

# **Universidad de las Ciencias Informáticas**

## **Facultad 2**



***Diseño de un modelo de prueba de la confiabilidad, según los tipos de productos que se desarrollan en DATEC.***

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

**Autor:**

Yenisleidis Gómez Galbán

**Tutor(es):**

MSc. Lic. Yadira Ruíz Constanten

**Junio de 2010**

***No se me ha olvidado el lugar de donde vengo,  
Y puede que no exista el sitio donde voy...***

## ***Declaración de Autoría***

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

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

---

Yenisleidis Gómez Galbán  
Firma del Autor

---

Yadira Ruíz Constanten  
Firma del Tutor

*A los 3 ángeles que más quiero en la vida...*

*Mi madre, mi padre y a mi Yuni, por creer en mí como lo hicieron siempre...*

*Lejos en labor de esta ilusión, pero muy cerca en mi corazón...*

*A mi novio...*

*A mi familia...*

*A todos mis amigos...*

*A mi tierra querida...*

*A todos los que han estado ahí para mí.*

## **Agradecimientos**

*A mi madre y mi padre que le he echado mucho de menos en estos tiempos, por el cariño tan especial que me han regalado toda la vida, por ese amor tan infinito que me hace crecer mucho mucho y del cual estoy muy orgullosa, le doy gracias a dios por tenerlos a uds conmigo... los amo.*

*A mi hermana Yuni que tanto amo y por la nunca desistí de llegar hasta el final de este laberinto interminable, eres la luz de mi vida.*

*A mi novio, que sin el creo si soy bien sincera, creo no merecer este título de ingeniera sino está presente hoy conmigo, te quiero mucho...*

*A Yaya, por esa “magia” que va regalando siempre, por su paciencia y por el trato tan exclusivo que me dio desde el primer día. Más que una tutora... una amiga.*

*A Yoa y Alberto, por cuidar y hacer feliz a mí y a los míos...*

*A mis suegros, les estaré siempre agradecida por acogerme como su niña y quererme como yo a ellos.*

*A mi familia de la Habana, Saylín y Nelson, gracias por tener a esas dos niñas más también, que siempre me llenan de alegría, Amandita y Noslencita...*

*A Naivis, Pablito y Enaide hermanos de corazón, lejos un poquito, pero siempre con ustedes.*

*Ana Rosa, Lianet, les debo parte de este título y saben que sin ustedes creo no haberlo logrado, les estaré infinitamente agradecida, estaremos unidas siempre, lo sé...*

*A mis abues.*

*A todos esos Gómez que no por muchos dejan de ser unidos y especiales.*

*A los Galbán por acogerme “siempre” así, con los brazos abiertos.*

*A mi madrina, por su amor, mal llamado de “segunda madre”.*

*A todos mis amigos que están y a los que no...*

*A mi Santiago de Cuba, que no por lejos deja de vivir cada momento dentro de mí.*

Propiciar la calidad en el software es una actividad que ha surgido como consecuencia de la fuerte demanda de Sistemas de Software en todos los procesos que se desarrollan en la actualidad; desde programas elementales de contabilidad hasta programas complejos. De allí el esfuerzo que se ha desplegado para obtener software de alta calidad. El aseguramiento de la calidad toma en cuenta todas aquellas acciones planificadas y sistemáticas necesarias para proporcionar la confianza de que un producto o servicio satisfaga los requisitos de calidad establecidos. Para que sea efectivo, se requiere una evaluación permanente de aquellos factores que influyen en la adecuación del diseño y de las especificaciones según las aplicaciones previstas.

El uso eficiente de todos los factores y técnicas que intervienen en la evaluación de la calidad del software son un objetivo indispensable para el éxito de un producto de software. En la Universidad de las Ciencias Informáticas existen varios centros productivos creados con el objetivo de organizar y potenciar la producción de software. Uno de ellos es el Centro de Tecnologías de Gestión de Datos (DATEC), el cual se enfoca esencialmente en el estudio y análisis de tecnologías de bases de datos. La presente investigación maneja el aseguramiento de la confiabilidad de los tipos de productos de DATEC, con la intención de crear un Modelo de Pruebas de Confiabilidad.

Una vez realizado el estudio actual en la institución y al detectar la problemática del centro, se plantea, en aras de enmendar la situación vigente, proponer un modelo para garantizar la confiabilidad del producto, a través de las dimensiones que responden a este atributo de calidad de software. Analizando para ello, los conceptos, los estándares y algunos modelos de calidad que tributan información a tener en cuenta en la propuesta.

**Palabras Claves:** calidad, confiabilidad, pruebas de software, modelo de pruebas de confiabilidad.

**Índice**

Introducción .....	12
1. Capítulo 1: Fundamentación Teórica .....	17
1.1. Proyectos de Software.....	17
1.1.1. Gestión de Proyectos.....	17
1.1.2. PMBOK .....	18
1.2. Gestión de Calidad del Proyecto.....	21
1.2.1. Aseguramiento de la Calidad .....	22
1.2.2. Organización Internacional para la Estandarización (ISO) .....	22
1.2.3. ISO 9126 .....	24
1.3. Confiabilidad.....	25
1.3.1. Análisis de la confiabilidad.....	26
1.4. Pruebas de Software .....	28
1.4.1. Plan de pruebas.....	30
1.4.2. Tipos de Pruebas.....	30
1.5. Pruebas de Confiabilidad.....	31
1.6. Riesgos.....	32
1.6.1. Tipos de Riesgos .....	33
1.6.2. Identificación de Riesgos .....	33
2. Capítulo 2: Propuesta de un Modelo de Pruebas de Confiabilidad.....	34
2.1. Modelos .....	34
2.2. Resumen de los Principales Modelos de Calidad .....	35
2.2.1. Modelo McCall .....	36
2.2.2. Modelo Boehm .....	37
2.2.3. Modelo Hewlett-Packard.....	38
2.2.4. Modelo Dromey .....	39
2.2.5. Norma ISO 9126.....	40
2.2.6. Dimensiones de Confiabilidad ISO 9126.....	41
2.2.7. Análisis de los modelos .....	42
2.3. DATEC .....	43

2.3.1. Tipos de productos de DATEC .....	44
2.4. Análisis de los riesgos asociados a la confiabilidad en DATEC .....	46
2.5. Función de las pruebas en el Modelo V .....	49
2.6. Solución Propuesta.....	50
2.6.1. Objetivos del modelo .....	50
2.7. Pruebas de Confiabilidad.....	52
2.7.1. Pruebas de Componentes .....	53
2.7.2. Pruebas de Integración.....	54
2.7.3. Pruebas de Recuperabilidad.....	55
2.7.4. Pruebas Estructurales.....	55
2.7.5. Pruebas de Sistema .....	56
Capitulo 3: Validación de la Solución .....	59
3. Validación del modelo de confiabilidad .....	59
3.1. Delphi como método de validación de experto.....	59
3.2. El Proceso de Selección de Expertos .....	59
3.3. Elaboración del Cuestionario .....	64
3.1. Validación de la Propuesta .....	64
3.1.1. Importancia de la propuesta.....	65
3.1.2. Satisfacción de las necesidades del centro.....	65
3.1.3. Necesidad de empleo de la propuesta.....	66
3.1.4. Adaptabilidad a proyectos productivos.....	66
3.1.5. Eficacia de la propuesta.....	67
3.1.6. Mejora de la confiabilidad de los productos .....	68
3.1.7. Recomendaciones de los Expertos.....	68
3.2. Resultados Finales de la Validación por Experto .....	69
3.3. Validación de la propuesta a través de un producto.....	70
Conclusiones .....	73
Recomendaciones .....	74
Bibliografía.....	75
Referencias Bibliográficas.....	76
Glosario de Términos.....	78

Anexos..... 79

Fig: 1 Grupos de procesos de la gestión de proyectos.....	19
Fig: 2 9 Áreas del Conocimiento. ....	20
Fig: 3 Procesos de la gestión de calidad de proyectos.....	21
Fig: 4 Atributos de calidad externa e interna. ....	24
Fig: 5 Tipos de pruebas de confiabilidad.....	32
Fig: 6 Modelo McCall .....	37
Fig: 7 Modelo Boehm.....	38
Fig: 8 Atributos de calidad Modelo FURPS .....	39
Fig: 9 Atributos de calidad Modelo Dromey.....	40
Fig: 10 Modelo de ISO 9126 .....	41
Fig: 11 Modelo V.....	49
Fig: 12 Importancia de la propuesta.....	65
Fig: 13 Satisfacción de las necesidades del centro.....	66
Fig: 14 Necesidad de empleo de la propuesta .....	66
Fig: 15 Necesidad de empleo de la propuesta .....	67
Fig: 16 Eficacia de la propuesta .....	67
Fig: 17 Resultados finales de la validación.....	69

Tabla 1: Nivel de cumplimiento-----	52
Tabla 2: Nivel de Confiabilidad-----	53
Tabla 3: Listado de expertos-----	60
Tabla 4: Grado de conocimiento en el tema-----	61
Tabla 5: Coeficiente de conocimiento-----	61
Tabla 6: Grado de influencia de los expertos-----	62
Tabla 7: Tabla Patrón-----	62
Tabla 8: Coeficiente de argumentación-----	63
Tabla 9: Coeficiente de competencia-----	63
Tabla 10: Objetivos de la propuesta por preguntas-----	64
Tabla 11: Asignación del grado de importancia y las no conformidades-----	71

### Introducción

La industria del software cuenta sólo con algunas décadas y está aún proyectando sus caminos, buscando la manera adecuada de desarrollarse. Hace algún tiempo las pruebas de software eran sólo una actividad que realizaba el programador para encontrar fallas en sus productos. Pero con el paso del tiempo se han convertido en un proceso indispensable, cuyo propósito fundamental es evaluar la funcionalidad del software, con un conjunto de herramientas, técnicas y métodos que hacen más completo el desempeño de un software, que aligeran el costo del tiempo del desarrollo del producto e intensifican la calidad del mismo.

Un buen proceso de pruebas necesita establecer una *Estrategia de Pruebas* bien diseñada y un mecanismo que permita evaluar la calidad durante el desarrollo de este proceso. De esta manera se puede tener una idea de cómo se llevó a cabo el proceso de pruebas y detectar cuáles fueron las principales deficiencias que afectaron la calidad del software para proceder a su mitigación tomando las medidas necesarias y acciones correctivas para contrarrestarlos.

Actualmente el concepto calidad introduce el término de mejora continua en cualquier organización y a todos los niveles de la misma. La necesidad de construir un producto de software con buena calidad y que además satisfaga al cliente es un reto que tienen hoy en día las grandes empresas productoras de software. A nivel mundial se han ido buscando alternativas y ha tomado auge el interés por la planificación y el control de la calidad del software. Además, se han diseñado metodologías y herramientas de apoyo para la aplicación de diferentes técnicas de aseguramiento de la calidad, así como sistemas que persigan este objetivo, basándose en las normas y estándares establecidos por la Organización Internacional para la Estandarización (ISO), ofreciendo numerosas ventajas entre las que se cuentan:

Monitorizar los principales procesos.

- ✓ Asegurar su efectividad.
- ✓ Mejora continua de procesos.
- ✓ Mantener registros de gestión, procesos y procedimientos.
- ✓ Mejorar la satisfacción de los clientes o usuarios.
- ✓ Reducir las incidencias de producción o prestación de servicios.

Queda claro que un software con calidad es uno de los principios fundamentales que transmite confianza

en el producto. Pero ¿cómo saber que el producto cuenta con la calidad que se requiere?

Un punto importante dentro de la gestión de calidad de proyecto, es el aseguramiento de la calidad, el cual verifica que el proyecto emplee todas las herramientas necesarias en el cumplimiento de los requisitos definidos. Un producto de software con calidad, bien definido arroja con seguridad al producto, confiabilidad, la cual se define como la probabilidad en que un producto realizará su función prevista sin incidentes por un período de tiempo especificado y bajo condiciones indicadas.

Cuba, inmersa en la informatización de los procesos de la sociedad, no se encuentra ajena a estos temas. La Industria Cubana del Software tiene la ardua tarea de lograr que los productos desarrollados en el país cumplan con las normas y estándares internacionales de calidad. La Universidad de las Ciencias Informáticas (UCI) tiene como una de sus principales misiones producir software y servicios informáticos.

Actualmente en la UCI el aseguramiento de calidad de las producciones de software es una de las prioridades de la universidad. Para ello se hace necesario contar con lineamientos, procedimientos y estándares que normen las producciones. La Dirección de Calidad de Software (CALISOFT) de la UCI es una de las principales referencias de calidad y órgano de certificación de software, conocido y acreditado a nivel nacional.

Dentro de la universidad se crearon varios centros productivos con el objetivo de organizar y potenciar la producción. Uno de ellos es el Centro de Tecnologías de Gestión de Datos (DATEC), enmarcado esencialmente en el estudio y análisis de las tecnologías de bases de datos. DATEC mantiene estrecha colaboración con importantes empresas, centros de investigación y desarrollo de alto nivel, así como con comunidades internacionales de desarrollo. Uno de sus principales objetivos es proveer soluciones integrales y consultorías relacionadas con tecnologías de bases de datos y análisis de información, además de desarrollar nuevas tecnologías de bases de datos, de procesamiento y representación de la información. Es un centro que trabaja por líneas temáticas y los grupos de trabajo son internos, y su funcionamiento es sustentado en líneas de productos de software.

La presente investigación maneja el aseguramiento de la confiabilidad de los tipos de productos de DATEC. Al realizar diferentes análisis en el centro, se detectaron riesgos que afectan la confiabilidad de los productos que se desarrollan. El no contar con métodos que asuman la capacidad de prevenir defectos por errores internos o externos en el software pudiera ocasionar pérdidas de recursos y de tiempo; cuando algún componente del sistema falla, puede afectarse el correcto funcionamiento del producto. No se especifican con métodos de verificación y control para la utilización de estándares de codificación y

documentación para controlar las fallas. El sistema automáticamente no tiene forma de recuperarse y reanudar la actividad que estaba realizando y si el servidor se detiene prematuramente, pudiera perderse alguna información. Actualmente los datos con que prueba el software son datos generados por ellos, pues no existe precisión en los cálculos y emisiones de resultados.

Teniendo en cuenta la situación anteriormente planteada se define el siguiente **problema científico**: ¿Cómo contribuir a la detección de defectos en los productos desarrollados en DATEC y evaluar su impacto?

Para dar solución al problema planteado se propone como **objeto de estudio**: aseguramiento de la confiabilidad de productos de software, el cual se enmarca en el siguiente **campo de acción**: pruebas de confiabilidad a productos de software en DATEC.

La **idea a defender** que se establece en la presente investigación es la siguiente:

Con la definición de un modelo de prueba que permita evaluar la confiabilidad de los productos, según el tipo al que pertenezcan, DATEC contará con una herramienta que le facilite determinar la calidad de los mismos.

Se plantea como **objetivo general** de la investigación: Diseñar un modelo de prueba que permita evaluar la confiabilidad de los productos desarrollados en DATEC.

Objetivos específicos.

- ✓ Caracterizar la situación actual en la región y en el país sobre estos temas y definir la posición como investigador.
- ✓ Caracterizar la situación existente en torno a los modelos de prueba, haciendo énfasis en las pruebas de confiabilidad de los productos desarrollados en DATEC.
- ✓ Evaluar la efectividad del modelo diseñado a través de métodos estadísticos.
- ✓ Aplicar el modelo diseñado en los productos que se desarrollan en el DATEC.

Para dar cumplimiento al objetivo planteado se han trazado las siguientes **Tareas de la Investigación**:

- ✓ Revisión y selección de la bibliográfica para identificar las insuficiencias del tema y las tendencias actuales existentes sobre las pruebas de confiabilidad realizada a los productos.
- ✓ Análisis de los modelos de calidad tales como ISO/IEC 9126-1:2001, NC ISO/IEC 9126-1:2005, Modelos [Mccall, Boehm, FURPS, Dromey].

- ✓ Análisis de los métodos y procedimientos que se emplean para realizar pruebas de confiabilidad de un producto.
- ✓ Análisis de los riesgos potenciales asociados a la confiabilidad.
- ✓ Análisis de los tipos y modelos de pruebas.
- ✓ Evaluación de la información obtenida a partir de los métodos teóricos (Analítico – Sintético), métodos empíricos (Entrevista Observación y Revisión Documental), se define la posición como investigador.
- ✓ Identificación los modelos de calidad relacionado con el tema (pruebas de confiabilidad en productos) y caracterización de su marco de actuación apoyándose en los métodos empíricos (Observación, Entrevista, Análisis de documentos, Criterios de especialistas).
- ✓ Adaptación de algunos métodos de prueba encontrados en función del entorno en el que se realiza la investigación.
- ✓ Elaboración del modelo de prueba de confiabilidad de los productos
- ✓ Descripción del modelo diseñado acorde a las condiciones del entorno analizado.
- ✓ Evaluar la efectividad del modelo diseñado utilizando métodos de validación por expertos y métodos de evaluación del producto.

### Posibles resultados:

- ✓ Obtención un modelo de pruebas de confiabilidad de los productos desarrollados en DATEC.

Para realizar las tareas de la investigación se emplearán los siguientes **métodos científicos**:

### Métodos teóricos:

- ✓ **Método analítico – sintético:** Se utilizará para captar y resumir varios documentos y procedimientos legales por los cuales se rige las normas ISO a la hora de elaborar el Fundamento Teórico. De ellos se extraerán las ideas fundamentales y al mismo tiempo se detallará la información necesaria para el diseño correcto del modelo de pruebas.

### Métodos empíricos:

- ✓ **Entrevista:** Se utilizará la entrevista como una conversación planificada con los usuarios de DATEC, para obtener información acerca del problema en cuestión. Su uso constituye un medio para el conocimiento cualitativo de las características particulares de este proyecto y puede influir en el posterior análisis y diseño del producto de software para su posterior.

- ✓ **Observación:** Se utilizará a la hora de poner en práctica los métodos usados al obtener un valor observable.
- ✓ **Revisión Documental:** Se utilizará para la revisión de las referencias bibliográficas y las bibliografías que se usarán.

Este trabajo de tesis consta de tres capítulos. Estos quedaron estructurados de la siguiente manera:

**Capítulo I:** Fundamentación Teórica. Conceptos fundamentales hasta llegar a los de Confiabilidad y Pruebas de Software.

**Capítulo II:** Estudio de los diferentes modelos de calidad. Identificación de los distintos tipos de productos de software que realiza DATEC. Análisis de los riesgos asociados. Solución propuesta.

**Capítulo III:** Validación del modelo utilizando el método Delphi. Evaluación de la propuesta a través producto del centro.

### 1. Capítulo 1: Fundamentación Teórica

Una necesidad imperiosa y creciente en la actualidad, a la hora de crear un software con calidad, son las pruebas que se realizan para garantizar la calidad adecuada en dicho producto. Es por esto que se hace necesario conocer los conceptos, comenzando por los más generales como la definición de proyectos de software hasta llegar a las especificidades de las características de las pruebas de confiabilidad, así como su clasificación, niveles y tipos de pruebas. Se analizan también puntos de vistas del Project Management Institute (PMI), a través de su guía el Project Management Body of Knowledge (PMBOK).

#### 1.1. Proyectos de Software

Durante muchos años, ante la necesidad del hombre de transformar en soluciones prácticas cada uno de los problemas que enfrentaba, fue necesitando de una guía que dieran una metodología de trabajo y transformara en resultados palpables cada barrera que se imponía en su desarrollo, surge así los inicios del concepto de proyecto, aunque muy variable entre sus definiciones, pudiera verse como proyecto en un ámbito informático a:

- ✓ Un sistema de cursos de acciones simultáneas y/o secuenciales que incluye personas, equipamientos de hardware, software y comunicaciones, enfocadas en obtener uno o más resultados deseables sobre un sistema de información. (1)

Además se caracterizan por:

- ✓ Existe un objetivo claro que se tiene que alcanzar en plazo de tiempo limitado.
- ✓ Se utilizarán recursos de diversos tipos que en su mayoría son limitados.
- ✓ Tiene una fecha de inicio y otra de final.
- ✓ Se requiere una planificación.
- ✓ El producto final tendrá que cumplir las especificaciones.
- ✓ Se desea un determinado nivel de calidad en el producto.

##### 1.1.1. Gestión de Proyectos

La gestión de proyectos se ha convertido en una importante herramienta para el éxito de objetivos mediante la conformación de estructuras manejables. La misma ha sido definida por el Project Management Institute (PMI) como la aplicación de conocimientos, habilidades, herramientas y técnicas sobre un amplio tipo de actividades para alcanzar los requerimientos de un proyecto específico.

La gestión de proyectos se describe además como un proceso de planteamiento, ejecución y control de un proyecto, desde su comienzo hasta su conclusión, con el propósito de alcanzar un objetivo final en un plazo de tiempo limitado, con un coste y nivel de calidad determinados. (2)

Es importante destacar que la gestión de proyectos es cada vez más utilizada a nivel mundial, pues ayuda a mejorar, optimizar y estandarizar los procesos de proyectos. En el mundo actual constituye una excelente herramienta para el logro de objetivos. Muchas son las instituciones reconocidas a nivel mundial que investigan el tema de gestión de proyectos, entre las más importantes, se encuentran:

- El PMI - Project Management Institute
- El IPMA - International Project Management Association
- El AIEPRO - Asociación Española de Ingeniería de Proyectos

### 1.1.2. PMBOK

El Project Management Institute (PMI) es considerado una asociación profesional para la gestión de proyectos. Radica en Pensilvania, Estados Unidos, fundado en 1969, en sus inicios el objetivo principal era identificar las prácticas de gerencias en los proyectos a través de la industria y regir las normas mundiales de la dirección de proyecto. Actualmente sus principales objetivos son formular estándares profesionales, generar conocimiento a través de la investigación, y promover la gestión de proyectos como profesión a través de sus programas de certificación. (3)

Estos objetivos se ven evidenciados en el Project Management Body of Knowledge **PMBOK**, uno de sus más conocidos trabajos. Este libro ha sido insignia de la labor formadora de esta institución, que engloba en gran medida la gestión y dirección de proyectos.

El Project Management Body of Knowledge (libro de estándares para la gestión de proyectos), constituye una suma de conocimientos, habilidades, herramientas y técnicas dentro de la dirección de proyectos, los cuales son aplicados a las actividades de un proyecto para satisfacer sus requerimientos, constituyendo hoy en día, el texto base para la enseñanza de gestión de proyectos, ya que está basado en las mejores prácticas y estándares internacionales. Entre sus principales objetivos se encuentran:

- ✓ Proporcionar referencias básicas acerca de la gestión de proyectos
- ✓ Indicar el conocimiento necesario para manejar el ciclo vital de un proyecto
- ✓ Definir un cuerpo de conocimiento para desarrollar cualquier aplicación (4)

#### Sus procesos y áreas de conocimientos

El PMBOK describe la naturaleza de los procesos de dirección de proyectos en términos de su integración, las interacciones dentro de ellos, y sus propósitos. Estos procesos se dividen en cinco grupos, definidos como los procesos de la dirección de proyectos.

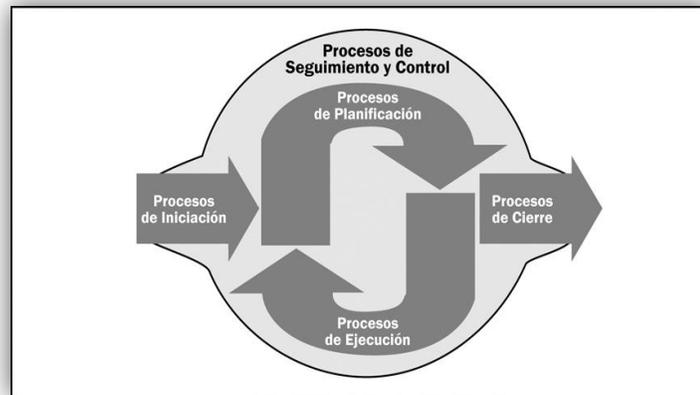


Fig: 1 Grupos de procesos de la gestión de proyectos

Los cinco Grupos de Procesos son:

**Iniciación:** define y autoriza el proyecto o una fase del mismo.

**Planificación:** define y refina los objetivos, planifica el curso de acción requerido para lograr los objetivos y el alcance pretendido del proyecto.

**Ejecución:** integra a personas y otros recursos para llevar a cabo el plan de gestión del proyecto.

**Control:** medición de análisis regular y frecuente del avance del proyecto para identificar variaciones con respecto al plan e implementar acciones correctivas, si fuese necesario.

**Cierre:** formaliza la aceptación del producto, servicio o resultado, y termina ordenadamente el proyecto o una fase del mismo.

Por otra parte el **PMBOK** comprende secciones básicas de conocimientos en su trabajo, necesarias para ciclo de vida de un producto que visualizan el camino en la dirección y gestión de proyecto. La dirección de proyectos es un esfuerzo íntegro entre los 5 procesos y las 9 áreas de conocimiento que lo componen, actúan como un todo donde se complementa el trabajo de una en otra.

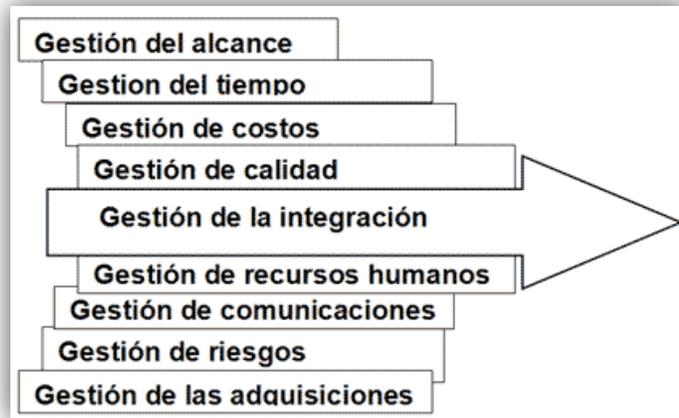


Fig: 2 9 Áreas del Conocimiento.

### Áreas del conocimiento

**Gestión de Integración del Proyecto:** comprende los procesos requeridos para coordinar adecuadamente los elementos de un proyecto. El objetivo principal de la integración de proyectos consiste en tomar decisiones sobre donde concentrar recursos y esfuerzos.

**Gestión de Alcance del Proyecto:** describe las fronteras del proyecto, define lo que el proyecto entregará y también lo que no entregará.

**Gestión de Tiempo del Proyecto:** se centrará la atención en el proceso que abarca todo lo relacionado con la estimación de la duración de actividades.

**Gestión de Estimación de Costes del Proyecto:** describe los procesos involucrados en la planificación, estimación, presupuesto y control de costes de forma que el proyecto se complete dentro del presupuesto aprobado.

**Gestión de Recursos Humanos del Proyecto:** incluye los procesos que organizan y dirigen el equipo del proyecto. Algunos de los puntos que se tratan son: el liderazgo, la delegación, el desarrollo de equipos y la evaluación del desempeño.

**Gestión de las Comunicaciones del Proyecto:** incluye los procesos necesarios para asegurar la generación, recogida, distribución, almacenamiento, recuperación y destino final de la información del proyecto en tiempo y forma.

**Gestión de Riesgos del Proyecto:** el objetivo principal es aumentar la probabilidad y el impacto de los eventos positivos, y disminuir la probabilidad y el impacto de los eventos adversos para el proyecto.

**Gestión de las Adquisiciones del Proyecto:** incluye los procesos necesarios para adquirir los productos, servicios o resultados necesarios, requeridos para la realización de trabajos del proyecto. (2)

## 1.2. Gestión de Calidad del Proyecto

¿Qué es Calidad?

Según Roger Pressman la calidad de un producto software no es más que la:

*“Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente”.* (5)

En la actualidad, un término muy utilizado es el concepto calidad, el cual aparece en disímiles contextos y con el que se despierta en quien lo escucha una sensación positiva, transmitiendo ideas de excelencia. El concepto “calidad” representa más bien una forma de hacer las cosas en las que, fundamentalmente, predominan la preocupación por satisfacer al cliente y por mejorar, procesos y resultados.

La gestión de la calidad es parte del tema central de investigación, incluye los elementos necesarios para el aseguramiento del producto y del proyecto.

El PMBOK plantea que el sistema de gestión de calidad genera los medios para obtener los indicadores que permitan medir la efectividad, tanto del producto, como el equipo de desarrollo, por ello, esta área es una de las más importantes, la misma está soportada por tres actividades básicas, que describen las pautas a seguir.

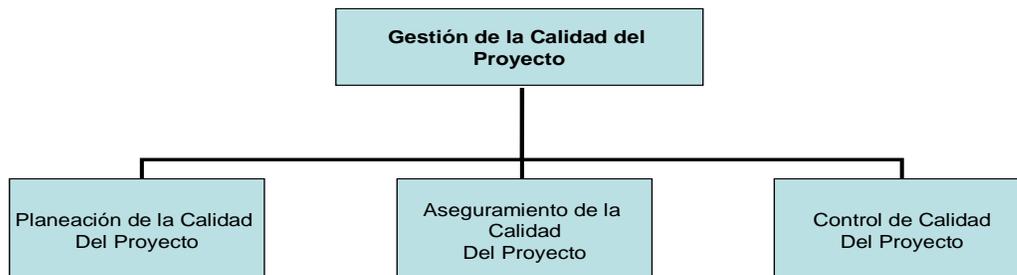


Fig: 3 Procesos de la gestión de calidad de proyectos

este sistema implementa a través de la política, los procedimientos y los procesos de planificación de calidad, aseguramiento de calidad y control de calidad, con actividades de mejora continua de los procesos que se realizan durante todo el proyecto, estos procesos son:

- ✓ **Planificación de Calidad:** identificar qué normas de calidad son relevantes para el proyecto y determinando cómo satisfacerlas.

- ✓ **Realizar Aseguramiento de Calidad:** aplicar las actividades planificadas y sistemáticas relativas a la calidad, para asegurar que el proyecto utilice todos los procesos necesarios para cumplir con los requisitos.
- ✓ **Realizar Control de Calidad:** supervisar los resultados específicos del proyecto, para determinar si cumplen con las normas de calidad relevantes e identificar modos de eliminar las causas de un rendimiento insatisfactorio. (2)

### 1.2.1. Aseguramiento de la Calidad

Los sistemas de calidad son el conjunto que reúne estructura, responsabilidades, actividades, recursos y procedimientos de la organización de una empresa, que fueron establecidas por ésta para llevar a cabo la gestión de calidad.

Las actividades de aseguramiento de la calidad del software (SQA) son básicamente las siguientes:

- Validaciones (Asegurar que los productos generados cumplan las expectativas de los clientes)
- Verificaciones (Asegurar que los productos generados sean técnicamente correctos). (6)

Para lograr el éxito en la producción de software, hay que hacerlo con calidad. Esto sólo es posible con la implantación de un sistema para el aseguramiento de la calidad del software. Cualquiera puede preguntarse cómo llevar a cabo un plan de esta envergadura, son precisamente las normas internacionales las que contribuyen a que esto sea posible. Por lo general, las que dirigen la línea organizativa de las compañías y la estandarización de calidad de los productos.

### 1.2.2. Organización Internacional para la Estandarización (ISO)

ISO es la Organización Internacional de Estandarización (en algunos casos Organización Internacional de Normalización), fue creada para medir y asegurar la calidad en la producción. Suele llamársele así por ser las siglas en inglés de International Standards Organization.

La norma ISO es reconocida a nivel mundial, pues conforman la base para el establecimiento de un sistema de gestión de calidad, permitiendo la amortización a nivel internacional de la calidad como un elemento imprescindible de los intercambios comerciales. La norma ISO incluye servicios y productos, en todos los campos del hacer humano (Ingeniería, Biología, Medicina, etc.). A continuación se recogen algunos de sus objetivos principales:

- Garantizan que un proveedor tenga la capacidad de producir los bienes y/o servicios requeridos, satisfaciendo las expectativas de los clientes.

- Facilita y promueve la actividad comercial e industrial.
- Impulsa a los trabajadores de la organización a un mejoramiento sucesivo.
- Optimiza las operaciones y procesos elaborados en la organización (se incrementa la eficiencia).
- Simplifica significativamente los costos ya que elimina desperdicios e ineficiencias de los sistemas y procesos.
- Fortalece la imagen de la empresa.

Esto se evidencia en las normas ISO 9000, que constituyen una familia de normas internacionales, enmarcado en las buenas prácticas de calidad, por ejemplo:

- ❖ **ISO 9000 1** es aplicable durante la implantación inicial del sistema, muestra una guía clara sobre los diferentes capítulos de las normas citadas.
- ❖ **ISO 9000 2** es utilizada en el funcionamiento y mantenimiento del software.
- ❖ **ISO 9000 3** está referida a la gestión de programas de seguridad de funcionamiento, la seguridad de funcionamiento de los productos.
- ❖ **ISO 9000 4** contiene directrices para relacionar los requisitos genéricos de los elementos del sistema de la calidad con los requisitos específicos de un producto. (7)

### **CMMI**

Por otra parte, mundialmente se conocen los modelos de calidad conocidos como SW CMM (Software Capability Maturity Model o en español, Modelo de Madurez de La Capacidad de Desarrollo del Software). Es un modelo de referencia que se diferencia de otros modelos por el hecho de estar basado en prácticas ajustables a cualquier dominio de producción y poseer un enfoque global e integrado de la organización, con el propósito de alcanzar los objetivos del negocio. De esa forma CMMI permite a empresas complejas compuestas por varias áreas de negocio instaurar de una forma más sencilla un sistema de aseguramiento de la calidad. (8)

### **SPICE**

Software Process Improvement and Capability (SPICE) es un modelo de calidad de suma importancia, se caracteriza por la evaluación de los procesos de software para la mejora continua, valoración de la capacidad de cada software y sirve como base para el comercio internacional de software. Su trayectoria está enmarcada a ejecutar, planificar, gestionar, controlar, y mejorar los procesos de adquisición, suministro, desarrollo, operación y soporte.

1.2.3. ISO 9126

Uno de los estándares más usados para lograr la calidad de productos de software es la norma **ISO 9126**, es un estándar internacional para la evaluación del Software. Actualmente en nuestro país, la Oficina Nacional de Normalización (NC), es el organismo nacional de normalización de la República de Cuba que representa al país ante las organizaciones internacionales y regionales de normalización. El estándar está dividido en cuatro partes bajo el título general de “*Ingeniería de Software Calidad del Producto*”, las cuales dirigen respectivamente, lo siguiente: *modelo de calidad, métricas externas, métricas internas y calidad en las métricas de uso*. Las métricas externas corresponden a proporcionar productos o servicios que satisfagan las expectativas del cliente para establecer lealtad con el cliente, por otro lado, las métricas internas tienen como principal objetivo implementar los medios para permitir la mejor descripción posible de la organización y detectar y limitar los funcionamientos incorrectos.

<p><b>Funcionalidad</b></p> <ul style="list-style-type: none"> <li>• Adecuación</li> <li>• Precisión</li> <li>• Interoperabilidad (compatibilidad)</li> <li>• Seguridad de acceso</li> <li>• Conformidad de funcionalidad</li> </ul>	<p><b>Eficiencia</b></p> <ul style="list-style-type: none"> <li>• Manejo del tiempo.</li> <li>• Manejo de los recursos.</li> <li>• Conformidad de eficiencia</li> </ul>
<p><b>Confiabilidad</b></p> <ul style="list-style-type: none"> <li>• Madurez</li> <li>• Tolerancia a fallas</li> <li>• Recuperación</li> <li>• Conformidad de confiabilidad</li> </ul>	<p><b>Mantenibilidad</b></p> <ul style="list-style-type: none"> <li>• Facilidad de análisis</li> <li>• Facilidad para introducir cambios</li> <li>• Estabilidad</li> <li>• Facilidad de prueba</li> </ul>
<p><b>Facilidad de uso</b></p> <ul style="list-style-type: none"> <li>• Facilidad de comprensión.</li> <li>• Facilidad de aprendizaje.</li> <li>• Operatividad.</li> <li>• Atractivo</li> <li>• Conformidad de facilidad de uso</li> </ul>	<p><b>Portabilidad</b></p> <ul style="list-style-type: none"> <li>• Adaptabilidad.</li> <li>• Instalabilidad.</li> <li>• Co-existencia</li> <li>• Compatibilidad o facilidad de reemplazo.</li> <li>• Conformidad de portabilidad</li> </ul>

Fig: 4 Atributos de calidad externa e interna.

**Funcionalidad:** Es la capacidad del producto de software para entregar funciones que cumplan con las necesidades explícitas e implícitas cuando el software se usa bajo condiciones específicas.

- ¿Las funciones implementadas están adecuadas a los objetivos del software?
- ¿Las tareas ejecutadas por las funciones existentes generan resultados deseados o correctos?

**Eficiencia:** Es la capacidad del producto de software para entregar un desempeño apropiado, en relación con la cantidad de recursos utilizados, bajo condiciones establecidas.

- ¿Cuánto demora el despliegue de la información de consultas?

**Mantenibilidad:** Es la capacidad del producto de software para ser modificado. Estas modificaciones pueden incluir correcciones, mejoras o adaptaciones del software a los cambios en el ambiente, en requisitos y en especificaciones funcionales.

- ¿Existe documentación que facilite entender el software?
- ¿El código está auto documentado?

**Portabilidad:** Es la capacidad del producto de software para ser transferido desde un ambiente a otro.

- ¿Existen manuales de instalación?
- ¿Están identificados todos los archivos, documentos, etc. para reinstalar el software en otro computador?

**Facilidad de uso:** Es la capacidad del producto de software para ser fácil de comprender, aprender y usar por el usuario y ser atractivo para él, bajo condiciones específicas de uso.

- ¿Un usuario nuevo puede comprender el sentido del software para empezar a usarlo?
- ¿Se requieren manuales para usarlo?

**Confiabilidad:** Es la capacidad del producto de software para evitar fallas como resultado de errores de software.

- ¿El software presenta fallas en uso normal de este y con qué frecuencia 'se cae'?
- ¿Cómo se comporta el software frente a un uso anormal del software, por ejemplo, ingresando valores inválidos, realizando acciones incorrectas?
- ¿En caso de producirse una falla, se recupera rápidamente? (9)

### 1.3. Confiabilidad

¿Confiabilidad, Fiabilidad o Reliability?

Aunque parezca trivial aparecen en múltiples contextos estos términos, es interesante observar la diferencia terminológica que existe a ambos lados del Atlántico. Mientras el original en inglés (*reliability*) ha sido traducido en España como *fiabilidad*, en Latinoamérica se acostumbra utilizar la traducción **confiabilidad**. ¿Cuál es la correcta? Se puede asumir entonces que no existe una respuesta definitiva,

porque todas infieren a un mismo concepto (definido como la *probabilidad de buen funcionamiento de algo*).

La importancia que adquiere el software, nos obliga siempre a tratar de mejorar y buscar confiabilidad en los sistemas. En muchos factores el funcionamiento de un programa en un entorno determinado, depende de que sea fiable, que en caso de fallar inesperadamente puede causar pérdidas millonarias en dinero y vidas humanas.

En el mundo moderno, el concepto de “confiabilidad” adquiere gran trascendencia. Confiabilidad es la “capacidad de un ítem de desempeñar una función requerida, en condiciones establecidas”. Habremos logrado la Confiabilidad requerida cuando el “ítem” hace lo que queremos que haga. Al decir “ítem” podemos referirnos a una máquina, a una planta industrial, a un proceso, a un rodado, a un sistema y también a una persona. La confiabilidad impacta directamente sobre los resultados de cualquier sistema, debiendo aplicarse no sólo a máquinas o equipos aislados sino a la totalidad de los procesos que integran la cadena de valor de la Organización. (10)

Algunos conceptos definidos para confiabilidad son:

- Se refiere a la capacidad del software de mantener su nivel de ejecución bajo condiciones normales en un período de tiempo establecido. (11)
- Se define como la probabilidad de que un programa realice su objetivo satisfactoriamente (sin fallos) en predeterminado período de tiempo y en un entorno concreto (denominado perfil operacional). (12)

Tomando en cuenta los conceptos planteados se puede llegar a la conclusión que, a través del estudio de la confiabilidad de un producto de software, es posible reconocer si el producto conserva buen desempeño ante las diversas circunstancias que se pueden presentar, verificando siempre que el producto reaccione positivamente ante las posibles fallas que pueda presentarse por defectos en el programa o causas externas, protegiendo siempre la integridad de la información.

### 1.3.1. Análisis de la confiabilidad

La ejecución de un análisis de la confiabilidad en un producto o un sistema debe incluir muchos tipos de exámenes, para determinar cuan confiable es el producto o sistema que pretende analizarse.

Los diversos estudios del producto se relacionan, vinculan y examinan conjuntamente, para poder determinar la confiabilidad del mismo bajo todas las perspectivas posibles, determinando problemas, para poder sugerir correcciones, cambios y/o mejoras en productos o elementos.

¿Por qué fallan los productos de software?

El manejo de fallas es una llave en la ingeniería de la confiabilidad. Los modelos, *fallas validadas*, son esenciales para el desarrollo de técnicas de predicción, procedimientos de asignación, diseño y análisis de metodologías, pruebas, demostración y control de procedimientos. En otras palabras, todos los elementos necesarios como entradas para asegurar que un producto pueda ser diseñado y manufacturado de manera que se lleve a cabo satisfactoria y económicamente.

En DATEC, centro al cual está dirigida nuestra investigación, los productos que se desarrollan son empaquetados, (se refiere a las aplicaciones diseñadas para realizar una tarea de procesamiento particular que se considera un todo, incluyendo: manuales de usuario, instalación y otros elementos definidos para su comercialización), y no deben llegar al cliente con defectos significativos a menos que sea pactado con el cliente y este por escrito emita su conformidad con el defecto. Es necesario por esta razón que los productos que se desarrollen en el centro sean confiables.

Para que sean defectos no significativos:

- ✓ Los defectos detectados no trascienden el lugar donde estos se evidencian.
- ✓ Los defectos no impiden la ejecución satisfactoria de una funcionalidad.
- ✓ Los defectos no impiden que todos los flujos básicos de la aplicación se recorran evidenciando el cumplimiento de los requisitos funcionales descritos.

Se consideran críticos aquellos productos que:

Se consideran críticos aquellos productos que:

- ✓ No se presenta la última versión del producto comprobada y avalada por la etapa anterior de la metodología.
- ✓ No están presentes todos los elementos componentes del sistema, producto o entregable. (Hardware, Software, Partes y Documentación).
- ✓ No se corresponden totalmente los requisitos funcionales documentados con los implementados.
- ✓ Excede el producto el número máximo de iteraciones establecidas por plan de pruebas (entre 2 y 3 iteraciones).
- ✓ Existe incoherencia significativa entre los artefactos que tienen relación o dependencia entre sí.
- ✓ Existe incoherencia significativa entre lo documentado y lo implementado.

- ✓ La cantidad de faltas de ortografía excede la cantidad de páginas o pantallas que tiene el artefacto en cuestión.
- ✓ Durante las pruebas de regresión persisten al menos 2 defectos significativos de iteraciones anteriores.
- ✓ Se observan los mismos tipos de defectos, ya señalados en iteraciones anteriores, en otros o los mismos lugares donde fueron detectados, para el primer día de pruebas:  $\frac{1}{2}$  de la cantidad de defectos tipo encontrados en etapa anterior, o para todo el proceso de pruebas:  $\frac{1}{4}$  de la cantidad de defectos tipo encontrados en etapa anterior. (13)

### 1.4. Pruebas de Software

Con el fin de elevar la calidad de los productos desarrollados en DATEC, se realizan un conjunto de actividades, jugando las pruebas un papel fundamental, razón por la cual están enmarcadas dentro del proceso de desarrollo del software, con la intención de garantizar la calidad del mismo y satisfacer los requerimientos especificados. Mediante las pruebas es posible localizar la mayor cantidad de deficiencias de un producto, permitiendo que se puedan subsanar en el menor tiempo posible. Existen normas internacionales que se encargan de definir las como el Instituto de Ingenieros Eléctricos y Electrónicos: (IEEE 1012-1998), la (IEEE 829-1998) y por algunos autores como (Pressman, 2005).

Según (Pressman, 2005) las pruebas son: *“El proceso de ejecución de un programa con la intención de descubrir un error.”*

La prueba del software es un elemento crítico para la garantía de la calidad del software. La prueba no es una actividad sencilla, no es una etapa del proyecto en la cual se asegura la calidad, la prueba debe ocurrir durante todo el ciclo de vida del proyecto, podemos probar la funcionalidad de los primeros prototipos, probar la estabilidad, cobertura y rendimiento de la arquitectura, probar el producto final, etc. Lo que conduce al principal beneficio de la prueba, proporcionar una reacción mientras hay todavía tiempo y recursos para hacer algo. La prueba no puede asegurar la ausencia de defectos sólo puede demostrar que existen defectos en el software.

En fin las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo condiciones específicas, donde se valida la calidad y funcionalidad del producto, mediante el análisis y evaluación de los resultados de dicha ejecución; con el fin de detectar errores que impidan que dicho producto responda con las necesidades especificadas previamente. Y un proceso de prueba es el establecimiento de los objetivos de las pruebas, el diseño de los casos de pruebas a aplicar, la codificación de los casos de pruebas, la ejecución de los mismos y el análisis de los resultados de dicha ejecución.

### Causas que fundamentan las pruebas

Dentro de las causas que fundamentan las pruebas las más importantes son:

- ✓ **Propensión a equivocarse:** El ser humano es propenso a cometer equivocaciones, éstas se manifiestan en diversos problemas contenidos en los modelos (defectos o fallas) y pueden manifestarse como fallas en tiempo de ejecución.
- ✓ **Fallas de hardware:** La infraestructura empleada para el desarrollo de software (hardware, sistemas operativos, compiladores, etc.) no está exenta de fallas, lo que introduce defectos adicionales o permite que subsisten inadvertidos los que introdujo el desarrollador.
- ✓ **Creativa de desarrollo:** El desarrollo del software es una labor creativa y por ello es común que el producto desarrollado no coincida con el modelo contenido en las especificaciones.

### Objetivos de las Pruebas

Dentro de los objetivos fundamentales que se persiguen al aplicarles las pruebas a un software se encuentran las siguientes:

- Encontrar defectos
- Lograr confianza acerca del nivel de calidad
- Proveer información
- Prevenir defectos

### Etapas del proceso de las pruebas

El proceso de pruebas de un software consta de varias etapas dentro de ellas la más importantes están:

- Inspección del análisis (verifica si se cometieron errores o falla en la etapa de análisis)
- Inspección del diseño (debe ser completo y eficiente)
- Inspección del código (observar el entendimiento y facilidad del código)
- Pruebas unitarias (probar cada método de la clases implementados por separado)
- Pruebas de integración (probar todas las clases)
- Pruebas de validación de requerimientos (verificar que cumple con todos los requerimientos exigidos por cliente)
- Pruebas de sistemas (ejecutar el programa para verificar si cumple con los requisitos exigidos)

### Estrategias del proceso de pruebas de software

Una estrategia de pruebas de software integra las técnicas de diseño de casos de pruebas en una serie de pasos bien planificados que llevan a una construcción correcta del software.

El diseño efectivo de casos de prueba es importante, pero también lo es la estrategia para su utilización. En el proyecto la prueba a veces requiere más esfuerzo que cualquier otra actividad de la ingeniería de software. Si se efectúa sin un plan, el tiempo se desaprovecha y el esfuerzo es consumido innecesariamente y, en el peor de los casos, los errores inadvertidos quedan sin detectar. Por tanto, parece razonable establecer una estrategia sistemática para probar el software.

### 1.4.1. Plan de pruebas

El propósito del plan de pruebas es dejar de forma explícita el alcance, el enfoque, los recursos requeridos, el calendario, los responsables y el manejo de riesgos en un proceso de pruebas.

Está constituido por un conjunto de pruebas, en las que se debe:

- Dejar claro que tipo de propiedades se quieren probar (*corrección, robustez, fiabilidad*)
- Dejar claro cómo se mide el resultado.
- Especificar en qué consiste la prueba (hasta el último detalle de como se ejecuta)
- Definir cuál es el resultado que se espera (*identificación, tolerancia*)

Las pruebas carecen de utilidad, tanto, si no se sabe exactamente lo que se quiere probar, sí no se está claro como se prueba, o si el análisis del resultado se hace a simple vista. Por ello es muy importante destacar y tener en cuenta que mientras se realicen las pruebas de confiabilidad, estarán seguros los productos de cualquier sistema.

### 1.4.2. Tipos de Pruebas

Existen diferentes tipos pruebas, por las cuales se puede evaluar cualquier producto de software, teniendo en cuenta la dificultad que puede tener el sistema. Por ejemplo:

**Pruebas de Integración:** son aquellas que se realizan en el ámbito del desarrollo de software una vez que se han aprobado las pruebas unitarias. Únicamente se refieren a la prueba o pruebas de todos los elementos unitarios que componen un proceso, hecha en conjunto, una sola vez. (14)

**Pruebas de Validación:** se realizan sobre un software completamente integrado para evaluar el cumplimiento con los requisitos especificados.

**Pruebas de Sistema:** es donde el software ya validado se integra con el resto del sistema donde algunos tipos de pruebas a considerar son:

- ✓ Rendimiento: determinan los tiempos de respuesta, el espacio que ocupa el módulo en disco o en memoria, el flujo de datos que genera a través de un canal de comunicaciones, etc.
- ✓ Resistencia: determinan hasta donde puede soportar el programa determinadas condiciones extremas.
- ✓ Robustez: determinan la capacidad del programa para soportar entradas incorrectas.
- ✓ Seguridad: se determinan los niveles de permiso de usuarios, las operaciones de acceso al sistema y acceso a datos.
- ✓ Usabilidad: se determina la calidad de la experiencia de un usuario en la forma en la que este interactúa con el sistema, se considera la facilidad de uso y el grado de satisfacción del usuario.
- ✓ Instalación: se determinan las operaciones de arranque y actualización del software.

**Pruebas de Aceptación:** son las que harán los clientes, para determinar que el sistema cumple con lo deseado y obtener la conformidad.

**Pruebas Unitarias:** se encargan de probar una clase en concreto, testeando cada uno de sus métodos y viendo si dados unos parámetros de entrada, la salida es la esperada.

**Pruebas Funcionales:** como su propio nombre indica, prueban una funcionalidad completa, donde pueden estar implicadas una o varias clases o la propia interfaz de usuario.

**Pruebas de Regresión:** son aquellas pruebas cuyo objetivo es comprobar por qué ha dejado de funcionar algo que ya funcionaba. El objetivo de las pruebas de regresión es no tener que “volver atrás”. (14)

### 1.5. Pruebas de Confiabilidad

¿Qué son las pruebas de confiabilidad?

Consiste en probar una aplicación para descubrir y eliminar errores antes de que se ponga en marcha el sistema.

Pues existen infinidad de combinaciones distintas de recorridos alternativos a lo largo de una aplicación. Sin embargo es poco probable que encuentren todos los errores posibles de una aplicación compleja. No obstante, probar las situaciones más probables bajo condiciones normales de uso es garantizar que la aplicación proporciona el servicio previsto. Si se dispone de tiempo suficiente, pueden realizarse pruebas más complicadas para detectar defectos menos evidentes.

El objetivo de las pruebas de confiabilidad es descubrir posibles problemas con el diseño tan pronto como sea posible y, en última instancia, ofrecer confianza en que el sistema satisface sus requisitos de confiabilidad.

Un aspecto clave de las pruebas de confiabilidad es definir "fracaso". Aunque esto pueda parecer obvio, hay muchas situaciones donde no está claro si un fallo es realmente la culpa del sistema. Las variaciones en las condiciones de prueba, las diferencias del operador, el clima, y situaciones inesperadas crear diferencias entre el cliente y el desarrollador del sistema.

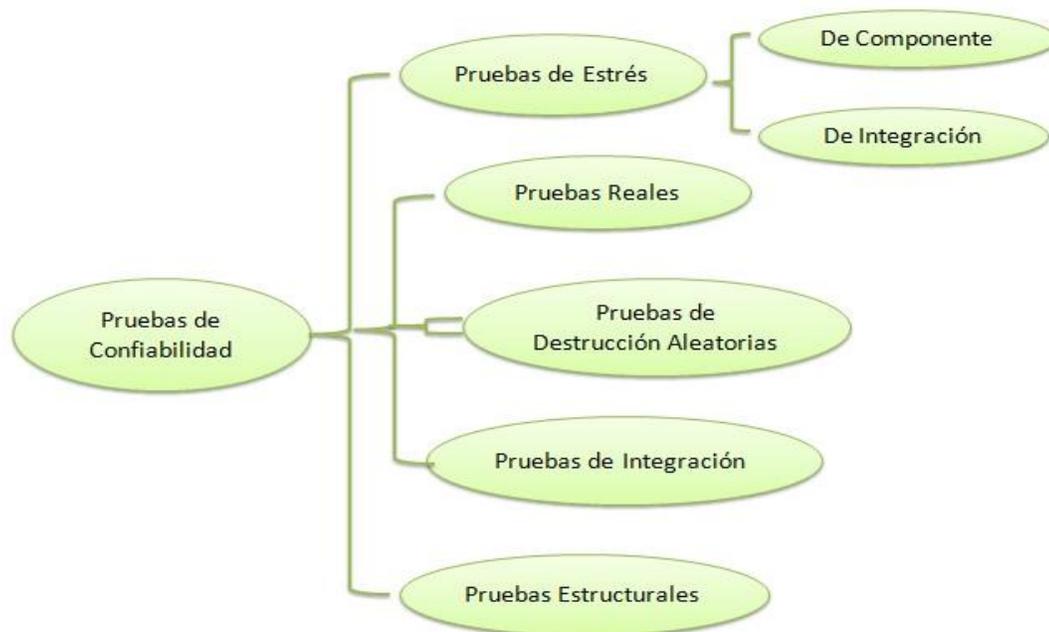


Fig: 5 Tipos de pruebas de confiabilidad

### 1.6. Riesgos

Actualmente en los proyectos de desarrollo de software, la gestión de riesgos es un punto importante para obtener buenos resultados. Llevar a cabo una buena gestión de riesgos, da paso e influye exponencialmente en la madurez de un proyecto.

¿Qué es riesgo?

El riesgo es la probabilidad de que una amenaza se convierta en un desastre. La vulnerabilidad o las amenazas, por separado, no representan un peligro. Pero si se juntan, se convierten en un riesgo, o sea, en la probabilidad de que ocurra un desastre. (15)

### 1.6.1. Tipos de Riesgos

Hay otra clasificación de los riesgos más general que engloba los riesgos en tres clases principales:

Los *Riesgos de Proyecto* amenazan al plan de proyecto, pues, identifican fundamentalmente problemas potenciales con el presupuesto, organización del personal, planificación temporal y su impacto en un proyecto de software.

Los *Riesgos Técnicos* por su parte, amenazan con la calidad y la planificación temporal del software. Estos riesgos identifican problemas potenciales de diseño, interfaz, implementación, mantenimiento y ambigüedades de especificaciones

Los *Riesgos del Negocio* amenazan contra la viabilidad del software a construir, por lo que a menudo ponen en peligro el proyecto o el producto.

### 1.6.2. Identificación de Riesgos

Actualmente los riesgos pueden ocurrir bajo cualquier circunstancia, y más aún cuando no son previstos con antelación. Es por ello que se hace necesario contar con lineamientos que manejen este proceso para llevar un plan para la mitigación de los riesgos que puedan afectar o que ponga en peligro los intereses del proyecto. Llevar una gestión de riesgo asociado a la confiabilidad, en este caso en el centro productivo DATEC, es uno de los puntos claves para lograr que el sistema cumpla con lo pactado con el cliente.

En el centro DATEC, actualmente existen diferentes tipos de riesgos que atentan contra la confiabilidad de los productos. Por ejemplo, riesgos de recuperabilidad, madurez, tolerancia ante fallas, exactitud, precisión, modularidad, consistencia, simplicidad; una vez que falle cualquiera de los atributos mencionados, fallaría la confiabilidad del producto, pudiendo esto causar la inconformidad del cliente con el producto.

### Conclusiones Parciales

Construir software fiable y con calidad es una actividad siempre llena de dificultades, sin embargo, el estudio de los problemas y fallos que presentan los software, unido al avance tecnológico de los últimos años y el desempeño de cada grupo de investigación, ha dado lugar a un gran número de propuestas, métodos, modelos, metodologías y herramientas en el ámbito profesional informático.

Puede inferirse que la evolución que ha tenido la calidad, la ha convertido en un aspecto relevante en el desarrollo del software y que la confiabilidad como parte de ella influye significativamente en el resultado del producto.

### 2. Capítulo 2: Propuesta de un Modelo de Pruebas de Confiabilidad

El uso eficiente de todos los factores y técnicas que intervienen en la evaluación de la calidad del software son objetivos aún distantes, y esto lo evidencian los reportes de fracaso y dificultades de muchos proyectos que después de haberse puesto a prueba tienen errores, necesitan cambios o sencillamente carecen de seguridad, por lo que no son productos fiables. Se dice que el éxito de un programa está en el análisis profundo que se haga en cada factor que integra la calidad y en el empleo de la metodología de desarrollo que se utilice para guiar el proceso, así como las técnicas que se empleen durante el mismo.

#### 2.1. Modelos

La elaboración de un modelo, es un proceso que se utiliza con frecuencia de forma inconsciente. El uso de información es un elemento clave para la dirección eficiente en cualquier organización. De igual forma al tomar decisiones para resolver problemas presentes pero con repercusiones en el futuro; se hace necesaria la construcción de un modelo.

¿Qué es un modelo?

Muchos han sido los criterios dados para conceptualizar los modelos. Las siguientes definiciones así lo demuestran:

- "Es la abstracción de los sucesos que rodean un proceso, una actividad, o un problema. Aíslan una entidad de su entorno de tal manera que pueda examinarse sin el "ruido" o perturbación de otras influencias del medio circundante".
- "Es una representación ideal y concreta de un objeto, de un fenómeno con fines de estudio y experimentación. Es un concepto de valores, creencias y normas de conducta que condicionan la actuación y el modo de pensar de todos".

Un modelo es un *“Sistema de elementos que reproduce determinados aspectos, relaciones y funciones del objeto que se investiga; desarrollado en un nivel avanzado del conocimiento, en el que recopila las características generales del objeto investigado y las unifica en un concepto global, del cual se puede visualizar el objeto en un momento dado.”* (16)

#### Características generales de los modelos

- Referencia a un criterio de uso.
- Indicación de su grado de terminación.

- Capacidad de aproximarse al funcionamiento real del objeto (validez y confiabilidad)
- Capacidad para incluir los cambios que se operan en la realidad (utilidad y permanencia)
- Capacidad referencial. Tener en cuenta la dependencia que tienen respecto al sistema social en el que se inserta.

### Tipos de Modelos

Existen diferentes criterios clasificatorios. A continuación se representa el resultado de una de las clasificaciones más generales utilizadas.

- ✓ **Icónico:** Es una reproducción a escala del objeto real; muestra su misma figura, proporciones y características.
- ✓ **Analógico:** No es una reproducción detallada de todas las cualidades del sistema real, sino que refleja solamente la estructura de relaciones y determinadas propiedades fundamentales de la realidad.
- ✓ **Teórico:** Utiliza símbolos para designar las propiedades del sistema real que se desea estudiar. Tiene la capacidad de representar las características y relaciones fundamentales del fenómeno, proporcionar explicaciones y servir como guía para generar hipótesis teóricas.

### Importancia de los modelos

La importancia de los modelos está fundamentada en dos ventajas que guardan estrecha relación entre sí, pero que no son idénticas.

- ✓ La primera es el ahorro en la presentación y en la búsqueda. Así, es más barato representar visualmente el plano de una fábrica o de un sistema de información administrativa que construir uno, y también hacer modificaciones de ese sistema mediante rediseño.
  - ✓ La segunda consiste en que los modelos permiten analizar y experimentar situaciones tan complejas en una forma que resultaría imposible si se reprodujera el sistema y su ambiente real.
- (17)

### **2.2. Resumen de los Principales Modelos de Calidad**

Un modelo de calidad integra un grupo de características y relaciones entre ellas que proporcionan la base para especificar requisitos de calidad y evaluarla.

Pressman (2002) indica que los factores que afectan a la calidad del software no cambian, por lo que resulta útil el estudio de los modelos de calidad que han sido propuestos en este sentido desde los años 70. Dado que los factores de calidad presentados para ese entonces siguen siendo válidos, se estudiarán los modelos más importantes propuestos hasta ahora: McCall (1977), Boehm (1978), FURPS (1987), Dromey (1996) y el estándar propuesto por la Norma ISO/IEC 9126 (1991).

### 2.2.1. Modelo McCall

El modelo propuesto por **Jim McCall** en 1977, uno de los modelos más usados, se basa en tres perspectivas generales, que son utilizadas para definir e identificar la calidad de un producto de software: *operación del producto*, *revisión del producto* y *transición del producto*, desde los cuales el usuario puede contemplar la calidad de un producto. La fiabilidad en este modelo, se enmarca en tres atributos fundamentales.

*Precisión*: Atributos del software que proporcionan el grado de precisión requerido en los cálculos y los resultados.

*Tolerancia a fallos*: Atributos del software que posibilitan la continuidad del funcionamiento bajo condiciones no usuales.

*Simplicidad*: Atributos del software que posibilitan la implementación de funciones de la forma más comprensible posible. (18)

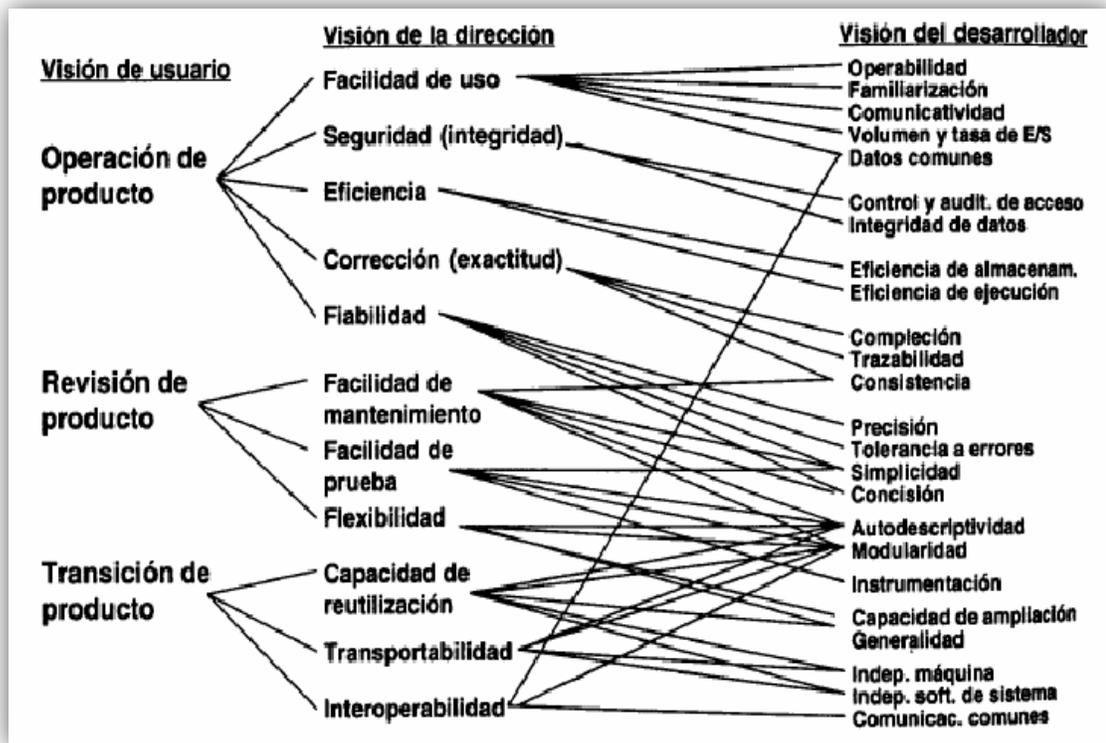


Fig: 6 Modelo McCall

### 2.2.2. Modelo Boehm

El Modelo **Boehm** es un modelo fijo sin posibilidad de ser modificado o adaptado por el usuario. Los criterios y factores son determinados y fijos, de forma que la medida de la calidad debe ajustarse a estas definiciones y a las relaciones entre criterios y factores de calidad que el modelo propone. El factor fiabilidad es uno de los criterios más importantes a la hora de medir el producto final, los atributos que lo evalúan son:

*Exactitud:* es la precisión de los cálculos y emisión de resultados.

*Compleitud:* grado en que han sido implementadas las funciones para el sistema.

*Consistencia:* el grado en que las especificaciones externas se corresponden con las especificaciones internas. (19)

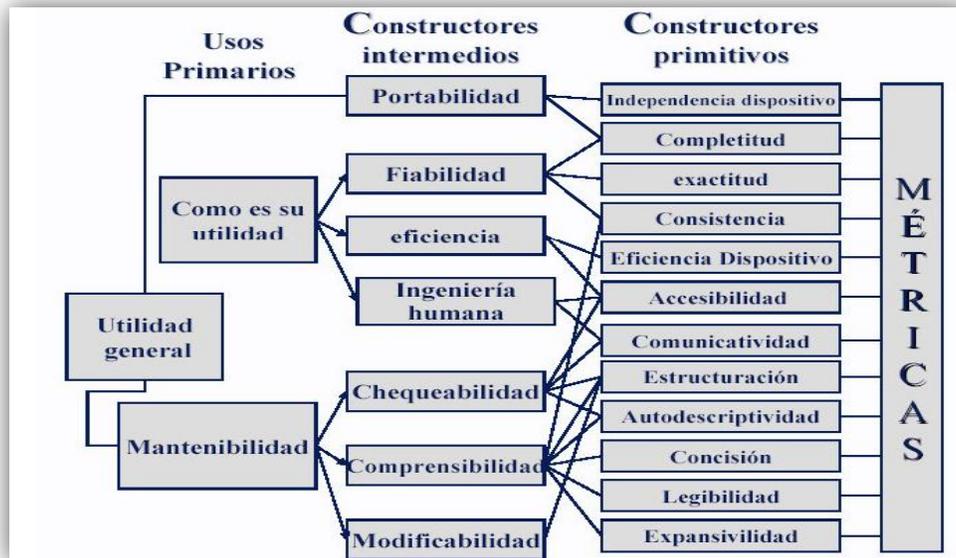


Fig: 7 Modelo Boehm

### 2.2.3. Modelo Hewlett-Packard

El modelo de McCall ha servido de base para modelos de calidad posteriores, y este es el caso del modelo **FURPS**, producto del desarrollo de Hewlett-Packard (Robert Grady, 1987). En este modelo se desarrollan un conjunto de factores de calidad de software, bajo el acrónimo de **FURPS**: funcionalidad (Functionality), usabilidad (Usability), confiabilidad (Reliability), desempeño (Performance) y capacidad de soporte (Supportability). La tabla 5 presenta la clasificación de los atributos de calidad que se incluyen en el modelo, junto con las características asociadas a cada uno (Pressman, 2002)

El modelo **FURPS** incluye, además de los factores de calidad y los atributos, restricciones de diseño y requerimientos de implementación, físicos y de interfaz. Su principal aporte es que divide los factores principales de evaluación en dos grupos: los basados en requerimientos funcionales y los basados en requerimientos no funcionales. (20)

Factor de Calidad	Atributos
Funcionalidad	<ul style="list-style-type: none"> <li>✓ Características y capacidades del programa</li> <li>✓ Generalidad de las funciones</li> <li>✓ Seguridad del sistema</li> </ul>
Facilidad de uso	<ul style="list-style-type: none"> <li>✓ Factores humanos</li> <li>✓ Factores estéticos</li> <li>✓ Consistencia de la interfaz</li> <li>✓ Documentación</li> </ul>
Confiabilidad	<ul style="list-style-type: none"> <li>✓ Frecuencia y severidad de las fallas</li> <li>✓ Exactitud de las salidas</li> <li>✓ Tiempo medio de fallos</li> <li>✓ Capacidad de recuperación ante fallas</li> <li>✓ Capacidad de predicción</li> </ul>
Rendimiento	<ul style="list-style-type: none"> <li>✓ Velocidad del procesamiento</li> <li>✓ Tiempo de respuesta</li> <li>✓ Consumo de recursos</li> <li>✓ Rendimiento efectivo total</li> <li>✓ Eficacia</li> </ul>
Capacidad de Soporte	<ul style="list-style-type: none"> <li>✓ Extensibilidad</li> <li>✓ Adaptabilidad</li> <li>✓ Capacidad de pruebas</li> <li>✓ Capacidad de configuración</li> <li>✓ Compatibilidad</li> <li>✓ Requisitos de instalación</li> </ul>

Fig: 8 Atributos de calidad Modelo FURPS

#### 2.2.4. Modelo Dromey

Dromey en el año 1996 propuso un marco de referencia o metamodelo para la construcción de modelos de calidad, basado en cómo las propiedades medibles de un producto de software pueden afectar los atributos de calidad generales, como por ejemplo, confiabilidad y mantenibilidad. El problema que se plantea es cómo conectar tales propiedades del producto con los atributos de calidad de alto nivel. Para solventar esta situación, Dromey sugiere el uso de cuatro categorías que implican propiedades de calidad, que son: correctitud, internas, contextuales y descriptivas.

La siguiente Tabla presenta la relación que establece Dromey entre las propiedades de calidad del producto y los atributos de calidad de alto nivel.

<b>Propiedades del producto</b>	<b>Atributos de Calidad</b>
Correctitud	<ul style="list-style-type: none"> <li>✓ Funcionalidad</li> <li>✓ Confiabilidad</li> </ul>
Internas	<ul style="list-style-type: none"> <li>✓ Mantenibilidad</li> <li>✓ Eficiencia</li> <li>✓ Confiabilidad</li> </ul>
Contextuales	<ul style="list-style-type: none"> <li>✓ Mantenibilidad</li> <li>✓ Reusabilidad</li> <li>✓ Portabilidad</li> <li>✓ Confiabilidad</li> </ul>
Descriptivas	<ul style="list-style-type: none"> <li>✓ Mantenibilidad</li> <li>✓ Reusabilidad</li> <li>✓ Portabilidad</li> <li>✓ Usabilidad</li> </ul>

Fig: 9 Atributos de calidad Modelo Dromey

El proceso de construcción de modelos de calidad propuesto por Dromey consta de 5 pasos, basados en las propiedades mencionadas. Los pasos del marco de referencia propuesto son:

1. Especificación de los atributos de calidad de alto nivel (por ejemplo, confiabilidad, mantenibilidad)
2. Determinación de los distintos componentes del producto a un apropiado nivel de detalle (por ejemplo, paquetes, subrutinas, declaraciones)
3. Para cada componente, determinación y categorización de sus implicaciones más importantes de calidad
4. Proposición de enlaces que relacionan las propiedades implícitas a los atributos de calidad, o, alternativamente, uso de enlaces de las cuatro categorías de atributos propuestas
5. Iteración sobre los pasos anteriores, utilizando un proceso de evaluación y refinamiento.

Para ilustrar sus planteamientos, Dromey demuestra el uso de su procedimiento para la construcción de un modelo de calidad de implementación, un modelo de calidad de requerimientos, y un modelo de calidad de diseño. (20)

### **2.2.5. Norma ISO 9126**

La norma ISO, ha tenido gran impacto a nivel mundial. A partir del trabajo realizado en los modelos anteriores, en 1993 la ISO propuso su propio modelo de calidad, la norma **ISO 9126**, “*de calidad de producto software*”, que recoge características específicas de calidad, que contribuyen a satisfacer las necesidades de usuario, con las que se puede medir la calidad y la seguridad de cualquier producto. La norma ISO 9126, que distingue entre la calidad en uso de un producto de software (*efectividad*,

productividad, seguridad y satisfacción con la que un usuario específico interactúa con el producto en un ambiente y contexto específico) y su calidad interna (características objetivas, medibles sobre el software: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad). De los atributos que expone la norma ISO 9126, la confiabilidad es uno de los más importantes, siendo un elemento clave para medir si el producto que realizamos es fiable o no. La confiabilidad es un mecanismo de seguridad, es con lo que el usuario se siente conforme y satisfecho con su producto.

Característica	Subcaracterística
Funcionalidad	✓ Adecuación ✓ Exactitud ✓ Interoperabilidad ✓ Seguridad
Confiabilidad	✓ Madurez ✓ Tolerancia a fallas ✓ Recuperabilidad
Usabilidad	✓ Entendibilidad ✓ Capacidad de aprendizaje ✓ Operabilidad
Eficiencia	✓ Comportamiento en tiempo ✓ Comportamiento de recursos
Mantenibilidad	✓ Analizabilidad ✓ Modificabilidad ✓ Estabilidad ✓ Capacidad de pruebas
Portabilidad	✓ Adaptabilidad ✓ Instalabilidad ✓ Reemplazabilidad

Fig: 10 Modelo de ISO 9126

Este estándar fue normado por la Oficina Nacional de Normalización de Cuba y admitido como una norma cubana, por lo que constituirá, una de las principales fuentes en las que se basará esta investigación, especificando sus elementos distintivos.

### 2.2.6. Dimensiones de Confiabilidad ISO 9126

La Confiabilidad está compuesta por cuatro sub-características esenciales en la que se describen atributos que la evalúan, como se muestra a continuación:

**Madurez:** se entiende como la capacidad del software para evitar fallas ocasionadas por defectos en el programa, los atributos que se pueden evaluar en relación a esta sub-características son:

- **Presencia de errores:** el objetivo es evaluar la existencia de errores en el producto. A menor cantidad de errores se tiene mayor madurez en el producto.

- **Prevención de fallas por errores internos:** el objetivo es evaluar la capacidad de prevenir fallas ocasionadas por errores internos en el programa. Para este atributo se deben tener en cuenta las posibles acciones preventivas que puede realizar el programa.
- **Evolución:** el objetivo es evaluar la transformación gradual que ha presentado el producto y está comprendida por un conjunto de modificaciones contenidas en diferentes versiones liberadas en el transcurso del tiempo.

**Recuperabilidad:** se entiende como la capacidad del software para restablecer un determinado nivel de desempeño en caso de una falla y recuperar los datos directamente afectados, de una manera consistente, en proporción al tiempo, recursos y esfuerzo necesario para garantizar que el sistema regrese a un estado estable, los atributos a evaluar en esta relación son:

- **Capacidad de retorno a un estado estable:** se relaciona con la capacidad del producto para regresar a un estado estable después de la ocurrencia de una falla, externa o interna, a la cual no se la haya podido dar un manejo por medio de acciones preventivas.
- **Recuperabilidad:** evalúa la capacidad del software para recuperarse de los daños producidos por una falla durante un período de tiempo determinado.

**Tolerancia a fallos:** se entiende como la capacidad del software para conservar un determinado nivel de desempeño frente a defectos en el software, los atributos que se pueden evaluar son:

- **Continuidad en la operación:** Evalúa la capacidad del aplicativo para seguir ejecutando sus funcionalidades, aun en casos de fallas internas o externas.

**Conformidad con la confiabilidad:** la capacidad del producto software para adherirse a estándares, convenciones o regulaciones relacionadas con la confiabilidad. (21)

### 2.2.7. Análisis de los modelos

Tomando en cuenta los rasgos de los modelos estudiados, se puede apreciar que cada uno de los modelos en singular, tiene formas similares de definir confiabilidad pero con atributos distintos, los cuales hacen de los modelos las diferencias. Es de destacar que cada una de estas características influye en la evolución de la calidad de un producto enteramente confiable. Es importante señalar que aunque parezcan similares los modelos de McCall y Boehm, las especificaciones de sus características las distinguen de formas diferentes.

McCall organiza los factores en tres ejes o puntos de vista desde los cuales el usuario puede contemplar la calidad del producto (*Operación del producto, Revisión del producto, Transición del producto*), además de basarse en 11 factores de calidad, donde la fiabilidad se incluye dentro del eje de *Operación del producto*, adquiriendo así un nivel profundo de cada uno de los atributos que componen el modelo.

Por su parte Boehm presenta una jerarquía de características primarias cada una de las cuales contribuye a la calidad de forma global, incluyendo características de desempeño de hardware. Afirma además que el software debe hacer lo que *el usuario quiere que haga, utilizar recursos de la computadora correcta y eficientemente, que sea fácil de aprender y usar para los usuarios*.

Por su parte el modelo propuesto por Dromey sintetiza elementos dirigidos expresamente a determinar la calidad de los productos de software. Sirviendo de base, con los 5 pasos que propone para la definición de su modelo, para la formulación del modelo de pruebas propuesto en esta investigación.

A partir del trabajo realizado en los modelos anteriores, la ISO propuso su propio modelo de calidad, la norma ISO 9126, que distingue entre la calidad en uso de un producto de software, (efectividad, productividad, seguridad y satisfacción con la que un usuario específico interacciona con el producto contexto específico) y su calidad interna (características objetivas, medibles sobre el software: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad). Además, la ISO postula que una mayor calidad interna/externa del producto de software incidirá de manera positiva en la percepción que el usuario tiene acerca de la calidad de la aplicación.

Estos modelos tienen sus límites, es difícil que las características y sub-características sean siempre perfectamente independientes, falta una asociación explícita entre los modelos y el proceso de software, y es que todos se centran en *¿cómo realizar software de calidad?*

### 2.3. DATEC

El centro de desarrollo de software **DAteTEC**nology, surge de la necesidad de fortalecer el desarrollo de tecnologías de bases de datos en nuestra universidad y en el país, mejorar los niveles en la formación de bases de datos en el pregrado y el postgrado, además de propiciar soluciones más robustas en nuestro entorno productivo en los temas referentes a las tecnologías de bases de datos. En fin, necesidades que tributan a encaminarse hacia una soberanía tecnológica.

Cuenta con tres líneas temáticas de desarrollo, que son: *Tecnologías de bases de datos*, *Almacenes de datos* y *Soluciones de Inteligencia de Negocios e Integración de Soluciones*, *Herramientas de Análisis de datos y toma de decisiones*. En el centro los principios de producción están sustentados en un modelo de producción basado en:

- Componentes y herramientas de cuarta generación; soluciones basadas en arquitecturas orientadas a servicios, alta reutilización y diseño arquitectónicos integrados.
- Modelo de producción basado en los principios de la calidad total, enfoque centrado en el usuario y un fuerte ejercicio de marketing.
- Modelo centrado en la mejora continua bajo los principios de la arquitectura de la empresa.

### Líneas de investigación y grupos

- Tecnologías de bases de datos
- Gestión de proyectos informáticos

### Líneas de productos de software de DATEC

- Almacenes de datos e inteligencia de negocios
- Desarrollo de Tecnologías de bases de datos (PostgreSQL)
- Integración de soluciones y desarrollo de herramientas para el análisis y el procesamiento de datos. (22)

#### 2.3.1. Tipos de productos de DATEC

Actualmente la producción está dirigida a productos vinculados a la gestión de datos, como son, los *Data Warehouse*, *Operational Data Store (ODS)*, *Inteligencia de Negocio*, entre otros.

Los **Data Warehouse** son un repositorio de datos de muy fácil acceso, alimentado de numerosas fuentes, transformadas en grupos de información sobre temas específicos de negocios, para permitir nuevas consultas, análisis, reportes y decisiones.

Entre los objetivos fundamentales que presenta un Data Warehouse se encuentran:

- Hacer que la información de la organización sea accesible, los contenidos entendibles y navegables, y el acceso a ellos son caracterizados por el rápido desempeño.
- Hacer que la información de la organización sea consistente
- La información es adaptable y elástica, por lo que está diseñado para cambios continuos.

- Tener los datos correctos para soportar la toma de decisiones.

Los procesos básicos del Data Warehouse (ETL)

- Extracción: este es el primer paso de obtener la información hacia el ambiente del Data Warehouse.
- Transformación: una vez que la información es extraída hacia el área de tráfico de datos, hay posibles pasos de transformación como, limpieza de la información, combinar fuentes de datos entre otras.
- Carga: al final del proceso de transformación, los datos están en forma para ser cargados. (23)

El **Operational Data Store (ODS)** es un sistema que recopila la información existente en los sistemas operacionales y la ofrece a los sistemas informacionales. Para ello existe una plataforma común donde conviven tanto la información procedente de los sistemas operacionales, como la información integrada en sistemas informacionales. Esta plataforma contiene herramientas/procedimientos de extracción, transformación, carga y consulta de la información común para todas las aplicaciones que se encuentran en el entorno del ODS.

Con ODS se solucionan varios de los problemas inherentes de los Data Marts (un **Datamart** es una base de datos departamental, especializada en el almacenamiento de los datos de un área de negocio específica) tradicionales:

- La extracción de información no afecta a la ejecución de trabajos en los Sistemas Operacionales.
- La información se replica en un sistema paralelo, y no en los propios Sistemas Operacionales.
- Se evita la sobrecarga de los equipos donde se albergan los Sistemas Operacionales con acciones externas al propio sistema propietario.

Por lo tanto, aporta las siguientes ventajas:

- Ahorro de costes por la reutilización de recursos.
- Disponibilidad de un dato único, evitando múltiples copias del mismo dato en sistemas dispersos.
- Posibilidad de cruzar la información de diferentes aplicaciones. (24)

La **Inteligencia de Negocios** abarca metodologías, técnicas, herramientas y procesos de toma de decisiones, basada en tecnología de punta. Con la inteligencia de negocios se analizan los bienes o datos acumulados en la institución y extraer conocimientos de ellos. La necesidad principal que cubre la Inteligencia de Negocios es el habilitar a los usuarios para poder acceder a la información, sustraer

información a través de consultas sencillas, de las cuales la mayoría son ejecutadas en el lenguaje SQL , para después dar formato y presentación a la información en la cual van a estar basando su toma de decisiones. (25)

### 2.4. Análisis de los riesgos asociados a la confiabilidad en DATEC

Al analizar la confiabilidad de los sistemas desarrollados en DATEC, a través de las diferentes entrevistas realizadas a especialistas se arriba a la formulación dos interrogantes que son necesarias para evaluar el producto, *¿Con cuánta exactitud la tarea a automatizar satisface la concepción inicial? ¿Con qué fidelidad responde el resultado al cliente?* Aunque casi todos los software tengan errores, la mayoría de ellos nunca son revelados bajo circunstancias normales, es ahí donde la *Confiabilidad* desempeña un papel importante, pues resulta un aspecto esencial en la elaboración del software, donde se combinan los elementos que apoyarán un buen funcionamiento del sistema y evaluarán cuán útil resulta el producto elaborado.

Las entrevistas realizadas a especialistas del centro, revelaron que, actualmente en DATEC, existen varios problemas que afectan en gran medida la confiabilidad de los productos que desarrollan, por ejemplo, no se llevan a cabo las políticas pertinentes para asegurar que el cliente se lleve un producto listo y sin posibilidad de fallas. Al realizar un análisis profundo de la situación existente de forma general, se llegó a la conclusión, que no se cuenta con un modelo específico para llevar a cabo evaluaciones a través de los atributos de confiabilidad para la elaboración del software. Independientemente de los resultados, se tuvieron en cuenta los siguientes aspectos referentes en cuanto a:

- Capacidad del software para evitar fallas ocasionadas por defectos en el programa.
- Capacidad del software para restablecer un determinado nivel en caso de falla y recuperar los datos perdidos.
- Capacidad del software para conservar un determinado nivel en caso de ocurrir una falla, ya sea interna o externa en el sistema.
- En qué nivel de precisión se encuentran los datos, cálculos y los resultados con los que se cuenta.  
(ver anexo # 1)

A continuación se muestra un resumen con las características que, en resumen, afectan a las tres líneas de desarrollo que integran el centro.

### **Madurez (*Presencia de errores, Prevención de fallas, Evolución*)**

En el centro de desarrollo DATEC, existen elementos críticos asociados a la calidad con la que se gestionan los datos, pues presentan inconsistencia en los mismos, esto trae consigo incoherencias en las salidas de los resultados. No siempre se cuenta con toda la documentación para manejar otros tipos de formato de datos, lo que conlleva a la pérdida de tiempo, o sea, se necesitarían horas para poder estudiar y saber cómo funcionan estos datos y poderlos transformar. La información de los datos no siempre está disponible, lo que significa que los usuarios no pueden trabajar con datos reales, lo que implicaría que las pruebas no sean un tanto certeras. No cuentan con métodos de verificación en cuanto a (estándares de codificación y estándares para la documentación) para controlar las fallas en el software. No existe un método que tenga la capacidad de prevenir defectos por errores internos y externos en el sistema, sólo se realizan pruebas para prevenir fallas cuando se finaliza el producto y no siempre realizan las pruebas necesarias para que el sistema sea probado enteramente, lo que implica atraso en la entrega del producto desarrollado, pues una vez que este producto esté terminado tendrá que ser evaluado por especialistas vinculados con el tema.

### **Recuperabilidad (*Capacidad de retorno a un estado estable, Recuperabilidad*)**

Uno de los problemas de mayor peso que presentan es que, identifican los posibles casos de pruebas, los documentan, pero no los llevan a cabo, los casos de prueba son muy importantes, pues representan un registro de defectos y dificultades detectados en el software. Sólo quedan plasmadas algunas políticas y rutinas, porque las pruebas que se hacen nunca son en función de respaldo de forma intencional, o sea, no se tienen en cuenta las pruebas de estrés, las cuales deciden hasta dónde puede soportar el programa determinadas condiciones extremas. Estas pruebas forman parte de las pruebas de sistema. El sistema automáticamente no tiene forma de que en caso de falla pueda restablecerse y volver a la actividad que estaba ejecutando, de lo contrario, lo más seguro es que se pierda parte del código y que el desarrollador tenga que empezar de cero la función que estaba realizando. No existe un mecanismo para saber si el backup está funcionando, saber si se mantiene disponible, para en caso de ocurrir algún error que afecte completamente al sistema dicho backup pueda mantener las salvadas activas.

### **Tolerancia ante fallas (*Continuidad en la operación*)**

Un detalle importante y de notar dentro del centro, es que, cuando algún componente del sistema falla, puede afectarse el correcto funcionamiento de algunos componentes, pero a la vez dejar otros componentes sin afectaciones. Sin embargo, no se tienen controles de verificaciones, lo cual influye en la

continuidad de la operación, lo que pudiera impedir las recuperaciones automáticamente en casos de fallos sin afectar el rendimiento, lo que se explica, que si existe un error en el sistema debería seguir operando de forma aceptable mientras se hacen las reparaciones.

Otro aspecto importante son los servidores, tema poco tratado pues no se está al tanto de su rendimiento y de las funciones que estos deberían estar realizando. Si el servidor se detiene prematuramente aun cuando instantes antes estaba funcionando de manera correcta, puede perderse alguna información de la que se maneja en el centro, pues una vez que estos errores ocurren, se hace difícil volver a recuperar los datos.

### ***Exactitud y Precisión***

Actualmente los datos con que prueban el software son datos generados por ellos, no existe precisión en los cálculos y emisiones de resultados, por lo que no se tiene control de los datos que toman como entrada, no se tiene constancia para saber si ese dato ya fue probado. Por otra parte esta operación se realiza manualmente.

### ***Pruebas***

En estos momentos dentro del centro, cada proyecto tiene su Plan de Pruebas, los cuales deben realizarse en función del Modelo V, que establece los procedimientos utilizados para probar si la aplicación cumple las especificaciones que ya deben haberse creado en la fase de diseño. El Modelo V demuestra las relaciones entre cada fase del ciclo vital del desarrollo y su fase asociada de prueba.

Como política está establecido usar el Modelo V para la gestión de la calidad, pero sólo se concreta el hecho de diseñar los tipos de pruebas, porque nadie del equipo de desarrollo lo aplica. Esto se debe a que no se documentan los casos de pruebas, pues en ellos se registran la mayoría de los defectos que se obtienen del producto. En este modelo se despliega un método bien estructurado, en el cual cada fase se puede poner en ejecución por la documentación detallada de la fase anterior. Las actividades de cómo diseñar la prueba se realizan en el principio del proyecto, antes de la codificación y por lo tanto se ahorraría una cantidad enorme del tiempo del proyecto.

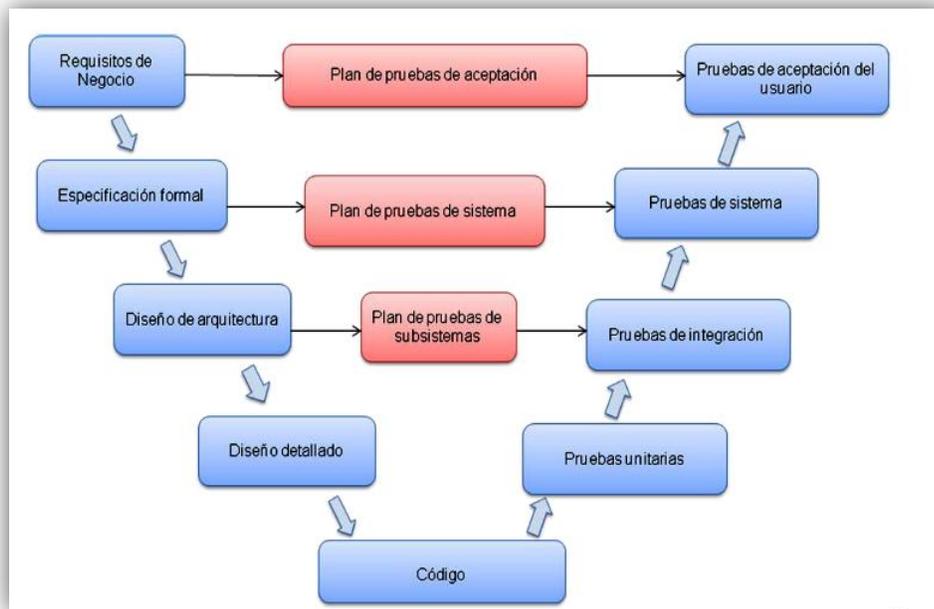


Fig: 11 Modelo V

### 2.5. Función de las pruebas en el Modelo V

Las pruebas deben ocurrir durante todo el ciclo de vida del software, a través del Modelo V utilizado en el centro, brindando la posibilidad de probar las funcionalidades de los prototipos de interfaz, la estabilidad del software, la cobertura y el rendimiento de la arquitectura, asegurando de ésta forma la calidad del producto final.

Dentro de la actividad *Requisitos del Negocio*, las **pruebas de aceptación** aseguran que el cliente determine si el sistema cumple con lo deseado y si está conforme con el mismo. Son básicamente pruebas funcionales sobre el sistema completo, y buscan las especificaciones de requisitos y del manual de usuarios. Estas pruebas no deben realizarse durante el desarrollo, pues sería impresentable al cliente, sino que se realizan sobre el producto terminado e integrado.

Para la realización de la actividad *Diseño de la Arquitectura*, **las pruebas de integración** complementan de forma general lo que se necesita probar, pues el objetivo fundamental es tomar los módulos probados en las pruebas de unidad y construir una estructura para el programa, que esté acorde con lo especificado en el diseño. Prevalciendo en este tipo de prueba las técnicas de caja negra, que son las que se centran en lo que se espera del módulo, intentando encontrar casos en que el módulo no se atiene a su

especificación, además están especialmente indicadas para aquellos módulos que van a ser interfaz con el usuario.

**Las pruebas unitarias** están dirigidas a los diferentes módulos que se desarrollan en cada proyecto. Para lo que se usa la descripción del *Diseño Detallado* como guía, se prueban los caminos de control más importantes, con el fin de descubrir errores dentro del ámbito del módulo. Haciendo un uso intensivo de las técnicas de prueba de caja blanca, que son las que están muy pendientes del código, se realizan sobre las funciones internas del módulo, entre las técnicas usadas se encuentran; la cobertura de caminos (pruebas que hagan que se recorran todos los posibles caminos de ejecución), pruebas sobre las expresiones lógico-aritméticas, pruebas de camino de datos (definición-uso de variables), comprobación de bucles (se verifican los bucles para 0,1 y n iteraciones, y luego para las iteraciones máximas, máximas menos uno y más uno).

Después de hacer un análisis se ha llegado a la conclusión que en DATEC no existe una guía de pasos para llevar a cabo un modelo de pruebas que les permita evaluar los atributos de confiabilidad necesarios para la seguridad de cualquier producto que se desarrolle. A pesar de los esfuerzos que hace el centro por mantener un alto nivel de calidad, se evidenció la necesidad de desarrollar una herramienta científicamente probada que los guiara en el proceso de pruebas, aun cuando la entrevista estuvo orientada a todos los atributos propuestos por los modelos estudiados, se evidenció que los problemas radicaban en los descritos anteriormente.

### 2.6. Solución Propuesta

Para el diseño de un modelo de prueba que permita evaluar la confiabilidad de los tipos de productos que en DATEC se desarrollan, se ha tomado la decisión de elaborar una guía de pruebas a seguir, que describa los pasos a tener en cuenta, a la hora de probar el producto; con el fin de que sea evaluado a través de las diferentes características de confiabilidad, tomadas de los distintos modelos analizados, como son: *Madurez, Tolerancia a Fallas, Recuperabilidad, Exactitud y Precisión*, definiendo éstas, porque son las dimensiones más relacionadas con las necesidades que presentan los productos en dicho centro.

#### 2.6.1. Objetivos del modelo

- Proponer un conjunto de pruebas que apoyen de manera consistente el Modelo V.
- Definir los pasos a tener en cuenta para realizar los diferentes tipos pruebas de confiabilidad, a los productos de DATEC.

- Definir el impacto de cada una de las pruebas realizadas en cuanto a la confiabilidad.

Entre los arquetipos que se persiguieron para desarrollar las pruebas a los productos, se encuentran los siguientes:

- El correcto funcionamiento de los componentes del sistema y detección de errores, así como el de las interfaces de los distintos subsistemas que lo componen.
- La validación de cada uno de los datos, además de la seguridad de la Base de Datos.

Para darle solución a los objetivos perseguidos se definieron primeramente los tipos de pruebas por cada uno de los módulos que rigen el Modelo V y el objetivo de cada una de ellas.

Para llevar a cabo este plan se necesita previamente:

1. Pensar en la confiabilidad: las aplicaciones para ser confiables han de ser compatibles con operaciones confiables, y por tanto necesitan también, procesos de implementación confiables. Observar el modo en que se proporciona el servicio y se busca posibles problemas, donde las alternativas de diseño o de procedimiento permitan reducir las causas de error.
2. Invertir en personal: El personal de operaciones y los programadores deben conocer a fondo las prácticas de administración del ciclo de vida y de la arquitectura, poniendo especial atención en la prevención de los errores más comunes. Proporcionar educación sobre los procedimientos de la compañía, las herramientas de programación, las tecnologías de aplicación y los conceptos de confiabilidad.
3. Eliminar los puntos con errores desde el diseño de aplicaciones, pues un sistema confiable es más fácil de mejorar que un sistema no confiable (con eventos de error ocultos y distribuidos por todo el programa), que es muy costoso cambiar.
4. Utilizar un sistema operativo consistente (Proporcionar supervisión de confiabilidad continua)
5. Todas las aplicaciones críticas para una misión deben proporcionar datos de supervisión. El análisis de los datos recopilados desempeña un papel importante en la observación del estado, de los problemas actuales y de las tendencias de largo alcance de las aplicaciones críticas.
6. Invertir en procesos de diseño de software de calidad:
  - Uso de revisiones de código y estándares de codificación
  - Desarrollo de procedimientos de recuperación
  - Uso de procedimientos de control de cambios probados

### 2.7. Pruebas de Confiabilidad

La comprobación de la confiabilidad consiste en probar una aplicación para descubrir y eliminar errores antes de que se implemente el sistema. Puesto que existen infinidad de combinaciones distintas de recorridos alternativos a lo largo de una aplicación. Para ello es importante destacar que para la aplicación de este modelo de confiabilidad, es necesario, plasmar los casos de pruebas, para recoger en ellos los defectos que se obtuvieron en cada una de las revisiones realizadas. Además, se debe especificar en cada uno de ellos, cuantas no conformidades se tienen por cada uno de los indicadores que se explican más adelante, con lo que se infiere que cada una de las pruebas propuestas va tener como entrada un caso de prueba con las especificaciones bien detalladas de los indicadores.

A cada una de las pruebas propuestas, la integran un conjunto de indicadores que van a responder a las dimensiones de confiabilidad seleccionadas para el modelo, (Madurez, Tolerancia ante falla, Recuperabilidad, Precisión y Exactitud).

El usuario debe especificar en cada prueba el nivel de cumplimiento del indicador, a este último, se le asociará un valor en dependencia del grado de importancia que el usuario le asigne, la suma de todos los valores debe ser 1, luego se multiplica ese valor asignado por el nivel de cumplimiento seleccionado, según las no conformidades que contenga el sistema, una vez hecho esto por cada uno de los indicadores, se hace una sumatoria de todos los valores obtenidos y se ubica ese número en la tabla de *nivel de confiabilidad* en la que se especifica por rango si la prueba evaluada tiene un nivel de confiabilidad *Alto, Medio o Bajo*.

$$C_{pb} = \sum G_j P_i$$

Donde  $C_{pb}$  = Confiabilidad de la Prueba

Donde  $G_j$ : Grado de indicador o valor esperado en la prueba (j) y  $P_i$ : **Peso o nivel de cumplimiento** del indicador (i).

1	Muy Insatisfactorio	a partir de 5 no conformidades
2	Insatisfactorio	a partir de 4 no conformidades
3	Aceptable	2-3 no conformidades
4	Satisfactorio	1 no conformidad
5	Muy satisfactorio	0 no conformidades

Tabla 1: Nivel de cumplimiento

Terminado el proceso para cada una de las pruebas, se multiplica el valor obtenido en las pruebas por 0.2, (que va a ser el peso que se le da a cada una de las dimensiones, por tener todas la misma importancia en el software), y luego se suman los resultados, el valor que se obtenga se ubica en la tabla de rangos, la misma determina si el nivel de confiabilidad del producto de dicha prueba es *Alto, Medio o Bajo*.

**$C_p = 0.2 \sum \text{Confiabilidad Prueba (j)}$**

Donde  $C_p$  = Confiabilidad del producto

Rango Confiabilidad	Nivel de Confiabilidad	% de Confiabilidad
(4 - 5)	Alta	(80 - 100)%
(3 - 3.9)	Medio	(60 - 79) %
(0 - 2.9)	Bajo	(0-59)%

Tabla 2: Nivel de Confiabilidad

El comportamiento de las dimensiones varía en dependencia de la importancia que adquiera en el producto.

**$C_p = \sum R \text{ Confiabilidad Prueba (j)}$**

R = Valor de importancia de la dimensión, según el tipo de producto.

Por ejemplo:

<b>Producto de almacenes de datos</b>	<b>R</b>
1. Pruebas de Componentes	0.2
<b>2. Pruebas de Integración</b>	<b>0.4</b>
3. Pruebas de Recuperabilidad	0.2
4. Pruebas Estructurales	0.05
5. Pruebas de Sistema	0.15

**2.7.1. Pruebas de Componentes**

A través de las pruebas de estrés de los componentes todos los servicios los y componentes que conforman el sistema se aíslan, se infieren los métodos de navegación, funcionamiento, interfaz, servicios y componentes, y se crea un cliente de prueba que llame a dichos métodos.

Para aquellos métodos que tienen acceso al servidor de base de datos o a cualquier otro componente, puede crear un cliente que proporcione datos simulados en el formato previsto. El equipo de prueba

inserta datos simulados una y otra vez mientras observa los resultados. La idea es forzar cada componente de forma aislada más de lo que la aplicación podría experimentar en condiciones normales. Los indicadores de las pruebas de componentes responden a la dimensión de confiabilidad *Madurez*.

Indicadores en Pruebas de Componentes

- **Navegación** (facilidad de movilidad para el usuario mediante los controles del sistema sin necesidad de uso de los botones del navegador)
- **Funcionamiento** (respuesta efectiva de cada una de las partes que componen el sistema en conjunto y funcionando como un todo)
- **Interfaz** (nivel de madurez en que se muestra el contenido al usuario, ordenado por categorías, por procesos, etc.)
- **Componente** (grado de respuesta de cada componente de forma separada)
- **Mecanismo de comunicación** (eventos, mensajes o pase de parámetros)

### 2.7.2. Pruebas de Integración

Las pruebas de integración están íntimamente relacionadas con las interacciones con otras estructuras de datos, procesos y servicios tanto de los componentes internos como de otros servicios externos de la aplicación. Las pruebas de integración comienzan con una comprobación básica del funcionamiento. Es necesario que conozca los recorridos codificados y las situaciones a las que se enfrentan los usuarios, que comprenda lo que intentan hacer estos y que identifique todas las maneras en las que el usuario se mueve por la aplicación. También es importante realizar los casos de pruebas para verificar que los componentes funcionan entre sí de la forma apropiada después de haber sido integrado en una construcción:

1. Comparar los resultados obtenidos con los esperados e investigar los que no coinciden.
2. Informar defectos al responsable de la prueba.
3. Utilizar las herramientas de apoyo.

Los indicadores de las pruebas de integración responden a la dimensión de confiabilidad *Exactitud* y *Precisión*.

Indicadores en Pruebas de Integración.

- **Interoperabilidad**(característica de los sistemas que les permite su interconexión y funcionamiento conjunto de manera compatible)

- **Veracidad**(se refiere a la precisión de los componentes)
- **Tiempo de respuestas**(basado en una vez integrados los componentes, capacidad que tiene el sistema de brindar una respuesta esperada)
- **Evolución de versiones**(control de cambios de las actualizaciones)

### 2.7.3. Pruebas de Recuperabilidad

Las pruebas de recuperación se hacen en función verificar la capacidad del sistema para recuperarse frente a errores. Muchos sistemas deben recuperarse en un tiempo previamente especificado o incluso en algunos casos deben ser totalmente tolerante a las fallas, o sea, en un proceso no deben hacer que el sistema cese completamente en su funcionamiento. Los indicadores de las pruebas de recuperabilidad responden a la dimensión de confiabilidad *Recuperabilidad*.

Indicadores en Pruebas de Recuperabilidad

- **Tolerancia ante falla**(capacidad del sistema a mantener un nivel de desempeño en caso de posible falla)
- **Mecanismos de protección**(refiere a los métodos, las herramientas, que posee el software para su protección)
- **Capacidad de retorno a un estado estable**(infiere a la comprobación de que en caso de falla, el sistema vuelva a su operación)
- **Disponibilidad de datos**(control de acceso a datos)

### 2.7.4. Pruebas Estructurales

Son también conocidas como "pruebas de caja blanca" o "pruebas basadas en código", donde se enfocan en probar cada una de las estructuras de código, para que su comportamiento sea el esperado. Son las pruebas donde se conoce la estructura interna del componente a probar, y se efectúa una prueba sobre dicha estructura. Se necesita para esta prueba tener en cuenta los posibles casos de pruebas, que verifiquen la estructura interna del producto.

1. Realizar casos de pruebas que ejerciten al menos una vez todos los caminos independientes de cada funcionalidad.
2. Ejecutar casos de prueba que ejerciten las decisiones lógicas.
3. Probar las estructuras de datos internas para asegurar su validez

Los indicadores de las pruebas de estructurales responden a la dimensión de confiabilidad *Madurez*.

Indicadores en Pruebas de Estructurales

- **Existencia de errores**(infiere a la revisión en todos los caminos de ejecución)
- **Prevención de errores por fallas**(revisión de mecanismos o métodos utilizados para la prevención de errores )
- **Evolución de versiones**(control de cambios de las actualizaciones)
- **Errores en estructuras de datos o acceso a base de datos externas**(infiere a la revisión del acceso a la base de datos)

### 2.7.5. Pruebas de Sistema

Verifican que cada elemento encaje de forma adecuada y que se alcance el rendimiento del sistema total. Además prueban que el sistema funcione globalmente de forma correcta.

- Probar que el sistema funciona correctamente como un todo.
- Probar con diferentes configuraciones (SW y HW), diferentes cargas al sistema, diferentes números de actores y distintos tamaños de BD.

Los indicadores de las pruebas de sistema responden a la dimensión de confiabilidad *Tolerancia ante fallas*.

Indicadores en Pruebas de Sistema

- **Respuestas ante la sobrecarga de datos**(se refiere a cargar el sistema con grandes cantidades de datos, para saber cuan soportable y resistible es el software elaborado)
- **Continuidad de las operaciones**(infiere a que si falla el software, este es capaz de continuar con sus operaciones, una vez que se restablezca)
- **Disponibilidad de recursos**(revisión de todos los recursos que debe usar el software, por ejemplo, la base de datos, la red, etc.)
- **Rendimiento**(respuestas del sistema)

Las **pruebas de sistema** también forman parte del Modelo V, pues se llevan a cabo dentro de la actividad *Especificación Formal*, con estas pruebas se pueden evaluar todas las dimensiones de confiabilidad, puesto que influyen significativamente en el desempeño de un software, ellas verifican que cada elemento encaje de forma adecuada y que se alcance el rendimiento del sistema total.

Además están compuestas por diferentes tipos de pruebas que son las que se van a especificar en la solución propuesta, y las cuales van proporcionar probar cada uno de las dimensiones de confiabilidad, que quizás hoy no constituyen un problema para la calidad y la confiabilidad del producto dentro del centro, pero que pudieran constituir una dificultad en algún momento, las dimensiones que se pueden evaluar con los tipos de pruebas que se proponen son: *completitud, modularidad, consistencia y simplicidad*, se hace necesario definir cuáles son las funciones de estas pruebas. Se define además que cada una de estas pruebas va a tener como entrada los *casos de pruebas*. A continuación se explican cada de las pruebas propuestas.

1. **Las Pruebas Funcionales:** van a tener como objetivo principal, verificar la función del sistema al fijar la atención en la validación de las funciones, métodos, servicios y casos de usos. Validan que el sistema cumpla con los requisitos funcionales y no funcionales especificados en el diseño de la solución. Una prueba funcional suele comenzar con la inicialización del objeto del navegador para pruebas. Este objeto permite realizar una petición a una acción de la aplicación y permite verificar que algunos elementos están presentes en la respuesta. Las funciones son probadas ingresando las entradas y examinando las salidas.
2. **Las Pruebas de Rendimiento:** se ejecutan tanto para determinar cómo responde un sistema ante una cierta carga, como para validar otros atributos relacionados con la calidad, como pueden ser la escalabilidad o el uso de recursos entre otros, prueban el rendimiento del software en tiempo de ejecución. Validan y evalúan la aceptabilidad de un elemento de un sistema sobre diferentes cargas de trabajo, mientras el sistema permanece constante. Generalmente se incluye la simulación de cargas de trabajo promedio o pico de desborde que puedan ocurrir dentro de la tolerancia operacional normal.
3. **Las Pruebas de Seguridad:** verifican los mecanismos de protección, pretenden evaluar la confidencialidad, integridad y disponibilidad de los datos, partiendo de identificar amenazas y riesgos desde el uso o interfaz de usuario final. Con estas pruebas se puede medir y cuantificar los riesgos a los cuales se ven expuestos las aplicaciones, tanto en la infraestructura interna como externa. También pueden simular un ataque informático desde cualquier perspectiva (Internet, red interna, redes asociadas, acceso remoto, etc.) para establecer qué tan posible sería para un atacante, comprometer la seguridad de la información y validar la posible ocurrencia de un fraude.
4. **Las Pruebas de Disponibilidad y Red:** verifican el comportamiento de la aplicación, cambiando la infraestructura de red al aplicar diferentes configuraciones y retardos. Valida que no se reduzca la

disponibilidad de los sistemas dentro de la solución. Probar la disponibilidad consiste además, en ejecutar una aplicación durante un período de tiempo previsto, recopilando los eventos de error y los tiempos de reparación y comparando el porcentaje de disponibilidad con el compromiso original de nivel de servicio. La comprobación de la disponibilidad está relacionada sobre todo con la medición y la reducción del tiempo real de reparación.

5. **Las Pruebas de Estrés:** se refieren a cargas extremas, memoria insuficiente, no disponibilidad de servicios y hardware o recursos compartidos limitados. Estas pruebas permiten comprender mejor cómo y qué áreas del sistema colapsarán, de este modo es posible planificar contingencias, actualizar el mantenimiento, planear y asignar recursos de antemano.
6. **Las Pruebas de Usabilidad:** verifican la estética, la consistencia de la interfaz de usuario y documentación. Dentro de las métricas para evaluar usabilidad podemos encontrar, *exactitud*, que responde al número de errores cometidos por los sujetos de prueba y si estos fueron recuperables o no al usar los datos o procedimientos adecuados, *tiempo* requerido para concluir la actividad, *respuesta emocional* como se siente el usuario al terminar la tarea.
7. **Las Pruebas de Fiabilidad:** verifican la probabilidad de que el sistema funcione o desarrolle una determinada función, bajo condiciones prefijadas y durante un período de tiempo determinado, en fin la confiabilidad consiste en buscar los defectos y reducir el número de errores.

### Conclusiones Parciales

En la solución planteada del capítulo, quedó evidenciado paso a paso como debe realizarse el nuevo modelo, garantizando desde un inicio que cada funcionalidad de los sistemas sean revisadas inicialmente desde el propio proyecto, y así a la hora de evaluar el producto en el Laboratorio Central de Calidad sea más ágil y eficiente el proceso.

## **Capítulo 3: Validación de la Solución**

### **3. Validación del modelo de confiabilidad**

Para aceptar y validar el Modelo de Pruebas de Confiabilidad que se propone, se realizó una selección de expertos que emitieron su criterio, el mismo estuvo compuesto por especialistas en el tema de calidad y pruebas de software de la Universidad de las Ciencias Informáticas. El proceso de validación se llevó a cabo por el Método Delphi, que es una técnica de investigación social que tiene como objetivo, obtener de forma confiable un consenso de opiniones y valoraciones sobre un tema en específico, en este caso es utilizado para validar la calidad del Modelo de Pruebas que se propone utilizar en el centro DATEC.

#### **3.1. Delphi como método de validación de experto**

El método Delphi, considerado como uno de los métodos subjetivos de pronósticos más confiables, constituye un procedimiento para confeccionar un cuadro de la evolución de situaciones complejas, a través de la elaboración estadística de las opiniones de expertos en el tema tratado. El mismo permite rebasar el marco de las condicionantes actuales más señaladas de un fenómeno y alcanzar una imagen integral y más amplia de su posible evolución, reflejando las valoraciones individuales de los expertos, las cuales podrán estar fundamentadas, tanto en un análisis estrictamente lógico como en su experiencia intuitiva.

En la actualidad, el método Delphi consiste en un proceso iterativo de rondas donde se le realizan encuestas a cada experto de forma individual evitando la interacción. La calidad de los resultados depende, sobre todo, del cuidado que se ponga en la elaboración del cuestionario y en la elección de los expertos consultados. Siempre se comenzaría este proceso enviando un modelo a los posibles expertos con una explicación breve sobre los objetivos del trabajo y los resultados que se desean obtener. Para la aplicación práctica del método es necesario considerar dos cuestiones fundamentales: (26)

- ✓ La selección de los expertos.
- ✓ La elaboración del cuestionario.

#### **3.2. El Proceso de Selección de Expertos**

Como bien se explica anteriormente, es muy importante después de haber decidido aplicar el método Delphi, seleccionar a los especialistas o expertos en el tema a evaluar. Estos, serán una persona o grupo de personas con un gran conocimiento sobre el tema tratado, que podrán emitir criterios incuestionables de cualquier problema y valoraciones importantes con un alto nivel de conocimiento.

Para la selección de los expertos que realizarán la valoración del Modelo de Confiabilidad propuesto, se tuvo en cuenta una serie de características, que posibilitaron obtener información de la trayectoria y actualizada sobre los posibles expertos. Basándose principalmente, en las participaciones de estos en eventos científicos, que contaran además con una ardua experiencia ocupacional y en el tema de interés. También se tuvo en cuenta los conocimientos que deben tener sobre el problema planteado. Estos conocimientos están basados en las siguientes áreas de trabajo: (ver anexo #2)

- ✓ Calidad del Software.
- ✓ Norma y Estándares de Calidad.
- ✓ Modelos de Calidad
- ✓ Confiabilidad del Software.
- ✓ Aplicación del Modelo de Pruebas(Modelo V)
- ✓ Proceso de Pruebas de Software.
- ✓ Tipos de Pruebas.

Basándose en las informaciones mencionadas anteriormente, la lista de expertos a consultar quedó conformada por 7 especialistas, de una forma u otra vinculados al área de calidad del software y constan con una ardua experiencia en los temas relacionados con el proceso de pruebas.

Nº	Listado de Expertos	Cargos
Nº1	Ing. Daimi Bretones Lorenzo	Asesora de Calidad DATEC
Nº2	Ing. Ramón Enrique González Peralta	Analista del centro Humanización Penitenciaria
Nº3	Ing. Aymara Marín Díaz	Asesora de Calidad del centro Telemática
Nº4	MSc. Violen Hernández Aguilar	Especialista del dpto. de pruebas de software CALISOFT
Nº5	Ing. Heney Días Pérez	Especialista del dpto. de pruebas de software CALISOFT
Nº6	Ing. Yadainy Betancourt Rodríguez	Especialista de la Calidad UCIFAR
Nº7	Ing. Ania Lobaina Legrá	Asesora de Calidad UCIFAR

Tabla 3: Listado de expertos

A este grupo de expertos, se le realizó una autoevaluación de los niveles de formación y argumentación que tiene sobre el tema especificado. En la misma marcaron con una X, en una sucesión creciente del 1 al 10, el valor que se corresponde con el grado de conocimiento o información que tienen cada uno, sobre el tema de prueba, donde los resultados obtenidos se muestran a continuación en la siguiente tabla.

Expertos	1	2	3	4	5	6	7	8	9	10
1										x
2							x			
3					x					
4									x	
5							x			
6									x	
7							x			

Tabla 4: Grado de conocimiento en el tema

Luego del análisis del grado de conocimiento con el que cuentan los especialistas, se pasó a calcular el Coeficiente de Conocimiento o Información (Kc), a través de la siguiente fórmula.

$$Kc = n (0.1)$$

Donde: Kc: Coeficiente de Conocimiento o Información.

n: Rango seleccionado por el experto.

Para el experto N°1  $Kc = 10 \times (0.1) = 1$

Para el experto N°5  $Kc = 8 \times (0.1) = 0.8$

Para el experto N°2  $Kc = 7 \times (0.1) = 0.7$

Para el experto N°6  $Kc = 9 \times (0.1) = 0.9$

Para el experto N°3  $Kc = 5 \times (0.1) = 0.5$

Para el experto N°7  $Kc = 7 \times (0.1) = 0.7$

Para el experto N°4  $Kc = 9 \times (0.1) = 0.9$

Experto	Valor (Kc)
1	1
2	0.7
3	0.5
4	0.9
5	0.8
6	0.9
7	0.7

Tabla 5: Coeficiente de conocimiento

No	Fuente de argumentación	Alto	Medio	Bajo
1	Análisis realizado por Ud.	4-5-6	1-3-7	2
2	Experiencia.	1-4-5-6	2-3-7	
3	Trabajos de autores nacionales.		1-3-4-5-7	2-6
4	Trabajos de autores extranjeros.	4-7	1-5-6	2-3

5	Conocimiento del tema en el extranjero	1-4	2-5-6-7	3
6	Su intuición.		1-2-4-7	3-5-6

Tabla 6: Grado de influencia de los expertos

Para calcular el coeficiente de argumentación o fundamentación de cada experto es necesaria utilizar como factores, los que aparecen en la siguiente tabla patrón:

Fuentes de argumentación.	Alto	Medio	Bajo
Análisis teóricos por Ud. realizados.	0,3	0,2	0,1
Su experiencia obtenida.	0,5	0,4	0,2
Trabajos de autores nacionales.	0,05	0,05	0,05
Trabajos de autores extranjeros.	0,05	0,05	0,05
Conocimiento del problema en el extranjero.	0,05	0,05	0,05
Su intuición.	0,05	0,05	0,05

Tabla 7: Tabla Patrón

A partir de la tabla patrón anterior y la autovaloración realizada por los expertos, se debe calcular Ka (coeficiente de argumentación) de la siguiente manera mediante la ecuación:

$$Ka = \sum ni = (n1 + n2 + n3)$$

Donde:

Ka: Coeficiente de Argumentación.

ni: Valor correspondiente a la fuente de argumentación i (1 hasta 6)

Para Experto N° 1

$$Ka = 0.2 + 0.5 + 0.05 + 0.05 + 0.05 + 0.05 = 0.9$$

Para Experto N° 2

$$Ka = 0.1 + 0.4 + 0.05 + 0.05 + 0.05 + 0.05 = 0.7$$

Para Experto N° 3

$$Ka = 0.2 + 0.4 + 0.05 + 0.05 + 0.05 + 0.05 = 0.7$$

Para Experto N° 4

$$Ka = 0.3 + 0.5 + 0.05 + 0.05 + 0.05 + 0.05 = 1$$

Para Experto N° 5

$$Ka = 0.3 + 0.5 + 0.05 + 0.05 + 0.05 + 0.05 = 1$$

Para Experto N° 6

$$Ka = 0.3 + 0.5 + 0.05 + 0.05 + 0.05 + 0.05 = 1$$

Para Experto N° 7

$$Ka = 0.2 + 0.4 + 0.05 + 0.05 + 0.05 + 0.05 = 0.8$$

Experto	Valor (Ka)
1	0.9
2	0.7
3	0.7
4	1
5	1
6	1
7	0.8

Tabla 8: Coeficiente de argumentación

Ahora estamos en condiciones de calcular el coeficiente de competencia K a través de la fórmula:

$$K = 0.5 \times (Kc + Ka)$$

Para Experto N° 1

$$K = 0.5 \times (0.9 + 1) = 0.95$$

$$K = 0.5 \times (1 + 0.9) = 0.95$$

Para Experto N° 5

Para Experto N° 2

$$K = 0.5 \times (0.8 + 1) = 0.9$$

$$K = 0.5 \times (0.7 + 0.7) = 0.7$$

Para Experto N° 6

Para Experto N° 3

$$K = 0.5 \times (0.9 + 1) = 0.95$$

$$K = 0.5 \times (0.5 + 0.7) = 0.6$$

Para Experto N° 7

Para Experto N° 4

$$K = 0.5 \times (0.7 + 0.8) = 0.75$$

Experto	Valor (K)
1	0.95
2	0.7
3	0.6
4	0.95
5	0.9
6	0.95
7	0.75

Tabla 9: Coeficiente de competencia

El código de interpretación de tales coeficientes de competencias:

Si  $0,8 < K < 1,0$  coeficiente de competencia alto.

Si  $0,5 < K < 0,8$  coeficiente de competencia medio

Si  $K < 0,5$  coeficiente de competencia bajo

Excepto los expertos N° 2, N° 3 y N° 7 que cuentan con un coeficiente de competencia medio, los demás responden a tener competencia alta para la validación de la propuesta.

Una vez escogido el listado de expertos, se invitó a cada uno de ellos de forma personal para que participara en el proceso de validación y aceptación de la propuesta. Se detalló de forma clara y precisa los objetivos de la propuesta, así como la realización del cuestionario. Con este último paso, se termina el proceso de selección del personal que conformará el panel de experto.

#### 3.3. Elaboración del Cuestionario

El cuestionario no es sólo un documento que contiene una lista de preguntas, sino un documento en el que se obtendrá la información para presentar los resultados. En la presentación del cuestionario se tuvieron en cuenta preguntas con enfoques investigativos y que se centran principalmente en los objetivos que debe cumplir la propuesta presentada, además de permitir que las respuestas fueran abiertas y en todos los casos con posibilidad de emitir su criterio personal en cuanto a la propuesta que se presenta para el centro DATEC. En este proceso los expertos que conforman el panel recibieron de forma personal un resumen del Modelo de Pruebas de Confiabilidad, unido al cuestionario que debían responder, en un mínimo de tiempo para analizar las respuestas y pedirles que realizaran preguntas en caso de que surgieran dudas durante el estudio de la misma.

#### 3.1. Validación de la Propuesta

Para que el Modelo de Pruebas de Confiabilidad propuesto, quede aprobado, se tuvieron en cuenta una serie de objetivos, que son expresados en el cuestionario a través de las preguntas aplicadas en el mismo, la correspondencia de los objetivos con cada pregunta se muestra en la siguiente tabla:

Objetivos / Preguntas	1	2	3	4	5	6
Importancia de la propuesta	x					
Satisfacción de las necesidades del centro		x				
Necesidad de empleo de la propuesta		x				
Adaptabilidad a proyecto productivos				x		
Eficacia de la propuesta					x	
Mejora de la confiabilidad de los productos			x			
Recomendaciones						x

Tabla 10: Objetivos de la propuesta por preguntas

En el cuestionario se exhiben 6 preguntas, la primera responde al objetivo número uno importancia de la propuesta, la segunda contribuye a la satisfacción de las necesidades del centro, la tercera a la necesidad de uso de la propuesta, la cuarta a las posibilidades del modelo como propuesta, la quinta el aporte al proceso de pruebas en el centro. Determinándose la eficacia del Modelo de Prueba de confiabilidad

mediante las preguntas 4 y 5. Mientras que las preguntas 6 son opiniones y criterios que pueden tener los expertos de aporte al modelo propuesto. Cubriéndose así en cada una de las preguntas del cuestionario los objetivos trazados. (Ver anexo # 3)

### 3.1.1. Importancia de la propuesta

Para valorar este aspecto se tuvo en cuenta las respuestas de los expertos a la pregunta 1, referidos a la importancia de desarrollar un Modelo de Pruebas de Confiabilidad, para facilitar el desarrollo del proceso de pruebas en el centro DATEC. Las respuestas podían ser positivas, negativas, para mostrar los resultados estadísticamente se evalúan las preguntas en un rango de 1-3, se le asigna a las respuestas positivas un valor de 3, 1 a las negativas y 2 para las restantes opciones. El resultado se muestra a continuación:

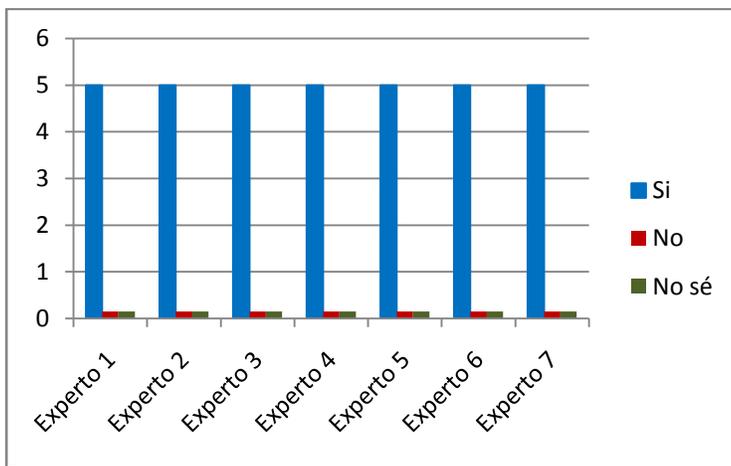


Fig: 12 Importancia de la propuesta

En la gráfica se muestra que los expertos seleccionados coincidieron en su totalidad, que el Modelo propuesto tiene gran importancia para el proceso de pruebas en el centro.

Argumentando su respuesta a través de los criterios que se emiten a continuación:

- Con el mismo se contará con una herramienta para los probadores que determine cuan confiable es el producto elaborado en el centro.
- De esta forma las personas que desarrollan las pruebas tendrán una guía para la ejecución de las pruebas de confiabilidad que se proponen.

### 3.1.2. Satisfacción de las necesidades del centro

En la pregunta 2 del cuestionario, los expertos dieron una valoración en un rango del 1 al 5 siendo el 5 el mayor valor, del nivel de satisfacción que provee el Modelo a las necesidades del centro. El resultado se muestra a continuación:

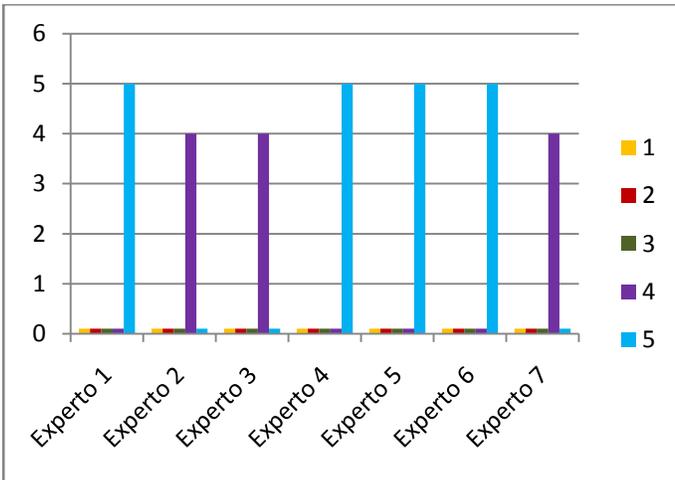


Fig: 13 Satisfacción de las necesidades del centro

Los expertos conciertan de manera general que el modelo de pruebas de confiabilidad, puede satisfacer las necesidades del centro a la hora de realizar las pruebas de software al producto, 3 expertos dan una puntuación de 4 y los otros de 5 puntos, sin mostrar muchas diferencias en los criterios expresados.

### 3.1.3. Necesidad de empleo de la propuesta

Este objetivo se evalúa de la misma condición que el anterior, los expertos dieron una valoración en un rango del 1 al 5 siendo el 5 el mayor valor, al grado de necesidad que existe en el centro de emplearse esta propuesta. El resultado se muestra a continuación:

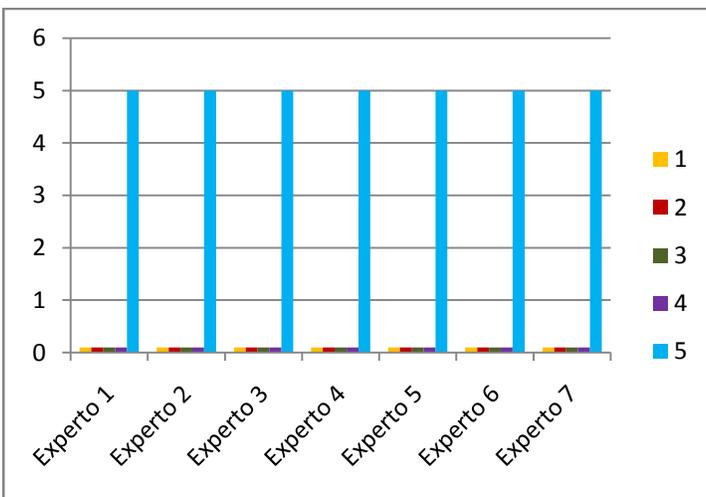


Fig: 14 Necesidad de empleo de la propuesta

### 3.1.4. Adaptabilidad a proyectos productivos

Al igual que los objetivos anteriores se evalúa la misma condición.

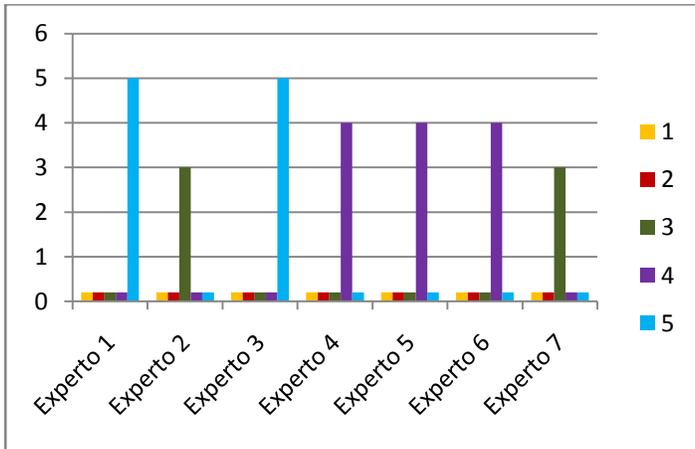


Fig: 15 Necesidad de empleo de la propuesta

Los expertos emitieron diferentes opiniones, ninguno por debajo de un valor de 3 puntos. Aunque en su totalidad expresaron que en dependencia de las necesidades de los proyectos a los cuales están vinculados, pudiera refinarse.

### 3.1.5. Eficacia de la propuesta

Para apreciar este aspecto se tuvo en cuenta las respuestas de los expertos a la pregunta 5, referidos a la eficacia de la propuesta. Las respuestas podían ser *No sería eficaz*, *Pudiera serlo o no serlo* y *Sería eficaz*, para mostrar los resultados estadísticamente se evalúan las preguntas en un rango de 1-3, se le asigna a las respuestas positivas un valor de 3, 1 a las negativas y 2 para las restantes opciones. El resultado se muestra a continuación:

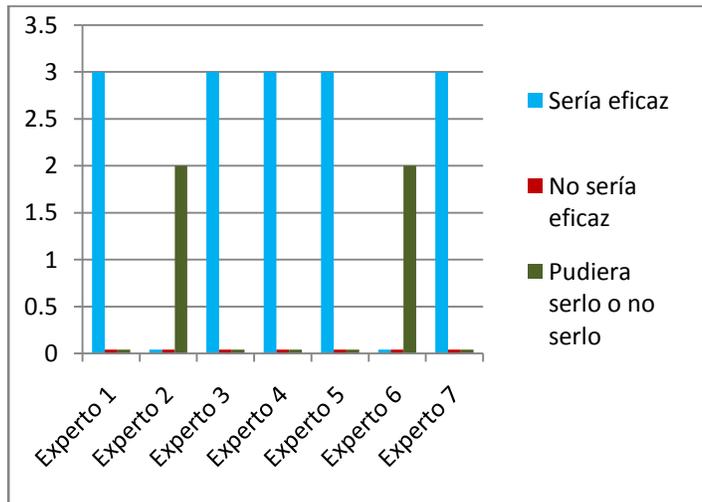


Fig: 16 Eficacia de la propuesta

### 3.1.6. Mejora de la confiabilidad de los productos

Las respuestas emitidas por los expertos en la pregunta 3, las cuales podían ser, *Sí*, *No sé* y *No*, arrojó los resultados que se muestran posteriormente. En los mismos se concuerda con que el modelo de pruebas propuesto para la confiabilidad de los productos del centro, si contribuye a la mejora de la calidad de los productos elaborados.

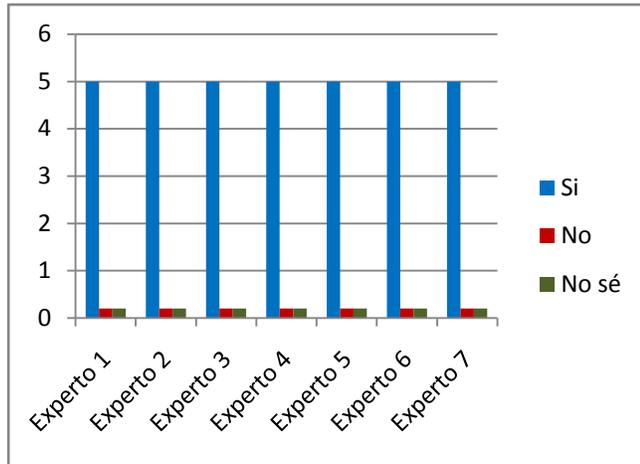


Fig. 17: Mejora de la confiabilidad de los productos

### 3.1.7. Recomendaciones de los Expertos

La pregunta número 6 del cuestionario brinda la posibilidad a cada uno de los expertos especificar los factores que creen que pueden estar a favor de la aplicación del Modelo de Pruebas de Confiabilidad y sugerencias para su perfeccionamiento. Mostrándose las siguientes opiniones:

Sugerencias:

- Realizar otras pruebas que respondan que tributen a las dimensiones de confiabilidad, según las necesidades de un proyecto en específico.
- Es importante detallar el significado de cada uno de los indicadores, definir qué significa y representa para cada prueba.
- Para cada indicador, definir los tipos o las clasificaciones de las no conformidades que responden a ese indicador.

A favor:

- Es importante tener conocimiento de cómo probar este atributo de calidad explicando la secuencia de pasos a seguir, se valora de muy útil para el mejoramiento del proceso de construcción del software el modelo propuesto.

- Es eficaz en el problema en cuestión para las problemáticas de DATEC.
- Se considera que el tema de pruebas es muy amplio, asignarle la responsabilidad de saber cuán buena es la confiabilidad de un centro es de muy buena idea y la presente tesis le da una muy buena importancia a la confiabilidad que es un atributo importante dentro de la calidad.
- Permite evaluar la confiabilidad de manera efectiva, lo cual es algo novedoso.
- La propuesta da una medida de la confiabilidad de los productos de software desde un punto de vista evaluativo.
- Abarca el tema de las pruebas, lo que hace el modelo muy útil, pues en nuestra universidad el tema de las pruebas está un poco distante con respecto a los atributos de la calidad, y este en específico trata la confiabilidad.

#### 3.2. Resultados Finales de la Validación por Experto

Los resultados obtenidos en la validación realizada por cada experto al Modelo de Pruebas de Confiabilidad propuesto, dieron resultados positivos en cada uno de los objetivos planteados. Aunque en algunos casos los expertos se expresaron de forma diferente, la mayoría de ellos coincidieron al opinar que el modelo resultaría importante para el centro. A continuación se muestra una figura con por ciento de concordancia de las respuestas de los expertos por objetivos.



Fig: 17 Resultados finales de la validación

Algunas de las opiniones aportadas por los expertos en el punto número 6:

- ✓ Se debe estudiar a fondo también el tipo de proyecto o negocio, muchos de estos aspectos son tratados de igual forma en productos de software.

- ✓ Es importante saber cómo probar este atributo de calidad explicando la secuencia de pasos a seguir, sin obviar ninguno de sus sub-características, por lo que se valora de muy importante para el mejoramiento del proceso de construcción del software el modelo propuesto.

#### 3.3. Validación de la propuesta a través de un producto

Para la validez del modelo, una vez analizadas y acogidas las sugerencias en apoyo a la elaboración del mismo, a través de los expertos, se tomó como muestra un producto del centro (*Generador de Reportes*), perteneciente a la línea *Integración de Soluciones*, este producto está sustentado en funciones de bases de datos, y su objetivo principal es generar reportes específicos, según las consultas realizadas, así como otras funciones.

Los valores que se le asignaron a los indicadores de cada una de las pruebas que se proponen, se hizo teniendo en cuenta un orden de prioridades, según lo que se observó en las planillas de los casos de pruebas del producto seleccionado.

Pruebas de Sistema	Grado de Importancia	No Conformidad
Respuesta ante sobrecarga de datos	0.2	0
Continuidad de las operaciones	0.1	1
Disponibilidad de Recursos	0.2	1
Rendimiento	0.5	2
Pruebas Estructurales		
Existencia de errores	0.2	2
Prevención de errores por fallas	0.1	1
Evolución de versiones	0.1	2
Errores de estructura y acceso a datos	0.6	1
Pruebas de Integración		
Interoperabilidad	0.4	0
Veracidad	0.1	1
Tiempo de respuestas	0.4	0
Control de versiones	0.1	4
Pruebas de Componentes		
Navegación	0.2	3
Interfaz	0.2	5
Mecanismos de comunicación	0.2	1
Funcionamiento	0.2	2
Componentes	0.2	0
Pruebas de Recuperabilidad		

Tolerancia ante falla	0.2	1
Mecanismos de protección	0.3	1
Capacidad de retorno a un estado estable	0.2	1
Disponibilidad de datos	0.3	1

Tabla 11: Asignación del grado de importancia y las no conformidades

Aplicando las fórmulas propuestas por cada uno de los indicadores:

$$C_{pb} = \sum G_j P_i$$

CpS = Confiabilidad de las pruebas de sistema

$$C_{pS} = \sum 0.2 * 5 + 0.1 * 4 + 0.2 * 4 + 0.5 * 3$$

$$C_{pS} = 3.7$$

CpE = Confiabilidad de las pruebas de estructura

$$C_{pE} = \sum 0.2 * 3 + 0.1 * 4 + 0.1 * 3 + 0.6 * 4$$

$$C_{pE} = 3.7$$

CpI = Confiabilidad de las pruebas de integración

$$C_{pI} = \sum 0.4 * 5 + 0.1 * 4 + 0.4 * 5 + 0.1 * 2$$

$$C_{pI} = 4.6$$

CpC = Confiabilidad de las pruebas de componente

$$C_{pC} = \sum 0.2 * 3 + 0.2 * 1 + 0.2 * 4 + 0.2 * 3 + 0.2 * 5$$

$$C_{pC} = 3.2$$

CpR = Confiabilidad de las pruebas de recuperabilidad

$$C_{pR} = \sum 0.2 * 4 + 0.3 * 4 + 0.2 * 4 + 0.3 * 4$$

$$C_{pR} = 4$$

Luego de obtener el resultado por cada una de las pruebas, se continúa con la siguiente fórmula:

$$C_p = 0.2 \sum \text{Confiabilidad Prueba (j)}$$

$$C_p = 0.2 \sum 3.7 + 3.7 + 4.6 + 3.2 + 4$$

$$C_p = 3.84$$

Teniendo en cuenta los resultados obtenidos, el producto cuenta con un por ciento de confiabilidad entre (60 - 79), por lo tanto, el nivel de confiabilidad de ese producto en específico es *Medio*.

**Conclusiones Parciales**

El análisis de los resultados anteriores permite afirmar de modo general que el Modelo de Pruebas de Confiabilidad de Pruebas, fue evaluado por los expertos como efectivo, correcto y necesario para desarrollar en el proceso de pruebas del centro DATEC. Además de obtener resultados positivos, en un producto específico desarrollado en el centro.

### Conclusiones

Al terminar esta investigación se ha llegado a las siguientes Conclusiones Generales:

- Se desarrolló un estudio exhaustivo de la bibliografía relacionada con el tema como parte de la fundamentación teórica definiéndose la posición del investigador.
- El uso de modelos y estándares ha sido muy conveniente y productivo, porque logró definir las bases para la elaboración del Modelo de Pruebas de Confiabilidad, cada uno de ellos introdujo la forma en la que definen confiabilidad, puesto es que es el punto clave de nuestra investigación.
- Seleccionar el Modelo V como base, garantizó los patrones para obtener un Modelo de Pruebas de Confiabilidad.
- Al realizar la propuesta del Modelo de Pruebas de Confiabilidad, basado en los diferentes modelos analizados, quedó confeccionada una herramienta que proporciona gran utilidad a los probadores del centro, la cual permite evaluar la confiabilidad de los productos elaborados en el centro.
- Con el diseño del Modelo de Pruebas de Confiabilidad, quedó elaborada una herramienta que contribuirá a la calidad del desarrollo del proceso de pruebas en el centro DATEC.

### **Recomendaciones**

De acuerdo la investigación desarrollada, se puede decir que para la aplicación del Modelo de Pruebas de Confiabilidad propuesto se recomienda lo siguiente:

- Aplicar la solución propuesta a la variedad de productos de DATEC, teniendo en cuenta los casos de pruebas, en función de los indicadores que se proponen.
  
- Incorporar conocimientos de pruebas de software en el Modelo V, el cual permitirá formar tempranamente estudiantes con cultura de pruebas y las herramientas necesarias para su desempeño como probadores.

### **Bibliografía**

1. Calidad en Modelos Conceptuales: Un Análisis multidimensional de Modelos Cuantitativos Basados en la ISO 9126.
2. Mejora de la Calidad de Desarrollos Orientados a Objetos Utilizando UML para la Obtención de Casos de Pruebas.
3. Una propuesta para la verificación de Requisitos Basada en Métricas.
4. Gestión de las Pruebas Funcionales.
5. Project Management Institute. Guía de los Fundamentos de la Dirección de Proyectos (Guía del PMBOK). EE.UU: s.n., 2004. Tercera Edición.
6. Pressman, Roger S. Un enfoque práctico. 2005

### Referencias Bibliográficas

1. Proyecto. *Qué es un proyecto informático*. [En línea] [Citado el: 11 3, 2009.]  
<http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c1/c1>.
2. (PMI), Project Management Institute. *Guía de los Fundamentos de la Dirección de Proyectos*. s.l. : Cuarta Edición.
3. [En línea] [Citado el: 11 15, 2009.] [http://scielo.sld.cu/scielo.php?pid=S1024-94352009001200003&script=sci\\_arttext](http://scielo.sld.cu/scielo.php?pid=S1024-94352009001200003&script=sci_arttext).
4. [En línea] [Citado el: 11 15, 2009.]  
[http://rigel.fca.unam.mx/~li406081049/gestion\\_de\\_la\\_informacion/PMBOK.doc](http://rigel.fca.unam.mx/~li406081049/gestion_de_la_informacion/PMBOK.doc).
5. [En línea] [Citado el: 1 5, 2010.] <http://prof.usb.ve/lmendoza/Documentos/PS-6117%20%28Teor%EDA%29/PS6117%20Calidad%20del%20Software.pdf>.
6. Sotés, Ing. MsC. Roberto Félix Zamuriano. *Las Inspecciones de Software y las Listas de Comprobación*.
7. [En línea] <http://prof.usb.ve/lmendoza/Documentos/PS-6117%20%28Teor%EDA%29/PS6117%20Calidad%20del%20Software.pdf>.
8. [En línea] <http://bdigital.eafit.edu.co/bdigital/PROYECTO/P005.12CDD946/capitulo2.pdf>.
9. *Resolución Minsiterial N° 139-2004-PCM. Guía Técnica sobre la Evaluación* .
10. [En línea] [Citado el: 12 15, 2009.]  
[http://www.mantenimientomundial.com/sites/latiman/news\\_analistas-internacionales%283%29.html](http://www.mantenimientomundial.com/sites/latiman/news_analistas-internacionales%283%29.html).
11. [En línea] [Citado el: 12 15, 2009.] [http://scielo.sld.cu/scielo.php?pid=S1024-94352009001200003&script=sci\\_arttext](http://scielo.sld.cu/scielo.php?pid=S1024-94352009001200003&script=sci_arttext).
12. [En línea] [Citado el: 12 15, 2009.] <http://www.monografias.com/trabajos-pdf/inspecciones-software/inspecciones-software.pdf>.
13. [En línea] [Citado el: 1 14, 2010.] <http://portal.centald.prod.uci.cu>.
14. *Actas de Talleres de Ingeniería del Software y Bases de Datos* .
15. [En línea] [Citado el: 12 20, 2009.]  
[http://www.unisdr.org/eng/public\\_aware/world\\_camp/2004/booklet-spa/page9-spa.pdf](http://www.unisdr.org/eng/public_aware/world_camp/2004/booklet-spa/page9-spa.pdf).
16. [En línea] <http://www.gestiopolis.com/canales7/fin/epistemologia-de-la-contabilidad.htm>.
17. [En línea] <http://www.trabajos20/pruebas-de-software/pruebas-de-software.shtml>.
18. Fillotrani, Pablo R. *Calidad en el desarrollo del software "Modelos de calidad del software"* .
19. [En línea] <http://prof.usb.ve/lmendoza/Documentos/PS-6117%20%28Teor%EDA%29/PS6117%20Calidad%20del%20Software.pdf>.

## Referencias Bibliográficas

20. Erika Camacho, Fabio Cardeso, Gabriel Nuñez. *Arquitecturas de software. Guía de estudio.*
21. *Norma Internacional ISO/IEC 9126-2001.* .
22. *Universidad de Ciencias Informática. Comunidad Técnica Cubana de PostgreSQL.* .
23. [En línea] <http://datawarehouse.ittoolbox.com/>..
24. [En línea] <http://www.lsi.us.es/redmidas/CEDI/papers/933.pdf>..
25. mitoloca.com. [En línea] <http://www.mitoluca.com.mx/articulos/bi.htm>.
26. Astigarraga, Eneko. *El Método Delphi.*
27. [En línea] [http://www.babylon.com/definicion/Pruebas\\_de\\_Integración/](http://www.babylon.com/definicion/Pruebas_de_Integración/)..

### **Glosario de Términos**

*Defecto:* Un defecto se encuentra en un artefacto y puede definirse como una diferencia entre la versión correcta del artefacto y una versión incorrecta. Coincide con la definición de diccionario, "imperfección".

*Error:* Una acción humana que puede producir resultados incorrectos.

*Casos de prueba:* Conjunto de entradas, precondiciones para la ejecución y salidas esperadas desarrolladas con el objetivo de testear un aspecto concreto del software (ejecutar un camino del programa en particular, verificar la conformidad de un requisito concreto, detectar tipos de errores específicos).

*Estrés:* Testeo del comportamiento del sistema bajo cargas muy altas con el objetivo de romper el sistema y encontrar los límites del sistema.

*ISO/IEC 9126:* Estándar que define un modelo de calidad de producto software.

*Modelo:* Es una abstracción del sistema, especificando el sistema desde un punto de vista y un determinando el nivel de abstracción.

## Anexos

### Anexo # 1

¿Qué tipos de errores considera que pudieran ocurrir a la hora de desarrollar su trabajo, no sólo en la implementación, sino también errores para los cuales Ud. no está preparado?

#### Precisión

¿Llevan el control del margen de errores que tienen los cálculos que realizan en las operaciones?

¿Lo tienen registrados en algún lugar o sólo lo archivan?

¿Cuentan con todos los recursos para evitar una posible falla?

#### Recuperabilidad

¿Existe algún mecanismo por el cual el software en ausencia de personal pueda automáticamente restablecerse, en caso de existir alguna catástrofe natural?

¿Cualquier usuario tiene acceso a toda la información del sistema o solo los que trabajan con ella directamente?

#### Continuidad en la operación

¿Realizan alguna prueba para determinar los errores ocasionados por fallas de operabilidad en el programa?

\_\_\_Sí ¿Qué tipo de prueba? ¿Se registran estas fallas?

\_\_\_No ¿Qué le ha faltado?

#### Evolución

¿En qué por ciento de desarrollo está el software?

¿Se ha logrado lo previsto desde la perspectiva con que se inició el proyecto?

¿Quedará su producto en una primera versión o está Ud. mejorando una versión existente?

¿Qué facilidades le brinda esta nueva versión?

¿Qué mejoras tiene sobre la cual estás trabajando?

#### Simplicidad

¿Están sus tareas documentadas o implementada de forma que sean perfectamente comprensibles por otro que no sea Ud.?

\_\_\_Sí

\_\_ Se entiende pero en algunos aspectos necesitarían de mi explicación

\_\_ Se logra el resultado pero no se entiende mucho

**Las pruebas son:** “el proceso de ejecución de un programa con la intención de descubrir un error”

¿Utilizan alguna estrategia de pruebas para evaluar el sistema?

\_\_ Sí ¿Cuál es? ¿Por qué esa?

¿Crees que esa estrategia evalúa de forma general el sistema?

¿Qué tipo de pruebas se realizan al producto o se le están realizando para evaluarlo?

¿Podrías mencionar alguna? ¿Por qué usan esas?

*Pruebas Negativas:* Hacer que el sistema falle intencionalmente para evaluar su capacidad de respuesta frente a un error.

¿Aplican técnicas de pruebas negativas?

\_\_ Sí ¿El resultado que muestra al hacerla es el esperado o el sistema no tiene capacidad de responder?

¿Qué hacen con esos resultados? ¿Crees que son factibles con el sistema?

\_\_ No ¿Crees que es necesario hacerlas?

*Pruebas de volumen:* Someter al software a una gran cantidad de datos para determinar si los límites alcanzados hacen que este falle.

¿Aplican pruebas de volumen?

\_\_ Sí ¿El resultado que mostró era el esperado o está fallando algo en el programa?

¿Qué cantidad de datos son insertados para esta prueba?

¿Escogen estos datos aleatorios o datos predeterminados para esta prueba?

\_\_ No ¿Pero crees que el sistema necesitaría hacer esta prueba?

*Pruebas de recuperación:* exponer el software a condiciones extremas y verificar que la recuperación se realiza correctamente.

\_\_ Sí. Siempre las mismas o tienen varias técnicas de recuperación

\_\_ No. Por qué, si saben que son importantes.

Anexo # 2

A continuación se muestran un conjunto competencias. Necesitamos determinar el nivel que Ud. tiene de dichos temas. La puntuación se realizará de la siguiente manera, mientras más positiva sea la respuesta mayor puntuación.

- A. Calidad del Software.
- B. Normas y Estándares de Calidad.
- C. Modelos de Calidad
- D. Confiabilidad del Software.
- E. Modelos de Prueba
- F. Aplicación del Modelo de Pruebas(Modelo V)
- G. Proceso de Pruebas de Software.
- H. Tipos de Pruebas.

Marque con una cruz (X) el grado de conocimiento que Ud. tiene sobre la temática que se investiga:

	1	2	3	4	5	6	7	8	9	10
A										
B										
C										
D										
E										
F										
G										
H										

Luego se aprecian un grupo de aspectos que influyen sobre el nivel de argumentación o fundamentación del tema a analizar, donde se determinan los aspectos de mayor influencia. Marque con una (X) el grado de influencia de cada una de sus fuentes.

No	Fuente de argumentación o fundamentación	Grado de influencia de cada una de las fuentes en sus criterios		
		Alto	Medio	Bajo
1	Análisis realizado por Ud.			
2	Experiencia.			

3	Trabajos de autores nacionales.			
4	Trabajos de autores extranjeros.			
5	Conocimiento del tema en el extranjero			
6	Su intuición.			

Anexo # 3

Cuestionario a Miembros Expertos

Nombre del Encuestado:

Ocupación:

1. ¿Considera usted que es importante el desarrollo de un Modelo de Pruebas de Confiabilidad, que presente una guía de pasos a realizar, para facilitar el desarrollo del proceso de pruebas en el centro DATEC?

Sí\_\_ No\_\_ No Sé\_\_

2. De una valoración de 1 a 5(siendo 5 el mayor valor) de los criterios expuestos a continuación según la propuesta a validar.

\_\_Satisfacción de las necesidades en el centro

\_\_Necesidad del empleo de la propuesta

\_\_Adaptabilidad a proyectos productivos

\_\_Aporte al proceso de pruebas implementado en la UCI

\_\_Impacto en el proceso de liberación de software.

3. ¿Considera usted que se pueda aplicar exitosamente en el centro el modelo de pruebas de confiabilidad propuesto, de manera que permita mejorar la calidad de los productos que se desarrollan dando el cumplimiento a su objetivo?

Sí\_\_ No\_\_ No Sé\_\_

4. ¿Considera usted que se logra demostrar mediante la guía de pasos que el modelo propone, que el software desarrollado tiene buena confiabilidad?

Sí\_ No\_\_ No Sé\_\_

5. Luego de conocer a profundidad el Modelo de Prueba para el ingeniero de prueba. Valore la eficacia que este pudiera tener.

No sería eficaz

Pudiera serlo o no serlo

Sería eficaz

6. Haga un comentario o aporte sobre el Modelo de pruebas de confiabilidad que usted está evaluando. (El comentario es libre y debe reflejar algún elemento de interés que aporte elementos a la mejora del mismo)