta en el programa. En caso de Documentación Técnica se debe revisar el código al cual corresponde la documentación y comprobar que dicha describe el código.

Para verificar que el documento es entendible, debe comprobar que se entiende correctamente, que no tiene ambigüedades y que sea fácil de leer.

Teniendo en cuenta lo anteriormente expuesto, los pasos a seguir durante las pruebas a la documentación son:

- Revisión del Manual de Usuario contra la Aplicación.
- Revisión del Manual de Instalación contra la Aplicación.
- Revisión de la Ayuda contra la Aplicación.

### **2.5.3 Evaluación del Software**

El desarrollo de productos de software seleccionados y de alta calidad es un tema de primera importancia. Una amplia especificación y evaluación de la calidad de los productos de software es un factor cardinal en el aseguramiento de una calidad adecuada. Ello puede ser alcanzado definiendo las características de calidad más apropiadas, tomando en cuenta el propósito del empleo del producto de software en cuestión. Resulta importante que todas las características de calidad relevantes del

producto de software sean especificadas y evaluadas, siempre que sea posible utilizando métricas validadas o comúnmente aceptadas.

La ISO/IEC 9126(1991) “Software product evaluation – Quality characteristics and guidelines for their use”, la que fue elaborada para sustentar este propósito, definió seis características de calidad y describió un modelo de evaluación de los productos de software.

Esta norma ha sido reemplazada por la ISO/IEC 9126:2005 “Calidad de los productos de Software”. La misma permite especificar y evaluar la calidad del producto de software desde las perspectivas de aquellos asociados con la adquisición, regulación, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoria del software. Por ejemplo puede ser utilizada por los programadores, los clientes, el personal de aseguramiento de la calidad y los evaluadores independientes, particularmente los responsables de especificar y evaluar la calidad de los productos de software. El modelo de calidad definido en esta parte de la NC ISO/IEC 9126 puede usarse para:

- Validar la integridad de la definición de los requisitos.
- Identificar los requisitos del software.
- Identificar los objetivos del diseño del software.
- Identificar los objetivos de ensayo del software.
- Identificar los criterios de aseguramiento de la calidad.
- Identificar los criterios de aceptación para un producto de software terminado.

Este modelo de calidad se usará al fijar los objetivos de calidad para los productos del software. La calidad del producto de software se debe desglosar jerárquicamente en un modelo de calidad compuesto de características y sub-características que puede usarse como una lista de chequeo de chequeo de problemas relacionados con la calidad.

### **2.5.3.1 Calidad Interna y Externa**

El modelo de calidad interna y externa categoriza los atributos de calidad del software en seis características: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y transportabilidad,

que a su vez son divididas en sub-características que pueden medirse a través de métricas internas o externas.

**1. Funcionalidad:** es la capacidad del software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas cuando el software se usa bajo las condiciones especificadas, está relacionada con lo que hace el software para satisfacer las necesidades, al tiempo que las otras características principalmente están relacionadas con el cuándo y cómo lo hace.

- Para las necesidades declaradas e implícitas en esta característica, la nota a la definición de calidad se aplica, (ver Anexo B.21 de NC ISO/IEC 9126).

- Para un sistema que se opera por un usuario, la combinación de funcionalidad, la confiabilidad, usabilidad y eficiencia pueden ser medidas externamente por la calidad en el uso (ver apartado 7 de NC ISO/IEC 9126).

- **Idoneidad:** capacidad del software para mantener un conjunto apropiado de funciones para las tareas y los objetivos del usuario especificados.
- **Precisión:** capacidad del software para proporcionar efectos o resultados correctos o convenidos con el grado de exactitud necesario.
- **Interoperabilidad:** capacidad del producto de software para interactuar recíprocamente con uno o más sistemas especificados. Se usa en lugar de la compatibilidad para evitar la posible ambigüedad con la reemplazabilidad (ver apartado 6.6.4 de NC ISO/IEC 9126).
- **Seguridad (informática):** capacidad del producto de software para proteger información y los datos, para que personas o sistemas desautorizados no puedan leer o pueden modificar los mismos, y las personas o sistemas autorizados tenga el acceso a ellos. Esto se aplica también a los datos en transmisión.

**2. Confiabilidad:** capacidad del producto de software para mantener un nivel de ejecución especificado cuando se usa bajo las condiciones especificadas.

- El software no sufre desgaste ni envejecimiento. Las limitaciones en la fiabilidad son debidas a los fallos en los requisitos, el diseño y la implementación. Los fallos totales debidos a estos fallos dependen de la manera en que el producto del software se utilice y las opciones del programa seleccionado y no del tiempo de uso transcurrido.

- **Madurez:** capacidad del producto de software de evitar un fallo total como resultado de haberse producido un fallo del software.
- **Tolerancia ante fallos:** capacidad del producto de software de mantener un nivel de ejecución o desempeño especificado en caso de fallos del software o de infracción de su interfaz especificada. Un nivel de ejecución especificado puede incluir la falta la capacidad de seguridad ante errores.
- **Recuperabilidad:** capacidad del producto de software de restablecer un nivel de ejecución especificado y recuperar los datos directamente afectados en caso de fallo total. Después de un fallo total, un producto del software a veces estará desactivado por un cierto periodo de tiempo, cuyo plazo se evalúa a partir de su recuperabilidad. La disponibilidad es la capacidad del producto de software de estar en condiciones de realizar una función requerida en un momento dado en el tiempo, bajo las condiciones declaradas de uso. Externamente, la disponibilidad puede ser evaluada por la proporción de tiempo total durante el cual, el producto del software está en estado activo y listo para su uso. La disponibilidad es por consiguiente una combinación de madurez (que determina la frecuencia de los fallos totales), la tolerancia ante fallos y la recuperabilidad (que determina el plazo de tiempo de desactivación, en que está de baja, que sigue a cada fallo total). Por ello no ha sido incluida como una sub-característica independiente.

**3. Usabilidad:** capacidad del producto de software de ser comprendido, aprendido, utilizado y de ser atractivo para el usuario, cuando se utilice bajo las condiciones especificadas. Algunos aspectos de funcionalidad, confiabilidad y eficiencia también influirán la usabilidad, pero para los propósitos de la NC ISO/IEC 9126 no son clasificados dentro de la usabilidad. Como usuarios pueden estar incluidos operadores, usuarios finales y usuarios indirectos que están bajo la influencia o dependientes del uso del software. La usabilidad debe abordar todos los ambientes del usuario que el software puede afectar, lo cual puede incluir la preparación para el uso y la evaluación de resultados.

- **Comprensibilidad:** capacidad del producto de software para permitirle al usuario entender si el software es idóneo, y cómo puede usarse para las tareas y condiciones de uso particulares. Esto dependerá de la documentación y la impresión inicial que ofrezca el software.

- **Cognoscibilidad:** capacidad del producto del software para permitirle al usuario aprender su aplicación. Los atributos internos se corresponden con la posibilidad del aprendizaje como está definido en la ISO 9241-10.
- **Operabilidad:** capacidad del producto del software para permitirle al usuario operarlo y controlarlo. Aspectos de idoneidad, flexibilidad, adaptabilidad e instalabilidad pueden influir en la operabilidad. La operabilidad está en correspondencia con la capacidad de ser controlado, la tolerancia ante errores y la conformidad con las expectativas del usuario como está definido en la ISO 9241-10. Para un sistema que se opera por un usuario, la combinación de funcionalidad, confiabilidad, usabilidad y eficiencia pueden ser medidas externamente por la calidad en el uso.
- **Atracción:** capacidad del producto del software de ser atractivo o amigable para el usuario. Esto se refiere a los atributos del software que se aplican para hacer el software más atractivo al usuario, tales como el uso del color y la naturaleza del diseño gráfico.

**4. Eficiencia:** capacidad del producto de software para proporcionar una ejecución o desempeño apropiado, en relación con la cantidad de recursos utilizados usados, bajo condiciones establecidas. Entre los recursos se pueden incluir otros productos del software, la configuración del software y el hardware del sistema y los materiales, (por ejemplo el papel de la impresión o disquetes). Para un sistema que se opera por un usuario, la combinación de funcionalidad, confiabilidad, usabilidad y eficiencia puede ser medida externamente por la calidad en el uso.

- **Rendimiento:** capacidad del producto de software para proporcionar apropiados tiempos de respuesta y procesamiento, así como tasas de producción de resultados, al realizar su función bajo condiciones establecidas.
- **Utilización de recursos:** capacidad del producto de software para utilizar la cantidad y el tipo apropiado de recursos cuando el software realiza su función bajo las condiciones establecidas. Los recursos humanos se incluyen como parte de la productividad (ver apartado 7.1.2 de NC ISO/IEC 9126).

**5. Mantenibilidad:** capacidad del producto de software de ser modificado. Las modificaciones pueden incluir las correcciones, mejoras o adaptaciones del software a cambios en el ambiente, así como en los requisitos y las especificaciones funcionales.

- **Diagnosticabilidad:** capacidad del producto del software de ser objeto de un diagnóstico para detectar deficiencias o causas de los fallos totales en el software, o para identificar las partes que van a ser modificadas.
- **Flexibilidad:** capacidad del producto del software para permitir la aplicación de una modificación especificada. La aplicación incluye cambios en el código, diseño y documentación. Si el usuario final va a modificar el software, la flexibilidad puede influir en la operabilidad.
- **Contrastabilidad:** capacidad del producto del software para permitir la validación de un software modificado.
- **Estabilidad:** capacidad del producto de software para minimizar los efectos inesperados de las modificaciones realizadas al software.

**6. Portabilidad:** capacidad de producto de software de ser transferido de un ambiente a otro. El ambiente puede incluir el ambiente del software, del hardware u organizacional.

- **Adaptabilidad:** capacidad del producto de software de ser adaptado a los ambientes especificados sin aplicar acciones o medios de otra manera que aquellos suministrados con el propósito de que el software cumpla sus fines. La adaptabilidad incluye el escalado de la capacidad interna (por ejemplo los campos de la pantalla, las tablas, los volúmenes de transacción o los formatos de informes.). Si el software será adaptado por el usuario final, la adaptabilidad corresponde a la idoneidad de la personalización como está definido en la ISO 9241-10, y puede influir en la operabilidad.
- **Instalabilidad:** capacidad del producto de software de ser instalado en un ambiente especificado. Si el software será instalado por un usuario final, la instalabilidad puede influir en la idoneidad y la operabilidad.
- **Coexistencia:** capacidad del producto de software de coexistir con otro software independiente en un ambiente común y compartir los recursos comunes
- **Reemplazabilidad:** capacidad del producto de software de ser usado en lugar de otro producto de software especificado para los mismos fines y en el mismo ambiente. Por ejemplo, la reemplazabilidad de una nueva versión de un producto de software es importante para el usuario cuando va a actualizar su sistema. La reemplazabilidad se usa en lugar de la

compatibilidad para evitar la posible ambigüedad con la interoperabilidad (ver apartado 6.1.3 de NC ISO/IEC 9126). La reemplazabilidad puede incluir atributos de instalabilidad como de la adaptabilidad. El concepto se ha introducido como una sub-característica por si mismo debido a su importancia.

- **Conformidad con la portabilidad:** capacidad del producto de software de adherirse a las normas o convenciones relativas a la portabilidad.

### **2.5.3.2 Calidad Durante el Uso**

La calidad en el uso es la visión del usuario de la calidad. El logro de la calidad durante el uso depende del logro de la calidad externa necesaria, que a su vez depende del logro de la necesaria calidad interna. Los atributos de la calidad durante el uso se clasifican en cuatro características: eficacia, productividad, seguridad y satisfacción.

Además la calidad durante el uso: capacidad del producto de software de permitir que los usuarios especificados alcancen los objetivos especificados con efectividad, productividad, seguridad y satisfacción en contextos de uso especificados.

La calidad durante el uso es la perspectiva que tiene el usuario de la calidad de un sistema que contiene el software, y es medida en lo que se refiere al resultado de usar el software, en lugar de las propiedades del propio software.

La usabilidad se define en la ISO 9241-11 de forma similar a la de calidad en el uso de esta parte de la NC-ISO/IEC 9126. La calidad en el uso puede recibir la influencia de cualquiera de las características de la calidad, y por tanto es más amplia que la usabilidad, la cual se define en esta parte de la NC-ISO/IEC 9126 en términos de comprensibilidad, cognoscibilidad, operabilidad, atracción y conformidad.

**1. Eficacia:** capacidad del producto de software de permitir que los usuarios logren objetivos especificados con precisión e integridad en un contexto especificado.

**2. Productividad:** capacidad del producto de software de permitir que los usuarios dediquen una cantidad de recursos apropiada en relación con la eficacia alcanzada en un contexto de uso especificado. Entre los recursos se pueden incluir el tiempo para completar la tarea, los esfuerzos del usuario, los materiales o el costo de utilización en términos financieros.

**3. Seguridad:** capacidad del producto de software de alcanzar niveles aceptables de riesgo de daños a las personas, el negocio, el software, la propiedad o el ambiente en un contexto de uso especificado. Por lo general los riesgos son ocasionados por deficiencias en la funcionalidad (incluyendo la seguridad informática), confiabilidad, usabilidad o mantenibilidad.

**5. Satisfacción:** capacidad del producto de software de satisfacer a los usuarios en un contexto de uso especificado. La satisfacción es la respuesta del usuario a la interacción con el producto, e incluye la actitud hacia el uso del producto.

### **2.5.3.3 Tareas de la Comisión de Evaluación**

La comisión de evaluación tiene que ser muy exigente a la hora de realizar su trabajo, ya que es quien avala la calidad de la revisión efectuada por el equipo de trabajo.

El equipo evaluador es el encargado durante el proceso de evaluación de clasificar los errores encontrados por los desarrolladores de las pruebas según su nivel de gravedad. Asignan un nivel de gravedad a los errores para poder capturar de alguna manera su impacto en el sistema. A continuación se muestran los cuatro niveles diferentes de gravedad de error:

- **Catastrófico:** un error cuya presencia impide el uso del sistema.
- **Crítico:** un error cuya presencia causa la pérdida de una funcionalidad crítica del sistema. Si no se corrige el sistema no satisfará las necesidades del cliente.
- **Marginal:** un error que causa un daño menor, produciendo pérdida de efectividad, pérdida de disponibilidad o degradación de una funcionalidad que no se realiza fácilmente de otra manera.

- **Menor:** un error que no causa perjuicio al sistema, pero que requiere mantenimiento o reparación. No causa pérdida de funcionalidades que no se puedan realizar de otra manera.

Esto niveles de gravedad del error quedan reflejados en el Informe Final que emite el equipo evaluador.

Otras de las tareas que realiza el equipo evaluador otorgarle un nivel de aceptación a los elementos verificados para poder establecer el estado en el que se encuentra el proyecto. Los niveles de aceptación definidos son:

- **No aprobado:** el elemento verificado tiene errores catastróficos (uno o varios) que impiden su uso o tiene errores críticos (uno o varios) que hacen que el elemento verificado no sea confiable. El usuario no puede depender de él para realizar el trabajo.
- **Aprobado con Observaciones:** el elemento verificado no tiene errores catastróficos, ni errores críticos, pero tiene errores marginales (uno o varios) que hacen que el elemento de software se degrade en algunas situaciones.
- **Aprobado:** el elemento verificado no tiene errores o tiene errores menores que no afectan el normal funcionamiento del elemento.

Estos niveles tanto de aceptación a los elementos verificados como de gravedad del error son definidos en los Cuestionarios de Evaluación del Producto de Software (Ver Anexo 10), estos son varios:

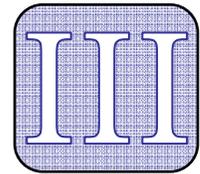
- ✓ Cuestionario Evaluación Introdutorio (Ver Anexo 10.1)
- ✓ Cuestionario Evaluación sección A: valoración de los resultados de las pruebas y el comportamiento de las características de calidad. (Ver Anexo 10.2)
- ✓ Cuestionario Evaluación sección B: Valoración de las comprobaciones documentales. (Ver Anexo 10.3)
- ✓ Cuestionario Evaluación sección C: Valoración de las Pruebas comprobaciones (Ver Anexo 10.4)
- ✓ Cuestionario Resumen General (Ver Anexo 10.5)

## **2.6 Conclusiones Parciales**

Con la aplicación de este procedimiento se logra una mejor estructura organizativa del personal del Laboratorio de Calidad de la Facultad 7 y se mejora la seguridad y confiabilidad de los productos revisados dentro del mismo.

Usando las flexibilidades que ofrece la Norma ISO/IEC 9126-1:2005, se conformaron los Cuestionarios de Evaluación del software para ser usados por los miembros del equipo evaluador, y de esta forma mejorar la situación actual que existe en el laboratorio.

El procedimiento propuesto anteriormente queda estandarizado según el modelo definido por la Norma ISO 9000 del 2005 (Ver Anexo 11)



---

## **Capítulo 3. Análisis comparativo entre el proceso de revisión actual y lo que se propone.**

En este último capítulo se procede a hacer un análisis comparativo del proceso actual de revisión y evaluación del software, en el Laboratorio de Calidad de la Facultad 7 de la UCI, con respecto al proceso propuesto, a partir de la puesta en marcha del procedimiento bajo la Norma Cubana ISO/IEC 9126-1:2005. Se hace un estudio comparativo de la evaluación del proceso en las condiciones actuales del laboratorio y los resultados que se obtendrían si se aplicara el procedimiento vigente en el laboratorio propuesto.

La puesta en práctica del procedimiento propuesto, permitirá que los software sean probados, evaluados y lleguen a sus proveedores con una mayor garantía de calidad; demostrándose la eficiencia y competitividad de la empresa que lo desarrolló, al resolver gran parte de las inconformidades y problemáticas vistas en el laboratorio. En dicho laboratorio se hace actualmente la revisión de todos los productos de software terminados, sin una evaluación definida del mismo.

### **3.1 Estructura Organizativa**

La no definición de todos los roles que debían existir en el Laboratorio de Calidad provocó que no existiera el Equipo de Evaluación por lo que en ningún momento se evaluó la revisión realizada por parte del equipo desarrollador de las pruebas. Tampoco se contó con una persona experta en la materia relacionada con el perfil del software por lo que las salidas y las validaciones pudieron ser incorrectas y aún así el software fue liberado.

Al no existir un Equipo de Evaluación integral no hay:

- Un seguimiento de la rectificación de los errores ni una evaluación de la envergadura del mismo.
- No existe un diseño de pruebas acorde a la situación actual del sistema que cumple con todos los requisitos de todas las características (Funcionalidad, Usabilidad. etc.) a evaluar.
- No existe un informe de No Conformidades y errores con sus respectivas trazas de tal forma que una segunda vuelta del sistema al Laboratorio de Calidad de la Facultad 7 sirva de guía para analizar si aún persisten estos errores.

La comunicación Equipo Evaluador y Desarrollador es diversa, cada quien expone al desarrollador sin antes consultarse un informe único que permite incluso dar sugerencias de soluciones.

De forma general el personal que realizó las pruebas no cuenta con habilidades tan elementales como: la experiencia en la administración de proyectos; para el análisis de riesgos; conocer los tipos de pruebas; definir defecto, tipo de defectos, registrarlos y reportarlos; así como habilidades para documentar los errores con claridad y precisión.

### **3.2 Documentación.**

Para la revisión de un software determinado se llevan a cabo tantas revisiones como hagan falta en cada una de ellas se entrega una documentación diferente, en la primera revisión se recibe por parte de los desarrolladores del producto: la descripción textual de los casos de uso, los requerimientos funcionales y no funcionales y la aplicación.

Según lo planteado en el procedimiento propuesto se debe recibir: el Expediente del Producto de Software que incluye la Descripción del Producto (Requisitos de software y hardware, Requisitos de documentación, Requisitos de personal y Otros) y los Programas y datos necesarios; la Descripción de los Casos de Uso; la Especificación de Requisitos (Funcionales y no Funcionales); el Manual de Instalación; el Manual de Usuario o Ayuda On-Line; la Instalación del Software y la Ayuda.

### **3.2.1 Documentos generados y no generados durante el proceso de revisión del producto**

El hecho que no se haya entregado toda la documentación que plantea el procedimiento trajo consigo consecuencias tales como:

- No tener una información organizada, imposibilita mantener una disciplina con respecto a la documentación.
- El hecho de no contar con un acta de entrega puede traer contradicciones con el cliente.
- El cliente puede que no sepa cómo instalar el software debido a la complejidad del mismo y la no existencia del Manual de Instalación es una deficiencia del mismo. Este manual de instalación no solo permite que se conozca como instalar el producto, sino que se revisa y se pueden probar los accesos permisibles, las demoras de respuestas, y problemáticas que pueden ocurrir en la propia instalación.

La documentación generada durante el proceso de revisión no estaba lo debidamente organizada, por ejemplo, los Registro de No Conformidades no están organizados, pues en ocasiones no cumplen con los estándares de nombramientos de los ficheros.

En un registro se pueden encontrar diferentes tipos de no conformidades, relacionadas tanto con el manual de usuario como en la interfaz. Esto trae como consecuencia que si en algún momento se quiere consultar esta documentación para obtener estadísticas de la cantidad de errores encontrados en la documentación con respecto a la cantidad de errores encontrados en la interfaz se hace casi imposible por estar todos los errores mezclados en un mismo documento.

Por cuestiones organizativas este proceso se debe realizar según lo planteado en el procedimiento propuesto.

A pesar de que una vez finalizado el proceso de revisión del software se obtiene una sólida documentación, esta no era la más correcta ya que en las mismas se encontraron deficiencias tales como información repetida, la documentación generada no tenía un nombre estándar según lo propuesto y no se encontraba correctamente organizada.

Si la revisión de este producto de software se hubiese realizado según lo planteado en el procedimiento el proceso de aplicación de las pruebas al producto hubiese generado documentos sumamente importantes tales como:

- Listas de Chequeo:
  - Lista de Chequeo para la Interfaz de Usuario.
  - Lista de Chequeo para Productos Exportables.
  - Listas de Chequeo para el Manual de Usuario
  - Lista de Chequeo para Plataformas Interactivas.
- Respuestas a No Conformidades.

Con las Listas de Chequeo se verifican elementos muy importantes que no se pueden probar durante las pruebas que se definen. Estos documentos constituyen un punto clave de referencia para cualquier persona interesada en verificar si se le realizaron correctamente las pruebas al software.

El documento de Respuesta a No Conformidades no es imprescindible para que la aplicación funcione correctamente, pero mediante el mismo se logra que la comunicación entre los desarrolladores y el equipo de calidad fluya mejor, además de agilizar el proceso de revisión de las no conformidades. En este documento es plasmado la opinión del desarrollador con respecto a los errores encontrados por parte del grupo de calidad, y especifica si el error fue corregido o no y por qué.

### **3.3 Entrada del Producto al Laboratorio**

Cuando se hace entrega de un producto por parte de los desarrolladores al Laboratorio de Calidad, este es recibido por el Líder del Proyecto de Calidad.

De haberse realizado lo planteado en este trabajo lo primero que se debió hacer por parte de los desarrolladores del producto en cuestión fue una solicitud de revisión de su aplicación a través del:

- Formulario de Solicitud de Revisión de Producto de Software.

Una vez aceptada esta solicitud el líder de Calidad debió planificar la entrada del producto a revisar y posteriormente proceder a la firma del:

- Acta de Entrega del Producto
- Acuerdo de Confidencialidad

Firmándose este último por todos los miembros del equipo de calidad.

### **3.4 Seguridad del producto dentro del Laboratorio**

El sistema a revisar en unas ocasiones se instala en el servidor del Laboratorio y en otras se instala en todas las maquinas con que cuenta el mismo.

El servidor del Laboratorio de Calidad no solo se utiliza para soportar los sistemas que se revisan, sino que cumple también con la función de almacenar toda la información que se genera en el laboratorio, información que debe ser confidencial y que solo la debe manejar el Líder del Proyecto de Calidad y el Jefe del Laboratorio de Calidad por cuestiones de seguridad de la información puesto que la mayoría de la documentación que se puede encontrar allí refleja errores que contiene la aplicación. Cualquier miembro de este laboratorio puede acceder al servidor mediante una contraseña y usuario determinado.

Si todas estas operaciones se realizan según lo planteado en el procedimiento se evita cometer errores relacionados con la seguridad y confidencialidad del producto. En un principio señalar de

manera significativa que para revisar cualquier producto el mismo debe estar instalado en Laboratorio de Calidad simulando que se es el cliente final. Otra cuestión importante es que debe existir una copia de la información guardada en el servidor para que de esta manera, ante cualquier fallo en el disco duro del servidor, la información esté debidamente conservada. El acceso a este servidor debe ser restringido para evitar pérdida o cambio de la información.

También se hace necesario la existencia de un archivo físico donde se guarde toda la información impresa de los productos revisados en el laboratorio previniendo así que por un error en el sistema o ajeno a la voluntad del hombre no se pueda acceder a una información determinada que se necesite en un momento dado.

### ***3.5 Hardware y Software.***

En lo relacionado con el hardware y software se tiene que a pesar de que el Laboratorio de Calidad cuenta con algunas de las condiciones mínimas necesarias para trabajar, requiere de un servidor con amplia capacidad de almacenamiento que no existe.

Este laboratorio se adapta a las condiciones de software que requiere el sistema a revisar, pero no logra simular las condiciones de hardware que el producto necesita para su óptimo funcionamiento.

El procedimiento propuesto anteriormente plantea la importancia de simular todas las condiciones del producto que se pretende revisar tanto de hardware como de software, de esta forma se evita que una vez implantado el mismo no ocurran eventos indeseados con respecto a esto, por tanto al no probarse el sistema bajo estas condiciones de hardware que requiere pueden ocurrir errores catastróficos durante su puesta en marcha.

### ***3.6 Estrategia de Verificación.***

Teniendo en cuenta las funcionalidades que se definen para que el sistema a revisar sea óptimo el equipo de trabajo del Laboratorio de Calidad define cuáles deben ser las pruebas de calidad idóneas para probar el software. Las pruebas que se definen son: pruebas exploratorias, pruebas al diseño de casos de prueba, pruebas a la documentación, pruebas de carga y se aplicaron unas listas de

chequeo para verificar la seguridad del producto. Todas estas pruebas se definen con el objetivo de verificar que:

- La aplicación funcionara correctamente.
- La documentación estuviese libre de errores.
- Los mensajes emitidos por la aplicación estuvieran correctos.
- Las entradas estén bien validadas.
- Encontrar defectos en el software.
- Ver si cumple las especificaciones de diseño.
- Ver si cumple los requisitos del análisis

Estas pruebas se realizan de manera tal que se cumplan los requisitos funcionales que plantea el producto de software.

El Equipo de Calidad Interno efectúa algunas pruebas de las que se plantean en el procedimiento aunque no concientes de su definición. Por ejemplo, cuando se prueba la accesibilidad de los diferentes tipos de usuarios al sistema, se están realizando **pruebas de seguridad y control de acceso**, verificando que cada usuario pueda acceder sólo a las funciones o datos para los cuales su tipo de usuario tenía permiso. Además prueban que el sistema esté o no protegido contra accesos no autorizados utilizando mecanismos de autenticación y autorización que pueden garantizar el cumplimiento de esto: usuario, contraseña dado por la asignación de roles, de manera que cada uno tenga datos de acceso propios.

Otra prueba que se realiza en el laboratorio es la de **interfaz de usuario** que les permite a los desarrolladores de las pruebas verificar que la interfaz de usuario proporciona al mismo el acceso y navegación deseado por él. Además este equipo se asegura que los objetos presentes en la interfaz de usuario se muestren como se espera y conforme a los estándares establecidos (interfaz sencilla, agradable, legible y de fácil uso para el planificador, contenido comprensible y fácil de leer).

Todas estas pruebas mencionadas hasta el momento, que coinciden con las propuestas en el procedimiento, son realizadas de manera implícita dentro de las pruebas exploratorias.

Las **pruebas de carga** son realizadas también para determinar y asegurar que el sistema funciona apropiadamente en circunstancias de máxima carga de trabajo esperada.

A la **documentación** se le realiza la mayor cantidad de pruebas enfocándose fundamentalmente en el manual de usuario.

Con la aplicación de estas pruebas a los productos de software se encuentran errores, estos deben documentarse para tenerse en cuenta y corregirlos posteriormente por parte de los desarrolladores; todo este proceso de pruebas genera documentos tales como:

- Registro de No Conformidades.
- Diseño de Casos de Prueba.
- Listas de Chequeo de Seguridad.

A pesar de que faltan múltiples pruebas por realizarle a la aplicación, el producto se libera aunque no se descarta que una vez puesto en marcha el software surjan errores. Se considera que durante el proceso de aplicación de las pruebas no se tienen en cuenta una serie de pruebas fundamentales para la correcta verificación de las funcionalidades del sistema que se plantean en el procedimiento.

Otras pruebas que se deben aplicar:

- **Prueba de integridad de los datos y la base de datos:** con ella se verifica que no sucedan eventos no aceptables y asegura que los métodos y procesos de acceso a la base de datos funcionen correctamente y sin corromper datos. El hecho de que no se aplique esta prueba trae grandes repercusiones en el futuro debido a que pueden existir errores en la base de datos así como ocurrir eventos no deseados nunca antes verificados.
- **Prueba de Funcionalidad:** asegura la funcionalidad apropiada del objeto de prueba, incluyendo la navegación, entrada de datos, proceso y recuperación, usando para esto datos válidos y no válidos logrando así que se desplieguen los mensajes de error o advertencia apropiados. Esta prueba se realiza como una prueba exploratoria, validándose sin mucha profundidad algunos datos de entrada como los mensajes de error emitidos por el sistema. Se

debe profundizar aún más en este tipo de prueba puesto que es una de las más importantes porque verifica el funcionamiento correcto de la aplicación.

- **Prueba de Ciclo del Negocio:** con esta prueba se asegura que la aplicación funciona de acuerdo a los requerimientos funcionales del negocio. Con la realización de esta prueba se verifica que se dió cumplimiento a todas las exigencias del cliente, por lo que si no se realiza existe la posibilidad de que no funcionen correctamente los casos de uso relacionados con estos requisitos funcionales.
- **Prueba de Performance (tiempo de respuesta):** verifica que se logren los requerimientos de performance, es decir, que el sistema sea rápido ante las solicitudes de los planificadores y en el procesamiento de la información. La eficiencia de la aplicación estará determinada en gran medida por el aprovechamiento de los recursos que se disponen en el modelo Cliente/Servidor, y la velocidad de las consultas a la base de datos. Se realizará la validación de los datos en el cliente y en el servidor aquellas que por cuestiones de seguridad, o de acceso a los datos lo requieran. Lográndose así un tiempo de respuesta más rápido, una mayor velocidad de procesamiento, y un mayor aprovechamiento de los recursos.
- **Prueba de Esfuerzo (stress, competencia por recursos, bajos recursos):** es un tipo de prueba de performance, además mediante ella se encuentran los errores cuando hay pocos recursos o cuando hay competencia por recursos, así como revelan fallas en el software que no aparecen bajo condiciones normales de cantidad de recursos cuando hay poca memoria o poco espacio de disco. También identifica y documenta las condiciones bajo las cuales el sistema falla y no continúa funcionando apropiadamente. Esta prueba no se realiza porque los estudiantes no tienen acceso a modificar la capacidad ni del disco duro ni de las memorias, pero de realizarse un trabajo de mesa anteriormente, es decir, una planificación correcta de las pruebas a realizarse, el equipo de dirección del proyecto puede planificar con la Dirección de Laboratorios para que les posibilite los medios necesarios para la realización de dicha prueba. Como todos estos elementos no funcionan y por tanto no se efectúa en ningún momento esta prueba, es posible que una vez implantada la aplicación comiencen a aparecer problemas tales como: poco espacio en el disco o en la memoria.

- **Prueba de Volumen:** somete el software a grandes cantidades de datos para determinar si se alcanzan límites que causen la falla del software, además identifica la carga máxima continua que puede manejar el software a prueba en un período dado. Con esta prueba se debe probar que el servidor de bases de datos soporta los esperados volúmenes de datos y una velocidad de procesamiento óptima. Además de no cumplirse con la velocidad de acceso y los volúmenes requeridos, se emitirán medidas o sugerencias técnicas al respecto.
- **Prueba de Fallas y Recuperación:** asegura que el software puede recuperarse de fallas de hardware, software o mal funcionamiento de la red sin pérdida de datos o de integridad de los datos. Puede darse bajo condiciones tales como: Interrupción de energía al cliente o al servidor, interrupción de comunicaciones mediante los servidores de la red, interrupción de comunicación o pérdida de energía de los discos del servidor. Si se efectúa esta prueba se verifica cómo se recupera el sistema cuando se estén realizando actividades.
- **Prueba de Configuración:** verifica el funcionamiento del software con diferentes configuraciones de software y hardware. Se prueba abriendo y cerrando varias sesiones de software que no son objeto de prueba, como parte de la prueba o antes de comenzar la prueba. Esta aunque no es una prueba complicada es muy importante pues es posible que las máquinas en las que se va a instalar el software tengan diferentes configuraciones que en las que se realizó el mismo produciéndose así un gran problema que se puede evitar.
- **Prueba de Instalación:** asegura que el software puede ser instalado en diferentes condiciones normales y anormales. Además verifica que, una vez instalado, el software opera correctamente. Uno de los aspectos que se debe tener en cuenta a la hora de analizar esta prueba es que el desarrollador debe hacerle entrega al Laboratorio de un Manual de Instalación, desde el mismo momento en que no se cumple con este requisito se hace prácticamente imposible realizar esta prueba; pero si se realiza todo lo que se plantea en el procedimiento de este Trabajo de Diploma la prueba se puede llevar a cabo de forma satisfactoria, verificando así que el software puede ser instalado bajo cualquier tipo de condición.

- **Pruebas de seguridad y control de acceso:** aunque ésta prueba se realiza no se prueba en su totalidad debido a que algunos software utilizan mecanismos de encriptación de datos que por cuestiones de seguridad no deben viajar al servidor en texto claro, como es el caso de las contraseñas, información que se debe guardar encriptada en la base de datos.

### **3.7 Evaluación de las pruebas.**

Durante el proceso de revisión de un producto que se lleva a cabo en el Laboratorio de Calidad de la Facultad 7 no existe un equipo que se encargue de evaluar la calidad del software, lo que trae consigo que el proceso de liberación no tenga la calidad requerida pues resulta de vital importancia que a un software se le apliquen los cuestionarios de evaluación, debido a que los mismo dan una medida mucho más eficiente del grado de calidad con que cuenta un software cuando el mismo está terminado y a punto de ser puesto en funcionamiento.

Con estos cuestionarios el proceso de revisión del producto de software evolucionará a favor del Laboratorio de Calidad que se encarga de esta tarea, ya que tendrán en sus manos las facilidades para evaluar la calidad de un software de una manera sencilla y eficiente. Si a un sistema previamente liberado por dicho laboratorio se le aplicaran los cuestionarios que se proponen en el procedimiento antes explicado se hubiera garantizado un grado de calidad mucho más elevado del mismo, o al menos hubiera sido puesto en marcha con una evaluación previa y medido por una norma de calidad (NC ISO 9126-1: 2005) que avala dichos cuestionarios. Estos cuestionarios son:

- Cuestionario de Evaluación Introductorio.
- Valoración de los resultados de las pruebas y el comportamiento de las características de calidad.
- Valoración de las comprobaciones documentales.
- Valoración de pruebas y comprobaciones.
- Cuestionario de Resumen General.

Al no aplicarse los cuestionarios de evaluación propuestos basados en esta norma se deja de verificar:

### **Funcionalidad:**

- Si el software responde a las tareas y objetivos que requiere el cliente. (Idóneo)
- Si el sistema brinda un alto grado de exactitud en su respuesta. (Preciso)
- Si el sistema es compatible con otros sistemas externos. (Interoperable)
- Si la transmisión de datos a través de este sistema es seguro, así como los privilegios de acceso que le debe brindar a los usuarios. (Seguro)

### **Confiabilidad:**

- La capacidad de recuperación del producto ante cualquier fallo. (Maduro)
- La posibilidad de que el software siga funcionando a pesar de existir un fallo de cualquier tipo. (Tolerante ante fallos)
- Si el software se puede recuperar luego de permanecer durante un periodo de tiempo desactivado. (Recuperable)

### **Usabilidad:**

- Si la documentación referente al software es clara, explicativa, entendible para cualquier tipo de usuario y si el ambiente del mismo es agradable. (Comprensible)
- Si la aplicación del producto aporta nuevos conocimientos al usuario. (Cognoscitivo)
- La posibilidad que brinda el software de ser controlado. (Operable)
- Si la apariencia del producto le es amigable al cliente. (Atractivo)

### **Eficiencia:**

- El rendimiento en cuanto a los tiempos de respuesta dada una función bajo condiciones establecidas. (Eficiente)
- Si el producto es capaz de interactuar con recursos dada una situación. (Uso de Recursos)

### **Mantenibilidad:**

- Si el software es capaz o no de prestarse para ser modificado ante cualquier diagnóstico. (Diagnosticable y Flexible)
- Los efectos inesperados ante algunas modificaciones (Estable)

**Portabilidad:**

- Si el software puede ser instalado bajo cualquier circunstancia. (Instalable)
- Si el producto puede estar instalado en una PC junto a otros sistemas y compartir recursos. (Coexistente)
- Si el producto de software no presenta problemas a la hora de instalar una nueva versión. (Reemplazable)

### **3.8 Conclusiones Parciales**

Como se observa en la comparación realizada la aplicación de este procedimiento bajo la norma ISO/IEC 9126-1:2005, permite una mejor organización estructural para la evaluación del software, una mejor comprensión del Equipo Evaluador sobre el trabajo a desarrollar y el software a evaluar. Además, de desarrollar un grupo de pruebas que permitan evaluar la integridad, los accesos, la funcionalidad y otros aspectos que mediante la evaluación de las inconformidades; la solución de las mismas y los errores detectados van garantizando la confiabilidad y eficiencia con vista a asegurar la calidad del producto.

## Conclusiones

---

A partir de un análisis detallado de cómo se realiza el proceso de pruebas a los productos informáticos revisados en el Laboratorio de Calidad de la Facultad 7, se arribó a las siguientes conclusiones:

- ✓ La principal causa que lo torna ineficiente es la no existencia de un mecanismo, basado en una norma o estándar de calidad que validara el buen funcionamiento del proceso de revisión y evaluación de los productos que se entregaban al laboratorio, garantizando de esta forma una buena estructura organizacional del laboratorio, la definición de los roles necesarios para el cumplimiento satisfactorio de las tareas asignadas, así como la evaluación del proceso basado en criterios definidos por modelos de calidad.
- ✓ Se identificaron y analizaron diferentes normas, con el propósito de seleccionar cual se ajustaba más a las características de funcionamiento del laboratorio y con la cual se obtendrían buenos resultados.
- ✓ La norma escogida para el establecimiento de los principios de funcionamiento del Laboratorio de Calidad fue la Norma Cubana ISO/IEC 9126-1:2005.
- ✓ Se propuso un procedimiento, cuya puesta en práctica favorecería un mejor desempeño en el proceso de pruebas que se lleva a cabo.
- ✓ Se demostró la necesidad de la puesta en práctica del procedimiento propuesto, para lograr una mayor calidad en proceso de revisión y evaluación de los software en el Laboratorio de Calidad de la Facultad 7.

Al finalizar este Trabajo de Diploma se dan por cumplidos los objetivos trazados en sus inicios, obteniéndose la propuesta de procedimiento, en el cual se reflejan los resultados de la investigación llevada a cabo, teniendo como principal aporte la utilización de una norma de calidad que validó el desarrollo del mismo.

## Recomendaciones

---

Al concluir este trabajo se proponen algunas recomendaciones en aras de mejorar el trabajo de revisión y evaluación de software que se lleva a cabo en los Laboratorios de Calidad:

Para la puesta en marcha del procedimiento dentro del Laboratorio de Calidad de la Facultad 7 es necesario:

- ✓ Organizar el laboratorio teniendo en cuenta las responsabilidades propuestas en el procedimiento.
- ✓ Preparar al personal para la utilización de las diferentes pruebas propuestas.
- ✓ Crear las condiciones mínimas necesarias respecto al equipamiento que debe tener el laboratorio.
- ✓ Fomentar el conocimiento de la Norma Cubana ISO/IEC 9126-1: 2005, para su uso efectivo en el trabajo del laboratorio.
- ✓ Proponer la evaluación del procedimiento y su puesta en práctica.
- ✓ Lograr que los desarrolladores del producto entreguen el Diseño de los Casos de Prueba.
- ✓ Mantener un control de versiones debidamente actualizado en el Portal de Calidad de la Facultad 7, actualmente en desarrollo.

## Referencias Bibliográficas

---

1. Menéndez, R.V. [Discurso Inaugural de la Feria Informática 2007] 2007 [cited; Available from: <http://www.mic.gov.cu/hnewdetails.aspx?257>].
2. Carrasco, O.M.F. *Un enfoque actual sobre la calidad del software*. 21-03-2007 [cited; Available from: [http://bvs.sld.cu/revistas/aci/vol3\\_3\\_95/aci05395.htm](http://bvs.sld.cu/revistas/aci/vol3_3_95/aci05395.htm)].
3. Sanchez, R.T., *Calidad concepto y generalidades* in *Area de Planificacion y Calidad*, Consorcio Hospital General Universitario Valencia: España.
4. R.S.Pressman, *Ingeniería del software. Un enfoque práctico*. 3ª ed. 1993.
5. [cited 10-12-2006]; Available from: <http://www.dnv.es/certificacion/sistemasdegestion/calidad/>
6. Navarro, M.R. in *Presentación del Sello a la Calidad UCI y Laboratorio de Certificación*. 2005.
7. Sans, M.C., *Las normas ISO* in *Revista Bibliográfica de Geografía y Ciencias Sociales*, Universidad de Barcelona 1998.
8. Frakes W., T.C., *Software Reuse: Metrics and Models*. Vol. 28. 1996.
9. Alvarez, G., *La enseñanza universitaria de la Auditoría de los Sistemas de Información*. Octubre 1999.
10. Lovelle, J.M.C. in *Calidad del Software*. 21 de octubre 1999. Universidad Nacional de la Pampa.
11. *La Informatización en Cuba*. 2004 [cited 2007 8 de marzo]; Available from: [http://www.cubaminrex.cu/Sociedad\\_Informacion/Cuba\\_SI/Informatizacion.htm](http://www.cubaminrex.cu/Sociedad_Informacion/Cuba_SI/Informatizacion.htm)
12. ISO/IEC TR 15504-1:1998 "Information technology - Software process assessment Part1: Concepts and introductory guide". 1998, Ginebra, Suiza.
13. Wesley, A., "Software Process". 1992.

## Bibliografía

---

1. Barranco-García, M.J. and Granja-Alvarez J.C., «Maintainability as a key factor in maintenance productivity: a case study» in *Proceedings of the International Conference on Software Maintenance*, Monterey, CA, IEEE Computer Society Press, Los Alamitos CA, Noviembre 1996, pp. 87-93.
2. Boehm, B.W., J.R. Brown, and M. Lipov. «Quantitative Evaluation of Software Quality». Pags. 592-605. *Proc. 2nd Intl. Conf., on Software Engineering*. Long Beach, Calif. IEEE Computer Society Oct. 1976.
3. Granja Alvarez J.C., «Contribución al estudio de las técnicas de garantía de calidad», *Novatica*, num. 99, Septiembre/Octubre de 1992, pp. 34- 38.
4. ISO/IEC TR 15504-1:1998 “*Information technology - Software process assessment Part 1: Concepts and introductory guide*”, International Organization for Standards, Ginebra (Suiza).
5. ISO 9001, “*Quality Systems – Model for quality assurance in desing and development*”, International Organization for Standards, Ginebra (Suiza).
6. J. Sanders, E. Curran. *Software Quality*. Addison-Wesley (1994) • M. O. Tingey. *Comparing ISO 9000, Malcom Baldrige and the SEI CMM for software*. Prentice-Hall (1997).
7. M.G. Piattini, J.A. Calvo-Manzano, J. Cerveza, y L. Fernández. *Análisis y diseño detallado de aplicaciones informáticas de gestión. RA-MA (1996)*.
8. Norman Schneidewind, “*Comparing ISO Standard and SEI Model*”, –Naval Postgraduate School, Monterrey (México)
9. Oskarsson Ö, Glass R.L. *An ISO 9000 approach to building uality Software*. Prentice-Hall (1996)
10. Rodriguez M.L., Garvi E., Granja J.C., «Calidad y reusabilidad del software: estudio de la funcionalidad», *Novatica*, num. 125, Enero/Febrero de 1997, pp. 67-70.
11. R. S. Pressman. *Ingeniería del software. Un enfoque práctico*. 3ª Edición. McGrawHill (1993)
12. R. S. Pressman. *Ingeniería del software. Un enfoque práctico*. 4ª Edición. McGrawHill (1998)
13. W.S. Humphrey, “*Characterizing the Software Process: A Maturity Framework*”, IEEE Software, vol. 5 nro. 2, Marzo 1998.

# Anexos

---

**Anexo 1** Norma ISO/IEC 9126-1:2005

**Anexo 2** Formulario de Solicitud de Revisión de Producto de Software.

**Anexo 3** Acuerdo de Confidencialidad

**Anexo 4** Expediente del Producto de Software

**Anexo 5** Acta de Entrega

**Anexo 6** Diseño de los Casos de Pruebas

**Anexo 7** Listas de Chequeo

**Anexo 7.1** Lista de Chequeo de Seguridad

**Anexo 7.2** Lista de Chequeo para la interfaz de Usuario

**Anexo 7.3** Lista de Chequeo para Plataformas interactivas

**Anexo 7.4** Lista de Chequeo para Productos Exportables

**Anexo 8** Registro de No Conformidades

**Anexo 9** Respuesta a las No Conformidades

**Anexo 10** Cuestionarios de Evaluación del Producto de Software

**Anexo 10.1** Cuestionario Evaluación Introductorio

**Anexo 10.2** Cuestionario Evaluación sección A: valoración de los resultados de las pruebas y el comportamiento de las características de calidad.

**Anexo 10.3** Cuestionario Evaluación sección B: Valoración de las comprobaciones documentales.

**Anexo 10.4** Cuestionario Evaluación sección C: Valoración de las Pruebas comprobaciones

**Anexo 10.5** Cuestionario Resumen General.

**Anexo 11** Modelo del Procedimiento.

## **Anexo 1** Norma ISO/IEC 9126-1:2005

Esta norma se puede encontrar en:

- ✓ La Biblioteca de la Universidad de las Ciencias Informáticas en formato digital.
- ✓ La Oficina Nacional de Normalización, en formato físico, cita en Calle E No. 261 Vedado, Ciudad de La Habana. Cuba. Teléfono: 830-0835 Fax: (537) 836-8048  
Correo electrónico: [nc@ncnorma.cu](mailto:nc@ncnorma.cu)
- ✓ En el expediente de la presente Investigación ubicado en:  
[\\10.31.7.200\Areas Tematicas](#)

**Los siguientes anexos** se pueden localizar en el expediente de la investigación localizado en la siguiente dirección: [\\10.31.7.200\Areas Tematicas](#)

**Anexo 2** Formulario de Solicitud de Revisión de Producto de Software.

**Anexo 4** Expediente del Producto de Software

**Anexo 5** Acta de Entrega

**Anexo 6** Diseño de los Casos de Pruebas

**Anexo 7** Listas de Chequeo

**Anexo 7.1** Lista de Chequeo de Seguridad

**Anexo 7.2** Lista de Chequeo para la interfaz de Usuario

**Anexo 7.3** Lista de Chequeo para Manual de Usuario

**Anexo 7.4** Lista de Chequeo para Plataformas interactivas

**Anexo 7.5** Lista de Chequeo para Productos Exportables

**Anexo 8** Registro de No Conformidades

**Anexo 9** Respuesta a las No Conformidades

### **Anexo 3 Acuerdo de confidencialidad**

## **Acuerdo de Confidencialidad del Trabajo en el Laboratorio de Calidad**

Dadas las características y exigencias de un Laboratorio de Pruebas de Software de la Facultad 7, se dispone los siguientes acuerdos que deben ser cumplidos por la Dirección del Laboratorio, Equipo de Trabajo y Cliente (Desarrollador del producto).

1. Deben cumplir con los Procedimientos y ordenes establecidas en el Laboratorio.
2. Toda información que se maneje en el laboratorio debe ser protegida, reservada y tenida a buen resguardo.
3. Cualquier inconformidad que tenga el cliente con el equipo de trabajo debe ser tramitada a través de la Dirección del Cliente.
4. Los errores detectados por el equipo de trabajo no pueden ser ni debatidos ni divulgados por ningún integrante de dicho Equipo.
5. Los informes elaborados en el Laboratorio deben ser guardados en el propio laboratorio, y solo pueden ser consultados por el cliente.
6. Las conclusiones solo debe ser de dominio del cliente y la comisión de evaluación.
7. Mantener la ética profesional por ambas partes.
8. La confianza entre el cliente y el personal del Laboratorio o cualquier lazo afectivo que pueda medir entre ambos, no debe afectar el trabajo ni mucho menos ser motivo de revelación de secretos estatales, problemáticas internas del Laboratorio y la Facultad, ni temas que no sean del interés del cliente o estén fuera del alcance de la tarea o labor que va a desarrollar la Comisión de Trabajo.
9. No utilizar las posibilidades que brindan las tecnologías con fines particulares, ilícitos o de carácter personal.

10. Hacer constancia mediante actas y documentos establecidos de cada visita que se le haga al cliente.
11. Concluido el proceso de evaluación revisar todos los periféricos y documentos que se utilizaron con el objetivo de eliminar cualquier información y rastro del proceso de tal forma que quede listo para el próximo proceso de evaluación.
12. Preservar la integridad moral y revolucionaria, disciplina y respeto es uno de los principios que debe tener presente y cumplir cada integrante del Laboratorio, ante cualquier tarea que se esté desempeñando, tanto en la Facultad como en el Laboratorio.

Firma el presente acuerdo:

\_\_\_\_\_  
Cliente

\_\_\_\_\_  
Integrante de la Comisión

\_\_\_\_\_  
Jefe del Laboratorio

Anexo 10.1 Cuestionario de Evaluación: Valoración Introductoria

**CUESTIONARIO DE EVALUACIÓN DE LA CALIDAD DEL SOFTWARE**

VALORACION INTRODUCTORIA

Proyecto No. \_\_\_\_\_ Siglas: \_\_\_\_\_

Denominación: \_\_\_\_\_

Nivel de Clasificación: \_\_\_\_\_ Software crítico:      no crítico:

Control No.: \_\_\_\_\_ Etapa: \_\_\_\_\_

CARACTERÍSTICAS Y OTROS ATRIBUTOS DE CALIDAD	PUNTUACIÓN			
	3	2	1	0
<b>1. Comprobación de la Descripción del producto</b>				
a) Verificación de las <b>identificaciones e indicaciones</b>				
b) Verificación de las <b>afirmaciones referentes a funcionalidad</b>				
c) Verificación de las <b>afirmaciones referentes a confiabilidad</b>				
d) Verificación de las <b>afirmaciones referentes a usabilidad</b>				
e) Verificación de otras <b>afirmaciones</b>				

# CUESTIONARIO DE EVALUACIÓN DE LA CALIDAD DEL SOFTWARE

## RESUMEN DEL CUESTIONARIO

PUNTUACIÓN	
Identificaciones	
Afirmaciones de funcionalidad	
Afirmaciones de confiabilidad	
Afirmaciones de usabilidad	
Otras afirmaciones	
<b>VALOR</b>	

GRADO DE CALIDAD	
<b>3</b>	<b>Muy bueno</b>
<b>2</b>	<b>Bueno</b>
<b>1</b>	<b>Regular</b>
<b>0</b>	<b>Malo</b>

CRITERIO DE EVALUACIÓN
<input type="checkbox"/> <b>Sin modificaciones</b>
<input type="checkbox"/> <b>Pequeñas modificaciones</b>
<input type="checkbox"/> <b>Grandes modificaciones</b>
<input type="checkbox"/> <b>Nueva elaboración</b>

Conclusión	
	APROBADO
	APROBADO OBSERVACIONES
	NO APROBADO

(área de firma)	
Evaluador:  _____ Cargo:	Fecha:

**Anexo 10.2 Cuestionario de Evaluación: Sección A**

**CUESTIONARIO DE EVALUACIÓN DE LA CALIDAD DEL SOFTWARE**

**SECCIÓN A: VALORACIÓN DE LOS RESULTADOS DE LAS PRUEBAS Y EL COMPORTAMIENTO DE LAS CARACTERÍSTICAS DE CALIDAD**

Proyecto No. \_\_\_\_\_ Siglas: \_\_\_\_\_

Denominación: \_\_\_\_\_

Nivel de Clasificación: \_\_\_\_\_ Software crítico:      no crítico:

Control No.: \_\_\_\_\_ Etapa: \_\_\_\_\_

CARACTERÍSTICAS Y SUBCARACTERÍSTICAS DE CALIDAD	PUNTUACIÓN			
	3	2	1	0
<b>1 Funcionalidad</b>				
<b>1.1 Idoneidad.</b> Capacidad del software para mantener un conjunto apropiado de funciones para las tareas y los objetivos del usuario especificados.				
a) Funcionamiento correcto (ausencia de fallos totales - ciclos infinitos, interrupción de la ejecución o salidas abruptas-, errores críticos, errores de ejecución, resultados incorrectos, correspondencia de las descripciones con los objetos, por ejemplo en el nivel de ayuda solicitado o los mensajes de error según el fallo detectado).				
b) Correspondencia de las funciones con los requisitos funcionales de la Descripción del software (especificaciones).				
<b>1.2 Precisión.</b> Capacidad del software para proporcionar efectos o resultados correctos o convenidos con el grado de				

exactitud necesario.				
a) Exactitud de los cálculos				
<b>1.3 Interoperabilidad.</b> Capacidad del producto de software para interactuar recíprocamente con uno o más sistemas especificados.				
a) Funcionamiento correcto al interactuar (ausencia de fallos totales -ciclos infinitos, interrupción de la ejecución o salidas abruptas-, errores críticos, errores de ejecución, resultados incorrectos).				
<b>1.4 Seguridad de la información (seguridad informática).</b> Capacidad del producto de software para proteger información y los datos, para que personas o sistemas desautorizados no puedan leer o pueden modificar los mismos, y las personas o sistemas autorizados tenga el acceso a ellos.				
a) Auditabilidad (instrumentación de controles de la auditoria informática).				
b) Seguridad (definición de controles de acceso adecuados al software, los subsistemas, las funciones, los ficheros de datos).				
c) Auto chequeo (diseño de la autoprotección contra virus informáticos).				

CARACTERÍSTICAS Y SUBCARACTERÍSTICAS DE CALIDAD	PUNTUACIÓN			
	3	2	1	0
<b>2 Confiabilidad</b>				
<b>2.1 Madurez.</b> Capacidad del producto de software de evitar un fallo total como resultado de haberse producido un fallo del software.				
a) Mantenimiento del funcionamiento correcto ante la aparición de un fallo (ausencia de fallos totales -ciclos infinitos, interrupción de la ejecución o salidas abruptas-, errores críticos, errores de ejecución, resultados incorrectos,)				
b) Uniformidad del retorno al procesamiento ante la aparición de un fallo (restablecimiento de pantallas, menús y ayudas)				
<b>2.2 Tolerancia ante fallos.</b> Capacidad del producto de software de mantener un nivel de desempeño o ejecución				

específico en caso de fallos del software o de infracción de sus interfaces especificadas.				
a) Verificación de la memoria interna y externa (para la instalación del producto, la reconfiguración del producto o para la salva de la información)				
b) Validación previa de condiciones potenciales de errores (particiones de trabajo inexistentes, disponibilidad e integridad de los ficheros, disponibilidad de los periféricos)				
c) Tratamiento de los errores (procedimientos para la detección y corrección de errores internos del software)				
d) Procesamiento degradado (procedimiento para el funcionamiento degradado en caso de los fallos no recuperables como la ausencia de ficheros o deficiencias del hardware).				
<b>2.3 Recuperabilidad.</b> Capacidad del producto de software de restablecer un nivel de ejecución especificado y recuperar los datos directamente afectados en caso de fallo total.				
a) Opciones de recuperabilidad (pérdida o deterioro de datos y elementos componentes del software, errores del operador)				

CARACTERÍSTICAS Y SUBCARACTERÍSTICAS DE CALIDAD	PUNTUACIÓN			
	3	2	1	0
<b>3 Usabilidad</b>				
<b>3.1 Comprensibilidad.</b> Capacidad del producto de software para permitirle al usuario entender si el software es idóneo, y cómo puede usarse para las tareas y condiciones de uso particulares.				
a) Terminología (acorde con la actividad específica de la aplicación o actividad del usuario).				
b) Ayuda en línea (presencia de diferentes niveles de ayudas, sistema, pantalla, campo).				
c) Interfaz de usuario adecuada (representación de los objetos con análogos en el ambiente del usuario, utilización de iconos).				
<b>3.2 Cognoscibilidad.</b> Capacidad del producto del software para permitirle al usuario aprender su aplicación.				

a) Existencia de un DEMO.				
b) Existencia de un Tutorial.				
<b>3.3 Operabilidad.</b> Capacidad del producto del software para permitirle al usuario operarlo y controlarlo.				
a) Utilidad de las ayudas (utilidad de la información que brinda la ayuda).				
b) Operabilidad de las ayudas (facilidad de desplazamiento en la ventana de ayuda).				
c) Carga automática (generación durante la instalación del fichero de comandos para la ejecución del producto de software).				
d) Minimización del trabajo extra del usuario (durante la operación y en caso de fallos).				
e) Facilidades de operación opcional (selección de opciones mediante el cursor de barras, video invertido, brillantez de las opciones, mouse).				
<b>3.4 Atracción.</b> Capacidad del producto del software de ser atractivo o amigable para el usuario.				
a) Uniformidad de la estructura, del contenido y del formato de los elementos componentes.				
b) Uniformidad del vocabulario, de la simbología y de otras convenciones utilizadas.				

CARACTERÍSTICAS Y SUBCARACTERÍSTICAS DE CALIDAD	PUNTUACIÓN			
	3	2	1	0
<b>4 Eficiencia</b>				
<b>4.1 Rendimiento.</b> Capacidad del producto de software para proporcionar un tiempo apropiado de respuesta, procesamiento y de producción de resultados al realizar su función, bajo condiciones establecidas.				
a) Eficiencia en el tiempo de ejecución y de respuesta requeridos.				
<b>4.2 Utilización de recursos.</b> Capacidad del producto de software para utilizar la cantidad y el tipo apropiado de recursos cuando el software realiza su función bajo las condiciones establecidas.				
a) Eficiencia en la utilización de los recursos (memoria interna, utilización de compactadores).				

CARACTERÍSTICAS Y SUBCARACTERÍSTICAS DE CALIDAD	PUNTUACIÓN			
	3	2	1	0
<b>5 Mantenibilidad</b>				
<b>5.1 Diagnosticabilidad.</b> Capacidad del producto del software de ser objeto de un diagnóstico para detectar deficiencias o causas de los fallos totales en el software, o de permitir identificar las partes que van a ser modificadas.				
a) Verificación de la existencia de manuales o secciones de servicio o soporte del software, referencias a herramientas de diagnóstico u otros.				
b) Locación y posible causa del fallo o error (información suficiente para localizar y corregir la causa del error)				
<b>5.2 Flexibilidad.</b> Capacidad del producto del software para permitir llevar a cabo una modificación especificada.				
a) Verificación de la existencia en manuales o secciones de los mismos de los mecanismos documentales de modificación (gestión de configuración)				
<b>5.3 Contrastabilidad.</b> Capacidad del producto del software para permitir la validación de un software modificado				
a) Verificación de la existencia en los planes de pruebas de los casos correspondientes				
<b>5.4 Estabilidad.</b> Capacidad del producto de software para minimizar los efectos inesperados de las modificaciones realizadas al software.				
a) Verificación de la existencia de planes de cambios.				

CARACTERÍSTICAS Y SUBCARACTERÍSTICAS DE CALIDAD	PUNTUACIÓN			
	3	2	1	0
<b>6 Portabilidad</b>				
<b>6.1 Adaptabilidad.</b> Capacidad del producto de software de ser adaptado a los ambientes especificados sin aplicar acciones o medios de otra manera que aquellos suministrados con el propósito de que el software cumpla sus fines				
a) Verificar si el software será adaptado por el usuario final.				
<b>6.2 Instalabilidad.</b> Capacidad del producto de software de ser instalado en un ambiente especificado.				
a) Verificar si el software será instalado por un usuario final,				
<b>6.3 Coexistencia.</b> Capacidad del producto de software de coexistir con otro software independiente en un ambiente común y compartir los recursos comunes.				
a) Verificar si existe otro software instalado que puedan interactuar con él.				
<b>6.4 Reemplazabilidad.</b> Capacidad del producto de software de ser usado en lugar de otro producto de software especificado para los mismos fines y en el mismo ambiente.				
a) Verificar que existe una versión anterior.				

# CUESTIONARIO INDIVIDUAL DE EVALUACIÓN DE LA CALIDAD

## SECCIÓN A

### RESUMEN DEL CUESTIONARIO

PUNTUACIÓN	
Funcionalidad	
Confiabilidad	
Usabilidad	
Eficiencia	
Mantenibilidad	
Portabilidad	
<b>VALOR</b>	

GRADO DE CALIDAD	
3	Muy bueno
2	Bueno
1	Regular
0	Malo

CRITERIO DE EVALUACIÓN
<input type="checkbox"/> Sin modificaciones
<input type="checkbox"/> Pequeñas modificaciones
<input type="checkbox"/> Grandes modificaciones
<input type="checkbox"/> Nueva elaboración

Conclusión	
	APROBADO
	APROBADO OBSERVACIONES
	NO APROBADO

(área de firma)

Evaluador:

Fecha:

Cargo:

Anexo 10.3 Cuestionario de Evaluación: Sección B

**CUESTIONARIO DE EVALUACIÓN DE LA CALIDAD DEL SOFTWARE**

**SECCIÓN B:**

**VALORACIÓN DE LAS COMPROBACIONES DOCUMENTALES**

Proyecto No. \_\_\_\_\_ Siglas: \_\_\_\_\_

Denominación: \_\_\_\_\_

Nivel de Clasificación: \_\_\_\_\_ Software crítico:      no crítico:

Control No.: \_\_\_\_\_ Etapa: \_\_\_\_\_

CARACTERÍSTICAS Y OTROS ATRIBUTOS DE CALIDAD	PUNTUACIÓN			
	3	2	1	0
<b>1. Integridad del producto a evaluar</b>				
a) Existencia del producto de software en su totalidad, lo que incluye: - El instalador (si procede) - El programa ejecutable (todos los programas y ficheros del software) - La documentación del producto de software -- descripción del software, documentación de usuarios (manuales y otros) - Planes, especificaciones, casos y datos de pruebas				
b) Existencia de avales, constancias y certificaciones obtenidas				
<b>2. Corrección</b>				
a) Verificación de la ausencia de errores y ambigüedades en la documentación				

<b>3. Consistencia</b> Se comprueba si es consistente la documentación con cada características de calidad, en particular en cuanto a:				
<b>3.1 FUNCIONALIDAD</b>				
<b>3.1.1 Idoneidad.</b> Capacidad del software para mantener un conjunto apropiado de funciones para las tareas y los objetivos del usuario especificados.				
a) Verificación de la correspondencia de las funciones con los requisitos funcionales de la Descripción del software (especificaciones)				
<b>3.1.2 Precisión.</b> Capacidad del software para proporcionar efectos o resultados correctos o convenidos con el grado de exactitud necesario.				
a) Revisión de las descripciones relacionadas con la exactitud de los cálculos.				
<b>3.1.3 Seguridad de la información (seguridad informática).</b> Capacidad del producto de software para proteger información y los datos, para que personas o sistemas desautorizados no puedan leer o pueden modificar los mismos, y las personas o sistemas autorizados tenga el acceso a ellos.				
a) Auditabilidad (instrumentación de controles de la auditoría informática)				

CARACTERÍSTICAS Y OTROS ATRIBUTOS DE CALIDAD	PUNTUACIÓN			
	3	2	1	0
b) Seguridad (definición de controles de acceso adecuados al software, los subsistemas, las funciones, los ficheros de datos)				
c) Autochequeo (diseño de la autoprotección contra virus informáticos)				
<b>3.2 CONFIABILIDAD</b>				
<b>3.2.1 Tolerancia ante fallos.</b> Capacidad del producto de software de mantener un nivel de desempeño o ejecución específico en caso de fallos del software o de infracción de sus interfaces especificadas.				
a) Comprobación de la existencia de cálculos o descripciones sobre el particular.				

<b>3.2.2 Elementos de disponibilidad</b>					
a) Comprobación de la existencia de cálculos o descripciones sobre el particular.					
<b>3.3 USABILIDAD</b>					
<b>3.3.1 Comprensibilidad.</b> Capacidad del producto de software para permitirle al usuario entender si el software es idóneo, y cómo puede usarse para las tareas y condiciones de uso particulares.					
a) Terminología empleada en la documentación (acorde con la actividad específica de la aplicación o actividad del usuario).					
b) Ayuda en línea (referencia a la presencia de diferentes niveles de ayudas, sistema, pantalla, campo).					
<b>3.3.2 Cognoscibilidad o facilidad de entrenamiento.</b> Capacidad del producto de software para permitirle al usuario aprender su aplicación.					
a) Existencia de las referencias a un DEMO.					
b) Existencia de las referencias a un Tutorial.					
<b>3.3.3 Operabilidad.</b> Capacidad del producto del software para permitirle al usuario operarlo y controlarlo.					
a) Utilidad del manual correspondiente (utilidad de la información que brinda).					
b) Explicación de las ayudas.					
<b>3.3.4 Atracción.</b> Capacidad del producto del software de ser atractivo o amigable para el usuario.					
a) Utilización correcta del idioma.					
b) Uniformidad del vocabulario, de la simbología y de otras convenciones utilizadas.					

CARACTERÍSTICAS Y OTROS ATRIBUTOS DE CALIDAD	PUNTUACIÓN			
	3	2	1	0
<b>3.4 MANTENIBILIDAD</b>				
<b>3.4.1 Diagnósticabilidad.</b> Capacidad del producto del software de ser				

objeto de un diagnóstico para detectar deficiencias o causas de los fallos totales en el software, o de permitir identificar las partes que van a ser modificadas.				
a) Verificación de la existencia de manuales o secciones de servicio o soporte del software, referencias a herramientas de diagnóstico u otros.				
b) Locación y posible causa del fallo o error (información suficiente para localizar y corregir la causa del error)				
<b>3.4.2 Flexibilidad.</b> Capacidad del producto del software para permitir llevar a cabo una modificación especificada.				
a) Verificación de la existencia en manuales o secciones de los mismos de los mecanismos documentales de modificación (gestión de configuración)				
<b>3.4.3 Contrastabilidad.</b> Capacidad del producto del software para permitir la validación de un software modificado				
a) Verificación de la existencia en los planes de pruebas de los casos correspondientes				
<b>4. Conformidad</b>				
a) Capacidad del producto de software para adherirse a las normas que se apliquen, convenciones, regulaciones, leyes y las prescripciones similares				
b) Procedencia de los avales y certificaciones presentadas				
c) Normalización del vocabulario y la simbología				
d) Normalización del aseguramiento de la calidad				
<b>5. Comprensión y claridad</b>				
a) Selección de términos, representaciones gráficas, explicaciones detalladas referencias a fuentes de información útiles.				

# CUESTIONARIO INDIVIDUAL DE EVALUACIÓN DE LA CALIDAD

## SECCIÓN B

### RESUMEN DEL CUESTIONARIO

PUNTUACIÓN	
Integridad	
Corrección	
Consistencia	
Conformidad	
Comprensión claridad	
<b>VALOR</b>	

GRADO DE CALIDAD	
3	Muy bueno
2	Bueno
1	Regular
0	Malo

CRITERIO DE EVALUACIÓN
<input type="checkbox"/> Sin modificaciones
<input type="checkbox"/> Pequeñas modificaciones
<input type="checkbox"/> Grandes modificaciones
<input type="checkbox"/> Nueva elaboración

Conclusión	
	APROBADO
	APROBADO OBSERVACIONES
	NO APROBADO

(área de firma)	
Evaluador: _____	Fecha:
Cargo:	

**Anexo 10.4 Cuestionario de Evaluación: Sección C**

**CUESTIONARIO DE EVALUACIÓN DE LA CALIDAD DEL SOFTWARE**

**SECCIÓN C:  
VALORACIÓN DE PRUEBAS Y COMPROBACIONES:  
CALIDAD EN EL USO**

Proyecto No. \_\_\_\_\_ Siglas: \_\_\_\_\_  
 Denominación: \_\_\_\_\_  
 Nivel de Clasificación: \_\_\_\_\_ Software crítico:      no crítico:  
 Control No.: \_\_\_\_\_ Etapa: \_\_\_\_\_

CARACTERÍSTICAS Y SUBCARACTERÍSTICAS DE CALIDAD	PUNTUACIÓN			
	3	2	1	0
<b>1. Eficacia.</b> Capacidad del producto de software de permitir que los usuarios logren objetivos especificados con precisión e integridad en un contexto especificado.				
a) Existencia del aval correspondiente indicando la eficacia alcanzada.				
<b>2. Productividad.</b> Capacidad del producto de software de permitir que los usuarios dediquen una cantidad de recursos apropiada a la efectividad alcanzada en un contexto de uso especificado.				
a) Existencia del aval correspondiente.				
<b>3. Seguridad.</b> Capacidad del producto de software de alcanzar niveles aceptables de riesgo de daños a las personas, el negocio, el software, la propiedad o el ambiente en un contexto de uso especificado (para software crítico).				
a) Existencia y procedencia del Análisis de riesgo.				
<b>4. Satisfacción.</b> Capacidad del producto de software de satisfacer a los usuarios en un contexto de uso especificado.				
a) Existencia del aval correspondiente, indicando su satisfacción por la precisión e integridad alcanzada.				

# CUESTIONARIO INDIVIDUAL DE EVALUACIÓN DE LA CALIDAD

## SECCIÓN C

### RESUMEN DEL CUESTIONARIO

PUNTUACIÓN	
Eficacia	
Productividad	
Seguridad*	
Satisfacción	
<b>VALOR</b>	

GRADO DE CALIDAD	
3	Muy bueno
2	Bueno
1	Regular
0	Malo

\*Nota: El valor de la seguridad posee tal peso, que decide el resultado de la Evaluación para los software críticos, por lo que un resultado de 0 (malo) o 1 (regular) traslada ese mismo valor al grado de calidad.

CRITERIO DE EVALUACIÓN
<input type="checkbox"/> Sin modificaciones
<input type="checkbox"/> Pequeñas modificaciones
<input type="checkbox"/> Grandes modificaciones
<input type="checkbox"/> Nueva elaboración

Conclusión	
	APROBADO
	APROBADO OBSERVACIONES
	NO APROBADO

(área de firma)

Evaluador:

\_\_\_\_\_

Cargo:

Fecha:

Anexo 10.5 Cuestionario de Evaluación: Resumen General

**CUESTIONARIO DE EVALUACIÓN DE LA CALIDAD DEL SOFTWARE**

**RESUMEN GENERAL**

<b>GRADOS DE CALIDAD POR SECCIONES</b>		
<b>Sección A</b>		<b>( )Muy bueno ( )Bueno ( )Regular ( )Malo</b>
<b>Sección B</b>		<b>( )Muy bueno ( )Bueno ( )Regular ( )Malo</b>
<b>Sección C</b>		<b>( )Muy bueno ( )Bueno ( )Regular ( )Malo</b>
<b>Valoración Introdutoria</b>		<b>( )Muy bueno ( )Bueno ( )Regular ( )Malo</b>

<b>CRITERIO DE EVALUACIÓN</b>	
<b>Valoración Introdutoria</b>	<b>( ) Sin modificaciones ( )Pequeñas modificaciones ( )Grandes modificaciones ( )Nueva elaboración</b>
<b>Sección A</b>	<b>( ) Sin modificaciones ( )Pequeñas modificaciones ( )Grandes modificaciones ( )Nueva elaboración</b>
<b>Sección B</b>	<b>( ) Sin modificaciones ( )Pequeñas modificaciones ( )Grandes modificaciones</b>

	<input type="checkbox"/> <b>Nueva elaboración</b>
<b>Sección C</b>	<input type="checkbox"/> <b>Sin modificaciones</b> <input type="checkbox"/> <b>Pequeñas modificaciones</b> <input type="checkbox"/> <b>Grandes modificaciones</b> <input type="checkbox"/> <b>Nueva elaboración</b>

<b>CONCLUSIÓN</b>	
	<b>APROBADO</b>
	<b>APROBADO CON OBSERVACIONES</b>
	<b>NO APROBADO</b>

<p>(área de firma)</p> <p>Evaluador responsable: <span style="float: right;">Fecha:</span></p> <p>_____</p> <p>Cargo:</p> <p>Evaluador:</p> <p>Cargo:</p> <p>Evaluador:</p> <p>Cargo:</p>	
---	--

**Anexo 11 Modelo de Procedimiento según Norma ISO 9001**

	<b>PROCEDIMIENTO PARA LA REVISION DE PRODUCTOS SOFTWARE</b>
---	---

Elaborado por: Yelenys Delvoys Marchante Mireilys Martinez Miranda	Revisado por: Aprobado por:	Pág.
--	--------------------------------	------

## Glosario de términos y acrónimos

---

**Aseguramiento de la calidad:** Conjunto de acciones planificadas y sistemáticas, implementadas en el Sistema de Calidad, que son necesarias para proporcionar la confianza adecuada de que un producto satisfará los requisitos dados sobre la calidad.

**Certificación el software:** Es el proceso mediante el cual una empresa productora de software emite un certificado de garantía de la calidad, que proporciona confianza al consumidor del producto y ratifica que el software ha sido producido por dicha compañía.

**Control de cambios:** es un conjunto de procedimientos y herramientas que se usan para gestionar el uso de los objetos.

**CMM (Capability Maturity Model for Software):** es un modelo de procesos para el desarrollo y mantenimiento de sistemas de software, diseñado sobre los criterios:

- La calidad de un producto o sistema es consecuencia directa de los procesos empleados en su desarrollo.
- Las organizaciones que desarrollan software presentan un atributo denominado madurez, cuya medida es proporcional a los niveles de capacidad e institucionalización de los procesos que emplean en su trabajo.

**Código fuente:** también denominado fuente o texto fuente. Es el texto que contiene las instrucciones del programa, escritas en el lenguaje de programación. Se trata de un archivo de texto legible que se puede copiar, modificar e imprimir sin dificultad.

**Costo de calidad:** Métrica para cuantificar el tamaño de la calidad. Forma parte integral del costo de producción, estando presente en los resultados que se reflejan en el Estado de Resultado de una organización, pero no se cuantifican por separado, lo que impide su

adecuado control y análisis, dificultando la aplicación de posibles medidas correctivas y el proceso de toma de decisiones.

**Estándar:** es una especificación que regula la realización de ciertos procesos o la fabricación de componentes para garantizar la interoperabilidad.

**Evaluación:** es medio fundamental para conocer la relevancia social de los objetivos planteados, el grado de avance con respecto a los mismos, así como la eficacia, impacto y eficiencia de las acciones realizadas.

**INFOMED:** es el Portal de Salud Cubano y la red de personas e instituciones que comparten el propósito de facilitar el acceso a la información de salud en Cuba.

**ISO (International Organization for Standardization):** es una organización internacional no gubernamental, compuesta por representantes de los organismos de normalización (ONs) nacionales, que produce normas internacionales industriales y comerciales. Dichas normas se conocen como **normas ISO** y su finalidad es la coordinación de las normas nacionales, en consonancia con el Acta Final de la Organización Mundial del Comercio, con el propósito de facilitar el comercio, facilitar el intercambio de información y contribuir con unos estándares comunes para el desarrollo y transferencia de tecnologías.

**Listas de Chequeo:** son para chequear características generales del producto que no se verifican a través de las pruebas.

**Método:**

- Modo de obrar o proceder.
- Modo ordenado de actuar.
- Modo estructurado y ordenado de obtener un resultado, descubrir la verdad y sistematizar los conocimientos.
- Obra o compendio de reglas y ejercicios prácticos.

**Metodología:** es la parte del proceso de investigación que sigue a la propedéutica y permite sistematizar los métodos y las técnicas necesarios para llevarla a cabo. Es pues, una etapa, una parte del proceso.

**Métrica:** es una guía formal, aunque flexible en su utilización, para la Planificación, Análisis, Diseño y Construcción e Implantación de Sistemas de Información empleando conceptos y técnicas de Ingeniería de Sistemas de Información y Tecnología de la Información.

**Medición del Software:** es un proceso que se realiza para entender, monitorizar, controlar, predecir y probar el correcto desarrollo y funcionamiento del software y persigue tres objetivos fundamentales:

- Entender qué ocurre durante el desarrollo y el mantenimiento del producto.
- Controlar qué es lo que ocurre en los proyectos.
- Mejorar los procesos y los productos.

**Modelo de Calidad:** El modelo de calidad consiste en reunir todas las actividades y funciones en forma tal que ninguna de ellas esté subordinada a las otras y que cada una se planee, controle y ejecute de un modo formal y sistemático.

**Norma de Calidad:** es un documento, establecido por consenso y probado por un organismo reconocido (nacional o internacional), que proporciona, para un uso común y repetido, reglas, directrices o características para las actividades de calidad o sus resultados, con el fin de conseguir un grado óptimo de orden en el contexto de la calidad. Los principales organizaciones internacionales, emisoras de normas de calidad son: ISO (Organización Internacional de Estándares) y IEC (Comisión Electrotécnica Internacional)

**Plan de Calidad:** consiste en un conjunto de principios, métodos y recursos organizados estratégicamente para movilizar a toda la empresa, con el fin de satisfacer las necesidades del cliente al mínimo coste.

**Procedimiento:** Sucesión cronológica de operaciones concatenadas entre sí, que se constituyen en una unidad de función para la realización de una actividad o tarea específica dentro de un ámbito predeterminado de aplicación. Todo procedimiento involucra actividades y tareas del personal, determinación de tiempos de métodos de trabajo y de control para lograr el cabal, oportuno y eficiente desarrollo de las operaciones.

**Proceso:** Es un conjunto de actividades que suceden de forma ordenada a partir de la combinación de materiales, maquinaria, gente, métodos, y medio ambiente, para convertir insumos en productos con valor agregado.

**Proyecto productivo:** todo proceso (o acción estratégica), encaminado a conseguir un objetivo previamente fijado, con unos recursos económicos y temporales limitados en los que la finalidad es desarrollar una actividad de tipo económico fundamentalmente caracterizada por la creación de productos (bienes y/o servicios).

**Rol:** Papel, cometido o función que tiene o desempeña que interpreta un actor.

**Sello de Calidad:** Es una herramienta efectiva para diferenciar a una empresa de sus competidores, ya que garantiza que sus servicios cumplen con los requisitos especificados.

**Sistema de calidad:** Conjunto de la estructura, responsabilidades, actividades, recursos y procedimientos de la organización de una empresa, que ésta establece para llevar a cabo la gestión de su calidad.

**Switch:** es un dispositivo electrónico de interconexión de redes de computadoras que opera en la capa 2 (nivel de enlace de datos) del modelo OSI (Open Systems Interconnection).