

**Universidad de las Ciencias Informáticas**

**Facultad 2**



***Diseño de un Modelo de Medición de la confiabilidad y análisis de las tecnologías en los productos de DATEC***

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

**Autores:** Maite Basulto Socorro

Norma Cabreja Quinta

**Tutora:** Ing. Liudmila Borges Calvo

Ciudad de la Habana, junio 2010, Cuba

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

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

---

**Firma del autor**

Norma Cabreja Quinta

---

**Firma del autor**

Maite Basulto Socorro

Liudmila Borges Calvo

---

**Firma del Tutor**

Quiero agradecer:

*A mi mamita Chicha, que aunque no esta aquí, yo se que esta presente con su pensamiento, y deseándome todo lo mejor en este momento.*

*A mi papa que siempre me ha apoyado y me ha dado toda su ayuda y comprensión.*

*A toda mi familia, mis abuelos, mis tías, mis primas, que siempre han estado pendientes de mí.*

*A mis hermanos de sangre: Julio y Yasmína, que los quiero muchísimo...*

*A mis hermanas, que no son de sangre pero son las hermanas que nunca tuve, mis amigas para siempre, Katy y Leydis.*

*A mi novio Aramis, por siempre cuidar de mí, y darme su apoyo.*

*A todas mis amistades, y a mis compañeras de apartamento.*

*A mi compañera de tesis, por ser tan preocupada, y soportarme tanto, y compartir conmigo todo este tiempo.*

*A todos los profesores y personas que de una manera u otra me han ayudado a formarme como ingeniera.*

**Maíte**

Mil gracias:

*A mi abuelita Norma, pues aun sin estar sigue siendo la luz de mis ojos y el aire que respiro, gracias a tí.*

*A mi mamá por estar siempre a mi lado en todo momento, sin tí no hubiese logrado esto, te amo.*

*A mi papá por no dejar de confiar en mí, siempre ha sido mi ejemplo como profesional.*

*A mis hermanos Maco, Zarita, Ale y Angelito, por darme su cariño y apoyo incondicional.*

*A mis tíos Angel Mario y Julita, por una vez acogerme y quererme, también como hija suya.*

*A mi primo Mario, por ser como un hermano y mi mejor amigo, te adoro.*

*A mis tías y tíos de aquí de la Habana por apoyarme en todo momento.*

*A toda mi familia en general por formar parte de mi vida, especialmente a mi tía Angélica, te quiero mucho.*

*A Maite por ser la mejor compañera de tesis que pude haber tenido.*

*A mis aldeanos Alex, Javico y especialmente a Orlin, a ti, gracias por estar a mi lado y enseñarme otra forma de ver la vida, te quiero mucho.*

*A Anabel, Noisy, Dayrelis, Katy, Anisley, Sucel y Annia, por estar siempre a mi lado de una forma u otra.*

*A mis amigas de toda la vida Yadira y Madelin y a todas mis amistades en general.*

*A todas las niñas y niños del grupo del año al que pertenezco por todos los momentos lindos que me hicieron pasar.*

*A todas las niñas y niños del grupo 2501 por acogerme como uno más de ellos.*

*A todos los profes que a lo largo de la carrera han sabido ser también amigos.*

*A la UCI por darme la oportunidad de ver mi sueño hecho realidad.*

*A todos los que menciono y los que no de verdad muchas GRACIAS.*

*Norma*

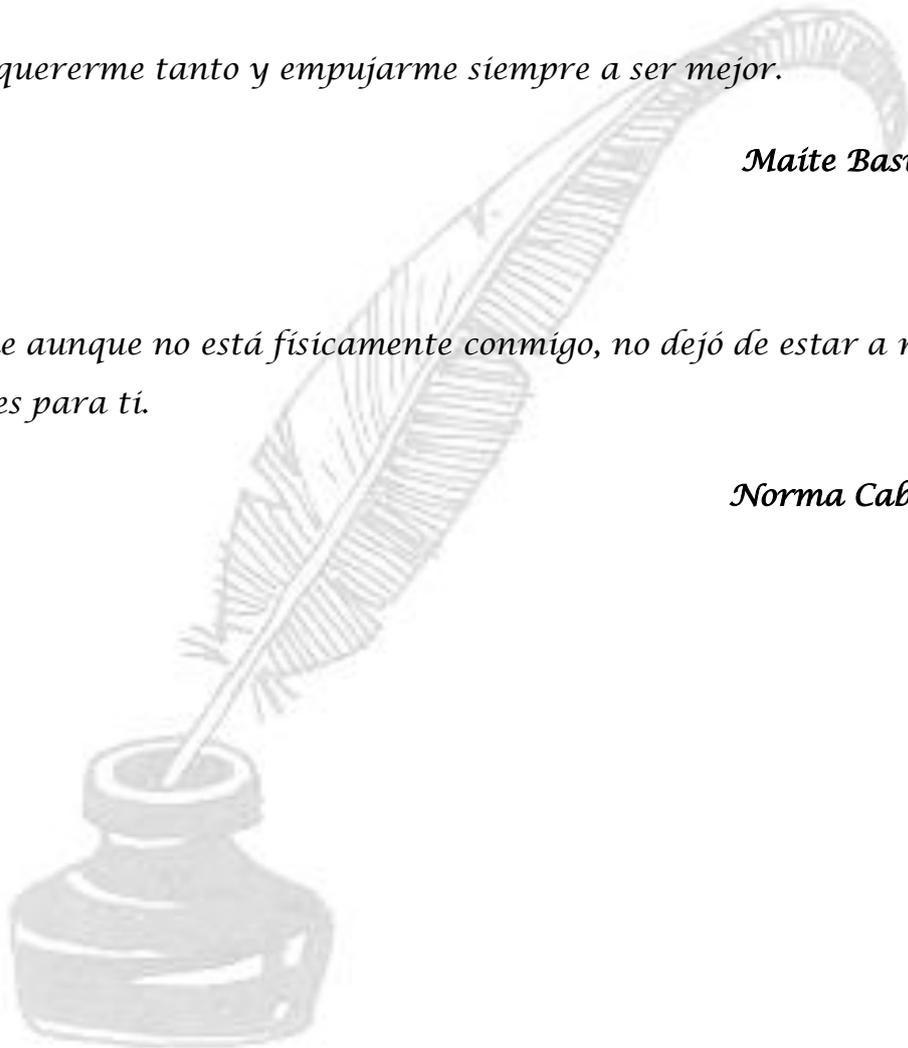
*A mi madre por ser la luz de mis días y la máxima fuente de inspiración. Te adoro.*

*A mi padre, por quererme tanto y empujarme siempre a ser mejor.*

*Maíte Basulto Socorro*

*A mi abuelita que aunque no está físicamente conmigo, no dejó de estar a mi lado y de cuidarme... Esto es para tí.*

*Norma Cabreja Quinta*



En el proceso de gestión y administración de proyectos, la calidad del software es una preocupación a la que se dedican muchos esfuerzos. Para que un producto quede con buena calidad, es necesario además, aplicar a lo largo del ciclo de vida de su realización modelos y estándares de calidad. La calidad es definida además por una serie de factores sin los cuales se puede asegurar que no existe calidad. Dentro de estos factores se encuentra la confiabilidad del software, el cual es uno de los más importantes y sobre el cual estuvo basada la investigación.

El desarrollo de un producto y el mismo al final estaría expuesto a varios riesgos en caso de no llevar un estricto control del factor confiabilidad.

La Universidad de las Ciencias Informáticas (UCI) cuenta hoy en día con varios centros de desarrollo de software dentro de los que se encuentra el Centro de Tecnología de Datos (DATEC). En dicho centro no se cuenta con ninguna herramienta o solución que les permita medir el factor confiabilidad de los proyectos que desarrollan.

Como resultado de la investigación y teniendo como guía el modelo de calidad CMMI (Capability Maturity Model Integration, Modelo de Madurez de Capacidad Integrada), se creó un modelo de medición de la confiabilidad para los productos del centro de DATEC, con el fin de obtener resultados cuantitativos a cerca del factor en cuestión, así como del nivel en el que se encuentra el mismo en el producto al que se esté aplicando el modelo.

**Palabras claves:** Calidad del software, modelos, estándares de calidad, confiabilidad, medición.

---

INTRODUCCIÓN .....	1
1. CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.....	6
1.1. Introducción .....	6
1.2. Calidad del software .....	6
1.2.1. La calidad en el mundo. ....	9
1.2.2. La calidad en Cuba .....	9
1.3. Confiabilidad del software .....	10
1.4. Medición .....	11
1.5. Modelo de McCall .....	14
1.6. Modelo de Boehm .....	16
1.7. Modelo de ISO 9126 .....	17
1.8. La calidad en la UCI.....	22
1.9. DATEC (Centro de Tecnologías de Datos) .....	23
1.10. Conclusiones del capítulo .....	24
2. CAPÍTULO 2. PROPUESTA DE LA INVESTIGACIÓN .....	26
2.1. Introducción.....	26
2.2. Caracterización de la situación existente en DATEC .....	26
2.2.1. Entrevista .....	27
2.2.2. Análisis de las entrevistas .....	29
2.2.3. Resumen de las entrevistas .....	34
2.3. Análisis de los diferentes resultados científicos.....	34
2.4. El modelo como alternativa a seguir .....	35
2.5. CMMI como guía a seguir .....	<b>¡Error! Marcador no definido.</b>

2.6.	Modelo de Medición de Confiabilidad y análisis de las tecnologías .....	36
2.1.1.	Análisis de las tecnologías.....	36
2.6.1.	Estructura del modelo.....	39
2.6.2.	Descripción del modelo .....	41
2.6.3.	Obtención de los resultados.....	56
2.1.	Conclusiones del capítulo .....	59
3.	CAPÍTULO 3. VALIDACIÓN Y APLICACIÓN DE LA PROPUESTA.....	60
3.1.	Introducción.....	60
3.2.	Validación de la propuesta por el método de Expertos.....	60
3.2.1.	Guía para la validación de la propuesta .....	61
3.2.2.	Resultados de la evaluación del Modelo .....	66
3.3.	Aplicación de la Propuesta en el proyecto Generador de Reportes.....	67
3.3.1.	Efectividad del Modelo de medición de la confiabilidad en el proyecto "Generador de Reportes ".....	67
3.3.2.	Resultado final de confiabilidad.....	70
3.4.	Conclusiones del capítulo .....	71
	CONCLUSIONES .....	72
	RECOMENDACIONES .....	73
	REFERENCIAS BIBLIOGRÁFICAS.....	<b>¡Error! Marcador no definido.</b>
	BIBLIOGRAFÍA .....	<b>¡Error! Marcador no definido.</b>
	ANEXOS.....	<b>¡Error! Marcador no definido.</b>
	GLOSARIO DE TÉRMINOS Y ACRÓNIMOS .....	<b>¡Error! Marcador no definido.</b>
✓	Términos .....	<b>¡Error! Marcador no definido.</b>
✓	Acrónimos .....	<b>¡Error! Marcador no definido.</b>

## INTRODUCCIÓN

Un proceso fundamental que se debe hacer antes, durante y después de la realización de un proyecto es la gestión del mismo. Este proceso constituye una rama de la Ingeniería de Software que es nombrada como Gestión de Proyectos.

La Gestión de Proyectos Informáticos ha evolucionado a lo largo de los años. Hoy en día, esta tarea, no es tanto una disciplina técnica misteriosa, sino un conjunto de principios, dirigidos a ofrecer un enfoque estructural hacia la toma diaria de decisiones que hacen que un negocio funcione de manera adecuada. Es una disciplina desarrollada por el PMI, del inglés Project Management Institute (Instituto de Administración de Proyectos) y se rige por el PMBOK que no es más que el cuerpo de conocimientos relativos a la administración de proyectos (Project Management Body of Knowledge), creado por el mismo PMI y considerado como un estándar reconocido a nivel internacional, de las buenas prácticas, herramientas y habilidades para administrar un proyecto. El mismo tiene definido nueve áreas de conocimiento para la administración siendo una de la más importante la destinada a la Calidad del Software. (1)

En el proceso de gestión y administración de proyectos, la calidad del software es una preocupación a la que se dedican muchos esfuerzos. Sin embargo, el software casi nunca es perfecto. Todo proyecto tiene como objetivo producir software de la mejor calidad posible, que cumpla las expectativas de los usuarios como se muestra en la Figura #1.

### **Figura 1. Propósito de los Sistemas de Gestión de Calidad**

Para que un producto quede con buena calidad es necesario a lo largo del ciclo de vida de su realización aplicar modelos y estándares, con el fin, como se plantea anteriormente de lograr un producto que satisfaga las necesidades reales de los clientes. Cuando se aplica un modelo de calidad, se busca desarrollar sistemáticamente productos y servicios que cumplan los requerimientos y las exigencias de los clientes.

Actualmente existen varios Modelos referentes a la Calidad del Software, dentro de los cuales se pueden destacar: McCall, Boehm y CMMI (Capability Maturity Model Integration).

También existen estándares de calidad, dentro de los cuales se pueden encontrar: Norma ISO/IEC 2207, Métrica 3, ISO 9000 (9001:2000), ISO 14598, entre otros.

En algunos de los modelos anteriormente expresados se analizan los factores indispensables con los que tiene que cumplir un software para que quede con buena calidad. De dichos factores se estudiará en profundidad el factor confiabilidad por ser uno de los más importantes.

Un producto se considera confiable si cuenta con varios atributos, los cuales deben ser medidos con el fin de tener un conocimiento cuantitativo de los mismos. Dentro de estos atributos se pueden mencionar: la madurez, la tolerancia a fallos, la precisión, la exactitud, la recuperabilidad, entre otros. De no tenerse en cuenta estos atributos el desarrollo del software y el producto final estarían expuestos a varios riesgos, que de no ser prevenidos con antelación provocarían la falla total del software. Algunos de estos riesgos podrían ser: fallos totales, pérdida de información y tiempo, se tendrían costos sumamente grandes de mantenimiento, riesgos de integridad del producto y de la información que se esté manejando, insatisfacción por parte de los clientes y resultados no acordes con lo especificado en los requisitos, entre otros.

La Universidad de las Ciencias Informáticas (UCI), es uno de los centros de producción de software más importantes del país. La misma cuenta con varios centros de desarrollo de software, dentro de los que se encuentra DATEC (Centro de Tecnologías de Datos).

DATEC es un centro encargado de proveer soluciones integrales y consultorías relacionadas con tecnologías de Base de Datos y análisis de información, enfocado a la toma de decisiones.

Se considera que el centro debería contar con alguna herramienta que les permita medir o llevar un control de la confiabilidad de los software que producen. Teniendo este control podrían evitar exponer sus productos a riesgos que atenten contra este factor tan importante que es la confiabilidad del software. De esta forma los productos que desarrollan en el centro saldrían con mejor calidad.

Teniendo en cuenta la problemática anteriormente explicada se plantea como **problema a resolver**: ¿Cómo medir la confiabilidad de los productos, teniendo en cuenta las tecnologías, en el Centro de Tecnologías de Datos (DATEC)?

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

Se plantea como **objetivo general** de la investigación: Diseñar un modelo de medición de la confiabilidad y análisis de las tecnologías de los productos pertenecientes a DATEC.

Con el fin de demostrar como **hipótesis** que la aplicación de un modelo de medición de la confiabilidad y el análisis de las tecnologías en los productos de DATEC permitirá medir cuantitativamente los atributos de confiabilidad (precisión, exactitud, tolerancia a fallos, madurez, recuperabilidad, modularidad, simplicidad, completitud y consistencia) de cada producto.

Para dar cumplimiento al objetivo general del trabajo se proponen las siguientes **tareas de investigación**:

- ✓ Revisión y selección de la bibliográfica para identificar las insuficiencias del tema y las tendencias actuales existentes sobre la medición de la confiabilidad de los productos.
- ✓ Análisis de los modelos y estándares de calidad tales como ISO/IEC 9126-1:2001, NC ISO/IEC 9126-1:2005, ISO/IEC TR 9126-n:200n, McCall, Boehm, CMMI, etc.
- ✓ Análisis de los métodos y procedimientos que se emplean para medir la confiabilidad de un producto.
- ✓ Análisis de los riesgos potenciales asociados a la confiabilidad.
- ✓ Análisis de otras tecnologías similares a Oracle y PostgreSQL. Ej.: MySQL
- ✓ Evaluación de la información obtenida a partir de los métodos teóricos (Analítico- Sintético, Inductivo- Deductivo, Análisis Histórico- Lógico y Sistémico- Estructural) y definir la posición como investigador.
- ✓ Identificación de los involucrados potenciales en el tema y caracterización de su marco de actuación apoyándose en los métodos empíricos.
- ✓ Estudio de las soluciones desarrolladas en DATEC.
- ✓ Adaptación de algunos métodos evaluativos encontrados en función del entorno en el que se realiza la investigación.
- ✓ Elaboración del modelo de medición de la confiabilidad de los productos teniendo en cuenta las tecnologías.

- ✓ Descripción del modelo diseñado acorde a las condiciones del entorno analizado.

Con la realización del presente trabajo se espera obtener los siguientes **resultados**:

- ✓ Modelo para efectuar una medición de la confiabilidad de los productos de software desarrollados en DATEC.
- ✓ Criterios de comprobación y métricas aplicables a Bases de Datos de las soluciones de la Línea de Almacenes de Datos del Centro de Tecnologías de Datos (DATEC).
- ✓ Propuesta y medición de herramientas que soporten el procedimiento de Bases de Datos.

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 rigen 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 medición.

### **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.

Este documento está estructurado en tres capítulos.

**Capítulo 1. Fundamentación Teórica:** Se analizan aspectos teóricos, principales conceptos y definiciones asociadas a calidad de software, confiabilidad en los productos de software y modelo de medición que permitirá de manera exitosa la creación de un modelo para medir la confiabilidad en los tipos de productos de software. Se realiza una valoración crítica de los diferentes modelos y estándares de calidad existentes para la medición de la confiabilidad según la fuente documental existente de la misma.

**Capítulo 2.** Propuesta de la Investigación: Se abordan aspectos relacionados con la solución propuesta como estructura, objetivos y descripción de cada una de las partes del modelo propuesto.

**Capítulo 3.** Validación y Aplicación de la propuesta: En este capítulo se describe la validación de la propuesta por el método de multicriterio de expertos, así como una guía para llevar a cabo esta validación. También se muestran los resultados obtenidos al aplicar el modelo a una muestra de los productos del centro. Además se hace un análisis en dichos resultados para ver el nivel de confiabilidad de dichos productos, y de esta manera se da un resultado final de la validación.

## 1. CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

### 1.1. Introducción

La gestión de proyectos es una de las disciplinas en el mundo del desarrollo de software más importante. Esta disciplina ha conseguido que se lleven a cabo diversos estudios y se ha dividido en partes, dentro de las que se destaca la gestión de la calidad. Actualmente se considera como una serie de actividades con las cuales se logra que un producto quede con la mejor calidad posible y la que a su vez incluye la medición de la confiabilidad de los productos de software.

En este capítulo se aborda el surgimiento y evolución de la gestión de la calidad, así como de la medición de confiabilidad como punto principal, haciendo referencia a algunas de sus características, incluyendo los conceptos de relevancia que son objeto de análisis durante la investigación. Se realiza además un estudio sobre el estado de la gestión de calidad y medición de confiabilidad en el mundo para conocer las tendencias actuales.

### 1.2. Calidad del software

La calidad del software ha sido clasificada de diferentes formas y en diferentes momentos. Dentro de las definiciones más importantes de este término se pueden apreciar las siguientes:

- *“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”*. (2)
- *“El conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas”*. (3)
- *“La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario”*. (4)

Teniendo en cuenta los conceptos y definiciones estudiados de calidad, se puede plantear que la calidad del software no es más que un conjunto de propiedades y características de un producto, proceso o servicio que le confieren su aptitud para satisfacer las necesidades establecidas o implícitas.

### Gestión de la calidad

La Gestión de la calidad no se puede ver desligada a las actividades de las empresas. Sin embargo, durante muchos años se desarrolló con criterios y aplicaciones dispares y su práctica fue ocasional e intuitiva. (5)

En la Figura #2 se muestra un sistema de administración de la calidad junto al proceso de mejora continua. En el mismo se aprecia como el proceso se inicia con los requisitos del cliente, los cuales deben ser obtenidos en el producto final y entregados al mismo. Estos requisitos pasan por un proceso en el que se encuentran las actividades de administración de recursos, realización del producto; incluyendo servicios; medición, análisis y mejora como otras de las actividades y las responsabilidades de la alta dirección de la gestión del proyecto. Todas estas actividades en un proceso cíclico para lograr un producto que satisfaga al máximo los requisitos de los usuarios o clientes.

### **Figura 2. Sistema de Gestión de la calidad**

De ahí que, un sistema de gestión de la calidad es el conjunto de normas interrelacionadas de una empresa u organización por los cuales se administra de forma ordenada la calidad de la misma, en la búsqueda de la satisfacción de las necesidades y expectativas de sus clientes. (5)

El propósito de la Gestión de la Calidad del Software es entender las expectativas del cliente en términos de calidad, y poner en práctica un plan proactivo para satisfacer esas expectativas.

Desde el punto de vista de la calidad, la gestión de la calidad del software está formada por 4 partes, las cuales son: planificación, control, aseguramiento y mejora de la calidad del software. (6)

La **planificación de la calidad del software** es la parte de la Gestión de la Calidad encargada de realizar el proceso administrativo de desarrollar y mantener una relación entre los objetivos y recursos de la organización; y las oportunidades cambiantes del mercado. El objetivo es modelar y remodelar los negocios y productos de la empresa, de manera que se combinen para producir un desarrollo y utilidades satisfactorias.

El **control de la calidad del software** son las técnicas y actividades de carácter operativo, utilizadas para satisfacer los requisitos relativos a la calidad, centradas en 2 objetivos fundamentales: mantener bajo control un proceso y eliminar las causas de los defectos en las diferentes fases del ciclo de vida.

El **aseguramiento de calidad del software** es el conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza de que el software satisfará los requisitos dados de calidad. Este aseguramiento se diseña para cada aplicación antes de comenzar a desarrollarla y no después.

La **mejora de la calidad del software** es la parte de la Gestión de la Calidad que contribuye, por medio de las mediciones, a los análisis de los datos y auditorias, a efectuar mejoras en la calidad del software.

Por otra parte también se define que en la gestión de la calidad se pueden distinguir solo tres etapas, diferentes y sucesivas: (5)

- ✓ El Control de Calidad.
- ✓ El Aseguramiento de la Calidad.
- ✓ La Calidad Total.

### **Control de la Calidad**

Esta primera etapa se caracteriza por la realización de inspecciones y ensayos para comprobar si un producto terminado, cumple con las especificaciones establecidas previamente.

Se trata, sin duda, de una concepción poco competitiva de la Gestión de la calidad, ya que las inspecciones o ensayos tienen lugar "a posteriori", cuando un proceso productivo ha concluido o cuando el producto final está terminado.

### **Aseguramiento de la Calidad**

Se asume que es más rentable prevenir los fallos de calidad que corregirlos o lamentarlos, y se incorpora el concepto de la "prevención" a la gestión de la calidad, que se desarrolla sobre esta nueva idea en las empresas industriales, bajo la denominación de Aseguramiento de la Calidad. (5)

El Aseguramiento de la Calidad es un sistema, o sea, un conjunto organizado de procedimientos bien definidos y entrelazados armónicamente, que requiere unos determinados recursos para funcionar.

### **La Calidad Total**

La Calidad Total supone un nuevo e importante enriquecimiento de la función de la calidad en las empresas, aunque, al no ser un sistema como el aseguramiento de la calidad y al dar lugar a la

descentralización de las actividades de prevención y control, hace que los departamentos de calidad pierdan su relevancia y, llegado el caso, su sentido. (7)

La Calidad Total es una filosofía, una cultura, una estrategia, un estilo de gerencia. De aquí que la Calidad Total sea entendida y aplicada de diferentes formas en distintas empresas y por diferentes asesores especializados.

### 1.2.1. La calidad en el mundo.

Hoy en día las compañías de todo el mundo industrializado reconocen que la calidad del producto se traduce en ahorro de costos y en una mejora general. La industria de desarrollo de software no es la excepción, por lo que en los últimos años se han realizado intensos trabajos para aplicar los conceptos de calidad en el ámbito del software.

Hablar de calidad del software implica la necesidad de contar con parámetros que permitan establecer los niveles mínimos que un producto de este tipo debe alcanzar para que se considere de calidad. (8) El problema es que la mayoría de las características que definen al software no se pueden cuantificar fácilmente; generalmente, se establecen de forma cualitativa, lo que dificulta su medición, ya que se requiere establecer métricas que permitan evaluar cuantitativamente cada característica dependiendo del tipo de software que se pretende calificar; en este sentido se han realizado muchos trabajos que establecen propuestas para el establecimiento de los factores cualitativos que afectan la calidad del software. Entre los principales están los factores de calidad de McCall. Además se han hecho varios intentos por estandarizar los mecanismos de evaluación de calidad del software. Entre los principales están la familia de normas ISO 9000, en especial la ISO 9001 y la ISO 9003-2 (9) , el modelo de niveles de madurez CMM (Capability Maturity Model) (10), el estándar para el aseguramiento de planes de calidad del IEEE 730:1984 (10) y la norma ISO/IEC 9126 (11).

### 1.2.2. La calidad en Cuba

Actualmente en Cuba, existe la necesidad de crear una cultura de calidad en función del mejoramiento en la industria del software del país, con la creación de la UCI en septiembre del 2002, como centro productor, esta preocupación ha ido tomando más fuerza. A este centro universitario se le han sumado otros como son las distintas entidades del Ministerio de la Informática y las Comunicaciones (MIC) y otras universidades, para colaborar con la creación e implantación de una cultura de calidad.

Entre las entidades pertenecientes al MIC, se encuentra DESOFT, empresa desarrolladora de software en el país, la cual cuenta con un grupo de desarrollo en cada provincia. Debido a estas características se le hacía difícil integrar el trabajo de todas las divisiones por lo que se dieron la tarea de adoptar un método que estandarizara el desarrollo del software en cada una de las divisiones que se encuentran distribuidas en todo el país. En la actualidad ya el método tiene años de estarse ejecutando y ha permitido hacer del desarrollo un proceso sostenible en función de la calidad. Actualmente se trabaja para lograr la estandarización del método elaborado acorde a las normas ISO para lograr la certificación del mismo.

Otra de las acciones que se llevan a cabo por el país, para lograr una mayor calidad en el desarrollo del software son los diferentes eventos, convenciones y conferencias. Una de ellos es la convención y exposición internacional de informática, que tiene lugar mayormente en el Palacio de Convenciones de La Habana, esta exposición se consagra al intercambio de experiencias y prácticas científico-profesionales para ofrecer soluciones, contribuciones y bases para la cooperación e integración, en un clima amistoso y fértil entre los actores entusiastas de estas disciplinas de desarrollo del software en función de la calidad. Auspiciado por el Ministerio de la Informática y las Comunicaciones (MIC) y la Cámara de Comercio de la República de Cuba.

### 1.3. Confiabilidad del software

La confiabilidad del software se encuentra en una etapa de formación de desarrollo y es la característica de rendimiento más costosa de conseguir y de garantizar. La medición de esta característica nos ayuda a desarrollar programas de una manera confiable, rápida y a menor costo. La confiabilidad se puede definir como la probabilidad de que se tengan operaciones libres de caídas en un tiempo definido y en un medio adecuado para propósitos específicos. (12)

Estimar la confiabilidad de un software no es algo sencillo de hacer, pero su cuantificación es muy importante para el buen funcionamiento de sus atributos en orden de cómo estos pueden ayudar a realizar la compensación entre las decisiones de confiabilidad, el funcionamiento del programa, costo y calendario del sistema de software.

Puede verse dividida la confiabilidad en dos partes: (13)

**Predicción de la confiabilidad:** Está definida como un lineamiento del programa basado en la evolución segura y fácilmente medible de las propiedades del código.

**Estimación de la confiabilidad:** Pronostica cambios confiables como una función en el calendario de tiempo, en el tiempo de uso de CPU o cantidad horas/hombres que son acumuladas.

La confiabilidad de un sistema está altamente vinculada con la cantidad de averías y fallos que este llega a tener. Las averías se manifiestan en el comportamiento externo del sistema, pero son el resultado de errores internos. Las causas mecánicas o algorítmicas de los errores se denominan fallos. (14)

Existen diferentes tipos de fallos, los cuales provocan inestabilidad dentro de la confiabilidad de los productos, dentro de los que se destacan los siguientes:

- ✓ Fallos transitorios: desaparecen solos al cabo de un tiempo.
- ✓ Fallos permanentes: permanecen hasta que se reparan.
- ✓ Fallos intermitentes: fallos transitorios que ocurren de vez en cuando.

Para aumentar la confiabilidad de un software existen diferentes formas, las cuales se agrupan en dos sectores:

- ✓ Prevención de fallos: se trata de evitar la introducción de fallos en el sistema antes de que empiece su funcionamiento, se divide en evitar los fallos y en la eliminación de estos una vez encontrados.
- ✓ Tolerancia de fallos: lograr que el sistema continúe funcionando a pesar de que se introduzcan fallos.

Por todo lo antes expuesto, la confiabilidad en la ingeniería es usada generalmente para asegurar cierto grado de funcionamiento con éxito de un dispositivo o sistema en un ambiente determinado, durante un período específico de tiempo. (15)

### 1.4. Medición

La medición del software es el proceso por el cual los números o símbolos son asignados a atributos o entidades en el mundo real tal como son descritos de acuerdo a reglas claramente definidas. (10)

Según Pressman una medida proporciona una indicación cuantitativa de extensión, cantidad, dimensiones, capacidad y tamaño de algunos atributos de un proceso o producto.

El IEEE define como métrica “una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado”. (16)

La medición del software se refiere a derivar un valor numérico para algún atributo de un producto de software. El proceso de medición es una actividad clave, ya que de éste depende que los resultados puedan ser confiables y fáciles de evaluar. Los pasos a seguir en este proceso son los siguientes: (17)

1. Seleccionar medidas a realizar:

- ✓ Formular preguntas que la medición intenta responder.
- ✓ Definir las métricas requeridas para responder a esas preguntas.
- ✓ No se recogen otras métricas que no respondan a esas preguntas.

2. Seleccionar componentes a valorar:

- ✓ No es necesario ni deseable estimar los valores de las métricas de todo un sistema de software (conjunto representativo, componentes críticos y fundamentales (utilización continua)).

3. Medir características de los componentes:

Calcular los valores de las métricas procesando la representación del componente (diseño, código,...) con herramientas adecuadas.

4. Identificar componentes anómalos:

- ✓ Comparación de los resultados con mediciones previas registradas en una base de datos.
- ✓ Atención especial a los valores más altos y más bajos pues pueden indicar problemas.

5. Analizar componentes con valores anómalos:

- ✓ Se examinan los componentes para decidir si los valores obtenidos indican que su calidad está en peligro.
- ✓ No siempre indican problemas (por ejemplo, la complejidad de un módulo puede ser alta pero necesaria).

Las métricas de software se definen como “La aplicación continua de mediciones basadas en técnicas para el proceso de desarrollo del software y sus productos para suministrar información relevante a tiempo, así el administrador junto con el empleo de estas técnicas mejorará el proceso y sus productos”. (18) Las métricas de software proveen la información necesaria para la toma de decisiones técnicas.

Las métricas son la maduración de una disciplina, que, según Pressman van a ayudar a la evaluación de los modelos de análisis y de diseño, en donde proporcionarán una indicación de la complejidad de diseños procedimentales y de código fuente, y ayudarán en el diseño de pruebas más efectivas. Es por eso que propone un proceso de medición, el cual se puede caracterizar por cinco actividades: (2)

1. **Formulación:** La obtención de medidas y métricas del software apropiadas para la representación de software en cuestión.
2. **Colección:** El mecanismo empleado para acumular datos necesarios para obtener las métricas formuladas.
3. **Análisis:** El cálculo de las métricas y la aplicación de herramientas matemáticas.
4. **Interpretación:** La evaluación de los resultados de las métricas en un esfuerzo por conseguir una visión interna de la calidad de la representación.
5. **Realimentación:** Recomendaciones obtenidas de la interpretación de métricas técnicas transmitidas al equipo de software.

Las métricas de software incluyen otras actividades, tales como: (15)

- ✓ Estimación de costo y el esfuerzo.
- ✓ Medición de la productividad.
- ✓ Acumulación de datos.
- ✓ Realización de modelos y mediciones de la calidad.
- ✓ Elaboración de modelos de seguridad.
- ✓ Evaluación y modelos de desempeño.

- ✓ Valoración de las capacidades y de la madurez.
- ✓ Administración por métricas.
- ✓ Evaluación del método y herramientas.

En la actualidad pocos dudan de la importancia de la medición para conseguir incrementar la calidad y la productividad en el desarrollo y mantenimiento del software. La necesidad y motivación por medir se ha incrementado notablemente con la preocupación de las organizaciones por alcanzar mayores niveles de madurez y las certificaciones basadas en modelos y normas como ISO 9000, ISO 15504 o CMMI. En efecto, para poder asegurar que un proceso presenta un determinado grado de calidad y mejorarlo es necesario aplicar los correspondientes indicadores. Además, la medición del software nos permite controlar qué es lo que ocurre en los proyectos y predecir su esfuerzo y duración.

La medición de software se clasifica en dos categorías: (19)

**Medidas directas** del proceso de software (Costo, esfuerzo) y del producto (Líneas de código producidas, rapidez de ejecución y efectos reportados.)

**Medidas indirectas** del producto que incluyen funcionalidad, calidad, complejidad, eficiencia, confiabilidad, facilidad de mantenimiento, y muchas otras habilidades.

En el desarrollo de software las métricas son sumamente importantes ya que se utilizan para definir el tamaño, el tiempo y el empuje que tendrá el proyecto, anticipando así costos y tiempo. Lo que ocurre gracias a un historial de métricas, donde cada puesto o responsables de los proyectos, pueden definir su desempeño, y de esta forma comparar con los antecedentes y realizar mejoras donde realmente se necesita mejorar y no incurrir en errores. Pues al estar documentado y especificado por las métricas, podemos ubicarnos exactamente en donde se cometieron errores, y tal vez no eliminarlos por completos, pero si que estos no repercutan de la misma forma en los siguientes proyectos.

### 1.5. Modelo de McCall

El modelo de McCall fue el primero en ser presentado en 1977. Se enfoca en el producto final, identificando atributos claves. Desde el punto de vista del usuario estos atributos se denominan factores de calidad y son normalmente atributos externos pero también se incluyen algunos atributos internos. (6)

Los factores de calidad son demasiados abstractos para ser medidos directamente, por lo que por cada uno de ellos se introducen atributos de bajo nivel denominados criterios de calidad según McCall.

Algunos criterios de calidad son atributos internos, reflejando la creencia de McCall que el atributo interno tiene un efecto directo en el atributo externo correspondiente.

Un nivel más de descomposición es necesario, mapeando cada criterio de calidad en un conjunto de métricas de calidad que son atributos (tanto del producto como del proceso) de muy bajo nivel, medibles directamente.

McCall propone tres perspectivas para agrupar los factores de calidad: (20)

1. Revisión del producto: habilidad para ser cambiado.
2. Transición del producto: adaptabilidad al nuevo ambiente.
3. Operación del producto: características de operación.

La **revisión del producto** incluye los siguientes factores de calidad:

- ✓ Mantenibilidad: esfuerzo requerido para localizar y corregir fallas.
- ✓ Flexibilidad: facilidad de realizar cambios.
- ✓ Testeabilidad: facilidad para realizar el testing, para asegurarse que el producto no tiene errores y cumple con la especificación.

La **transición del producto** incluye los siguientes factores de calidad:

- ✓ Portabilidad: esfuerzo requerido para transferir entre distintos ambientes de operación.
- ✓ Reusabilidad: facilidad de rehusar el software en diferentes contextos.
- ✓ Interoperabilidad: esfuerzo requerido para acoplar el producto con otros sistemas.

La **operación del producto** incluye los siguientes factores de calidad:

- ✓ Correctitud: el grado en el que el producto cumple con su especificación.
- ✓ Confiabilidad: la habilidad del producto de responder ante situaciones no esperadas.

- ✓ Eficiencia: el uso de los recursos tales como tiempo de ejecución y memoria de ejecución.
- ✓ Integridad: protección del programa y sus datos de accesos no autorizados.
- ✓ Usabilidad: facilidad de operación del producto por parte de los usuarios.

Según **McCall** el factor **confiabilidad** incluye los siguientes criterios:

- ✓ Precisión: Atributos del software que proporcionan el grado de precisión requerido en los cálculos y los resultados.
- ✓ Consistencia: Atributos del software que proveen el grado de concisión requerido fundamentalmente en las líneas de código.
- ✓ Tolerancia a fallos: Atributos del software que posibilitan la continuidad del funcionamiento bajo condiciones no usuales.
- ✓ Modularidad: Atributos del software que proporcionan una estructura de módulos altamente independientes.
- ✓ Simplicidad: Atributos del software que posibilitan la implementación de las funciones de la forma más comprensible posible.
- ✓ Exactitud: La precisión de los cálculos y del control.

Sobre la confiabilidad expresa que combina la tolerancia tanto a errores de hardware como de software. Plantea además que es importante el uso de técnicas de programación tales como manejo de excepciones y programación defensiva. Por otra parte dice que la confiabilidad puede ser medida en función del tiempo medio entre fallas, tiempo medio antes de mantenimiento, tiempo medio antes de recuperación y probabilidad de falla. (21)

La medición de cualquiera de estos factores está definida en este modelo en base a 41 métricas. Para cada criterio existe una lista de condiciones que se deben cumplir en distintas etapas: requerimientos (R), diseño (D), implementación (I). Se cuentan las condiciones que se satisfacen en cada una de las etapas, sobre el total posible. Eso da una medida del criterio, que se pondera en partes iguales para medir el factor con los otros criterios asociados al factor. (20)

### 1.6. Modelo de Boehm

Otro de los modelos de calidad más conocido es el presentado por Barry Boehm en 1978. Propone que la utilidad puede considerarse en correspondencia a los tipos de usuarios que quedan involucrados. El primer tipo de usuarios queda satisfecho si el sistema hace lo que el pretende que haga; el segundo tipo es aquel que utiliza el sistema luego de una actualización y el tercero, es el programador que mantiene el sistema. (20)

El modelo presenta una jerarquía de características, cada una de las cuales contribuye a la calidad global. Además el concepto de software exitoso para Boehm abarca las necesidades y expectativas de los usuarios. Incluye características de desempeño de hardware y plantea además que el software debe: (21)

- ✓ Hacer lo que el Usuario quiere que haga.
- ✓ Utilizar recursos de la computadora correcta y eficientemente.
- ✓ Ser fácil de aprender y usar para los usuarios.
- ✓ Estar bien diseñado, bien codificado y ser probado y mantenido fácilmente.

A continuación aparecen representados los factores de calidad según Boehm.

### **Figura 3. Factores de calidad según Boehm**

#### **1.7. Modelo de ISO 9126**

Es un modelo jerárquico con seis atributos principales que contribuyen a la calidad. A diferencia de McCall y Boehm, cada característica de la derecha solo está relacionada con un único atributo de la izquierda. Las características de la derecha se relacionan con la vista del usuario final. (6)

### **Figura 4. Factores de calidad según la ISO 9126**

Como parte del desarrollo de la calidad en Cuba en función de los productos de software se ha llevado a cabo la adaptación de la norma internacional ISO/IEC 9126 por la Oficina Nacional de Normalización (NC), creándose de esta manera la NC-ISO/IEC 9126-1:2005. (22)

Esta norma adaptada por Cuba conserva las mismas características de calidad de los productos de software. Las diferencias fundamentales son las siguientes: (22)

- ✓ la introducción de sub-características normativas, la mayor parte de las cuales están basadas en las sub-características informativas de la ISO/IEC 9126(1991).

- ✓ la especificación de un modelo de calidad.
- ✓ la introducción de la calidad en el uso.
- ✓ la exclusión del proceso de evaluación (que es ahora especificado por la ISO/IEC 14598).
- ✓ la coordinación del contenido con la ISO/IEC 14598-1.

La misma consta de las siguientes partes bajo el título general Ingeniería de software- Calidad del producto: (23)

- ✓ Parte 1: Modelo de calidad
- ✓ Parte 2: Métricas Externas
- ✓ Parte 3: Métricas Internas
- ✓ Parte 4: Calidad en el uso

Dentro de la parte 1: Modelo de calidad se describe un modelo en dos partes para calidad de los productos de software: (24)

- ✓ Calidad interna y externa.
- ✓ Calidad durante el uso.

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

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

A continuación se muestran los elementos que conforman el modelo de calidad para la calidad interna y externa. En el mismo se categorizan los atributos de calidad del software en seis características (la funcionalidad, la confiabilidad, la usabilidad, la eficiencia, la mantenibilidad y la transportabilidad), que a su vez son divididas en sub-características. Las subcaracterísticas pueden medirse a través de métricas internas o externas. (25)

### **Figura 5. Modelo de Calidad para la calidad interna y externa según la norma NC/ISO IEC 9126**

De las 6 características que se proponen en esta norma, se profundizará en la confiabilidad.

Según esta norma, la confiabilidad es: La capacidad del producto de software para mantener un nivel de ejecución especificado cuando se usa bajo las condiciones especificadas.

Como se puede observar la confiabilidad según la ISO está compuesta por 5 subcaracterísticas esenciales que son:

**Madurez:** Capacidad del producto de software de evitar un fallo total como resultado de haberse producido un fallo del software.

**Tolerancia ante fallos:** Capacidad del producto de software de mantener un nivel de ejecución o desempeño especificado en caso de fallos del software o de infracción de su interface especificada.

**Recuperabilidad:** Capacidad del producto de software de restablecer un nivel de ejecución especificado y recuperar los datos directamente afectados en caso de fallo total.

**Conformidad con la confiabilidad:** Capacidad del producto de software para adherirse a las normas que se le apliquen, convenciones, regulaciones, leyes y las prescripciones similares relativas a la confiabilidad.

**Fiabilidad:** Grado que se puede esperar de una aplicación lleve a cabo las operaciones especificadas y con la precisión requerida.

### 1.8. Modelo CMMI

El modelo de capacidad y madurez integrado (CMMI), es uno de los modelos de calidad de software más importantes y conocidos internacionalmente. Este clasifica las empresas en niveles de madurez y capacidad a la vez. Por lo que tiene definidas dos representaciones, una continua y otra escalonada. Son equivalentes, y cada organización puede optar por adoptar la que se adapte a sus características y prioridades de mejora.

#### Representación continua de CMMI

El modelo para ingeniería de sistemas (SE-CMM)

Establece 6 niveles posibles de capacidad para cada una de las 18 áreas de proceso implicadas en la ingeniería de sistemas. No agrupa los procesos en 5 tramos para definir el nivel de madurez de la

organización, sino que directamente analiza la capacidad de cada proceso por separado. Es lo que se denomina un modelo continuo.

### **Figura 6. Visión Continua de CMMI**

Los 6 niveles definidos en CMMI para medir la capacidad de los procesos son:

0- Incompleto: El proceso no se realiza, o no se consiguen sus objetivos.

1- Ejecutado: El proceso se ejecuta y se logra su objetivo.

2- Gestionado: Además de ejecutarse, el proceso se planifica, se revisa y se evalúa para comprobar que cumple los requisitos.

3- Definido: Además de ser un proceso "gestionado" se ajusta a la política de procesos que existe en la organización, alineada con las directivas de la empresa.

4- Cuantitativamente gestionado: Además de ser un proceso definido se controla utilizando técnicas cuantitativas.

5- Optimizado: Además de ser un proceso cuantitativamente gestionado, de forma sistemática se revisa y modifica o cambia para adaptarlo a los objetivos del negocio.

Representación escalonada de CMMI

El modelo para software (CMM-SW)

Establece 5 niveles de madurez para clasificar a las organizaciones, en función de qué áreas de procesos consiguen sus objetivos y se gestionan con principios de ingeniería. Es lo que se denomina un modelo escalonado, o centrado en la madurez de la organización.

### **Figura 7. Visión escalonada de CMMI**

Los niveles son:

1- Inicial: Estado inicial donde el desarrollo se basa en la heroicidad y responsabilidad de los individuos. Los procedimientos son inexistentes.

2- Gestionado: Se normalizan las buenas prácticas en el desarrollo de proyectos (con base en la experiencia y el método). En este nivel consolidado, las buenas prácticas se mantienen en los

momentos de estrés. Están definidos los productos a realizar. Se definen hitos para la revisión de los productos.

3- Definido: La organización entera participa en el proceso eficiente de proyecto software. Se conocen de antemano los procesos de construcción de software. Existen métodos y plantillas bien definidas y documentados. Los procesos no solo afectan a los equipos de desarrollo sino a toda la organización relacionada. Los proyectos se pueden definir cualitativamente.

4- Gestionado Cuantitativamente: Se puede seguir con indicadores numéricos (estadísticos) la evolución de los proyectos. Las estadísticas son almacenadas para aprovechar su aportación en siguientes proyectos. Los proyectos se pueden medir cuantitativamente.

5- En Optimización: Con base en criterios cuantitativos se pueden determinar las desviaciones más comunes y optimizar procesos. En los siguientes proyectos se produce una reducción de costes gracias a la anticipación de problemas y la continua revisión de procesos conflictivos.

### Área de Procesos: Medición y Análisis de CMMI

El modelo CMMI establece un conjunto de prácticas o procesos clave agrupados en Áreas de Proceso. Para cada área de proceso define un conjunto de buenas prácticas relacionadas que son ejecutadas de forma conjunta para conseguir un conjunto de objetivos. Las áreas de proceso son 25 y ayudan a mejorar o evaluar CMMI.

Una de las áreas de CMMI que está muy ligada al tema de la presente investigación es el área de Medición y Análisis. El objetivo de esta área para CMMI es desarrollar y sostener una capacidad de medición que sea usada para ayudar a las necesidades de información de la gerencia. Los datos tomados para la medición deben estar alineados con los objetivos de la empresa para proporcionar información útil a la misma.

Se ha de implantar un mecanismo de recogida de datos, almacenamiento, medición y análisis de los mismos de forma que las decisiones que se tomen puedan estar basadas en estos datos.

En esta área se establecen metas específicas para tener un orden de las actividades que se realizarán para la medición y el análisis. Además definen prácticas y subprácticas para especificar entre otras cosas, la forma de recolección de los datos y análisis de los resultados.

### 1.9. La calidad en la UCI

La UCI es un centro productivo, cuya misión es producir software y servicios informáticos a partir de la vinculación estudio – trabajo como modelo de formación. Es considerada la mayor organización productora de software en el país.

En la actualidad el centro está acometiendo un proyecto de mejora de sus procesos basado en el modelo CMMI y con la contratación de los servicios de consultoría del SIE Center (Software Industry Excellence Center) del Tecnológico de Monterrey. (26)

El proceso de mejora está encaminado a que la Universidad alcance en el 2010 una certificación internacional del nivel 2 del modelo CMMI. Hecho que la convertiría en la primera empresa cubana certificada con este modelo. (26)

CMMI es un modelo de referencia para el crecimiento de capacidades y madurez, que se enfoca tanto en procesos de Administración como de Ingeniería de Sistemas y Software. Con su instauración se espera alcanzar beneficios como:

- ✓ Calendarios y presupuestos predecibles en los proyectos.
- ✓ Mejora del ciclo de vida dentro del desarrollo de software.
- ✓ Mayor productividad.
- ✓ Mayor calidad de los productos y servicios que ofrece la universidad a sus clientes y por ende la satisfacción de los mismos.
- ✓ Mejorar la moral del personal que labora en el centro.

El servicio de consultoría que ofrece el SIE Center permite: (26)

- ✓ Ayudar a la UCI a revisar su estrategia de mejora de procesos de software, para asegurar que su organización está basada en procesos y con un programa de mejora continua alineado con sus objetivos de negocio.
- ✓ Ayudar a la UCI a establecer las bases y fundamentos para seguir mejorando sus procesos y fortalecer su cultura de calidad en el desarrollo de software

- ✓ Alinear los procesos de desarrollo de software con los principios y requisitos del modelo CMMI, estableciendo planes de mejora con los que la organización oriente sus procesos hacia la consecución de sus metas.

Actualmente en la UCI, se encuentran definidos y en pilotos cinco procesos: Aseguramiento de la Calidad a Procesos y Productos (PPQA), Administración de Requisitos (REQM), Planeación del Proyecto (PP), Monitoreo y Control del Proyecto (PMC) y Medición y Análisis (MA). En definición se encuentran los procesos de Administración de Acuerdos con Proveedores (SAM) y Administración de la Configuración (CM) (26)

En la búsqueda de la calidad de sus productos, esta institución lleva a cabo la definición de procesos, siguiendo las especificaciones de metodologías, estándares y modelos de desarrollo de software. Brindando además, asesorías, entrenamiento, métodos de medición y servicios de verificación-validación a las diferentes entidades.

El centro cuenta con una dirección general, la subdirección de desarrollo; dentro de la cual se agrupan, el grupo de normalización y métricas, el grupo de auditorías y revisiones y el departamento de pruebas; además de la subdirección de formación e investigaciones. Lo dicho se puede apreciar claramente en la siguiente Figura #6.

### **Figura 8. Estructura de la Dirección de Calidad en la UCI**

#### **1.10. DATEC (Centro de Tecnologías de Datos)**

La UCI cuenta con varios centros dedicados a la producción de software, en los cuales se llevan a cabo diversos procesos en busca de producir software con la mejor calidad requerida. Uno de dichos centros es el de tecnologías de datos (DATEC).

DATEC es un centro especializado en tecnologías de bases de datos con prestigio nacional e internacional. Alcanza prestigio internacional a mediano y largo plazo, su visibilidad se refleja en los proyectos y contratos que establece, la calidad de sus producciones y las publicaciones científicas de alto nivel que desarrolla. Mantiene estrecha colaboración con importantes empresas, centros de investigación y desarrollo de alto nivel así como con comunidades internacionales de desarrollo.

El centro se encarga de proveer soluciones integrales y consultorías relacionadas con tecnologías de bases de datos y análisis de información. Además realiza nuevas soluciones de bases de datos, de procesamiento y representación de la información a partir del desarrollo de proyectos de I+D (enfoque a la soberanía tecnológica). Como cualquier otro centro de producción de software, también debería utilizar para lograr la calidad de sus productos modelos, estándares y además métricas que le ayuden al desarrollo de productos de software con buena calidad.

En DATEC, para el desarrollo de sus proyectos utilizan diferentes tecnologías de prestigio internacional, alguna de las cuales le agregan funcionalidades para satisfacer las necesidades propias del centro en el desarrollo de las soluciones que realizan. Entre estas tecnologías se destacan:

- ✓ Los gestores de base de datos: generalmente PostgreSQL y Oracle.
- ✓ Los Sistemas operativos: Windows y Linux con sus versiones.
- ✓ Herramientas para Inteligencia de Negocios (BI): Suite de Pentaho.
- ✓ Framework de desarrollo: Simphony.
- ✓ Herramienta de mapeo de base de datos: Doctrine
- ✓ Herramienta de replica de datos: Slony.
- ✓ Herramienta de clustering: PG pool II y Heart Theat.
- ✓ Herramienta de monitoreo: Nagios.
- ✓ Metodología para el desarrollo de soluciones de almacenes de datos e inteligencia de negocios: Adaptación del Kimball
- ✓ Controlador de versiones: Subversion con los clientes Tortoise para Linux y RapidSVN.

### 1.11. Conclusiones del capítulo

En este capítulo se hizo un estudio de los conceptos asociados al problema planteado, lo cual posibilita un mejor entendimiento de los procesos que se llevan a cabo en la Gestión de Calidad. Se realizó un análisis profundo de las tendencias actuales en el mundo, los conceptos básicos de calidad y la

confiabilidad en específico, así como las asociaciones y centros de investigación dedicados a dichos temas. Además de las normas y estándares que definen una guía para llevar a cabo dichos procesos.

## 2. CAPÍTULO 2. PROPUESTA DE LA INVESTIGACIÓN

### 2.1. Introducción

En este capítulo se hace una caracterización de la situación actual existente en DATEC. Para lo cual fue necesario aplicar el método empírico, la entrevista. Además se hace un análisis en función del contenido de las entrevistas realizadas. Por otra parte se hace referencia a algunos de los tipos de resultados científicos que existen, con el objetivo de explicar el por qué del escogido para la propuesta. Y por último se describe detalladamente el modelo propuesto, explicando su estructura y definiendo cada una de las partes del mismo.

### 2.2. Caracterización de la situación existente en DATEC

La estructura del centro está dividida en líneas de desarrollo, las que a su vez se dividen en grupos de trabajo. A continuación se les presenta la estructura del centro por líneas de desarrollo y grupos de trabajo.

#### Figura 9. Líneas de Desarrollo y grupos de DATEC

- ✓ **Línea de Almacenes de datos:** Es la encargada de desarrollar soluciones que sirvan como apoyo a la toma de decisiones estratégicas.

Con los grupos de trabajo:

- **Análisis:** se centra en realizar el análisis de las soluciones de Almacenes de Datos y BI controlando el cumplimiento de los requisitos durante el ciclo de vida de la solución.
- **BI (Business Intelligent):** encargado de construir la capa de visualización de los almacenes de datos.
- **Almacenes:** encargado de realizar el repositorio de datos.
- **ETL (extracción, transformación y carga de datos):** responsable de integrar los datos existentes en las distintas fuentes y llevarlos hasta el repositorio de datos creado por el grupo de almacén.

- ✓ **Línea de Soluciones Integrales:** con la misión de desarrollar varios productos de soluciones integrales y/o herramientas.

Con los grupos de trabajo:

- Análisis: encargado de identificar los requerimientos de la solución y validar que se cumplan los requisitos con la calidad requerida.
  - Arquitectura: encargado de establecer la arquitectura de los sistemas, diseñar los componentes del sistema y desarrollar los componentes más importantes.
  - Desarrollo: encargado de implementar todos los componentes de los productos y documentar todas las clases que se desarrollan.
- ✓ **Línea de PostgreSQL:** con la misión de contribuir a la soberanía tecnológica potenciando tecnologías de bases de datos libres, tomando como base a PostgreSQL; además de proveer soluciones integrales y consultorías relacionadas con la explotación de PostgreSQL.

Con los grupos de trabajo:

- Soluciones de clústeres de altas prestaciones para PostgreSQL: encargado de dar soluciones de clústeres de alta disponibilidad para PostgreSQL.
- Soluciones de replicación de datos: se encarga de brindar servicios de implantación y transferencia de soluciones de réplicas de datos para servidores PostgreSQL.
- Personalización de PostgreSQL: encargado actualmente de la elaboración de una propuesta del PostgreSQL Empresarial Cubano.
- Personalización del Redmine: se encarga de dar mantenimiento al Redmine como herramienta de gestión de proyectos. En este grupo además se le da seguimiento al Portal de la Comunidad Técnica Cubana de PostgreSQL.

### 2.2.1. Entrevista

Teniendo en cuenta la necesidad de conocer la situación actual del centro es comprensible que en esta etapa de la investigación se aplique algún método empírico para alcanzar el objetivo del trabajo y para obtener las características del centro que no son observables directamente e ir más allá de las

características superficiales de la realidad. Se realizaron entrevistas al personal de las líneas, para poder hacer una recopilación de la información necesaria. Para obtener información del centro, sobre sus líneas, sus grupos, la forma en que se trabajan los proyectos, de la manera en que está definida la organización del personal, así como los roles de cada persona, se determinó llevar a cabo diferentes entrevistas para la recopilación de los datos requeridos. Todo esto se realiza para con el propósito de hacer una valoración de los elementos que se tendrán en cuenta para el desarrollo del modelo que se propondrá.

Del centro se entrevistó a varios especialistas:

- ✓ **Michael González Jorin:** Asesor de Calidad.
- ✓ **Asnioby Hernández López:** Jefe de la línea de Almacenes de Datos.
- ✓ **Yudisney Vázquez Ortiz:** Jefa de la línea de PostgreSQL.
- ✓ **Manuel Vázquez Acosta:** Jefe de la línea de Soluciones Integrales.
- ✓ **Anisley Delfino Rodríguez:** Jefa del grupo de BI, de la línea de Almacenes de Datos.
- ✓ **Mabel Medina Rodríguez:** Jefa del grupo de Análisis, de la línea de Almacenes de Datos.
- ✓ **Doris Medina Mustelier:** Jefa del grupo de ETL, de la línea de Almacenes de Datos.
- ✓ **Yonelbys Iznaga González:** Especialista de ETL, del grupo de ETL de la línea de Almacenes de Datos.
- ✓ **Adrian Misael Peña Montero:** Jefe del grupo de soluciones de clúster de altas prestaciones de la línea de PostgreSQL.
- ✓ **Ernesto Herrera:** Arquitecto de datos, de la línea de Soluciones Integrales.
- ✓ **Roberto Sarmiento:** Arquitecto de datos, de la línea de Soluciones Integrales.

En el cuestionario se hacen preguntas enfocadas a varios de los temas que dan origen a esta investigación.

### 2.2.2. Análisis de las entrevistas

A cerca del tema de las **métricas del software**: algunos de los entrevistados respondieron afirmativamente sobre el uso de las métricas en el centro, pero en su mayoría las respuestas sobre el uso de métricas en diferentes etapas de la construcción del software fueron negativas, pues plantearon que no tenían métricas definidas para ser aplicadas en los proyectos. Se llega a la conclusión, en las entrevistas, de que en DATEC no se utilizan métricas en el desarrollo de sus proyectos, las mismas no han sido definidas o establecidas por el centro o por las líneas.

Con respecto a los **atributos de confiabilidad**, se hicieron varias preguntas para tocar de forma particular cada uno de estos factores, y fueron agrupadas las respuestas en cuanto al nivel de semejanza que presentan los factores entre si.

- ✓ **Precisión y Exactitud**: aspectos estrechamente relacionados con las respuestas que debe dar el software una vez implementado y las pruebas que se le hacen al mismo para comprobar dichas respuestas, así como con la cantidad de pruebas y la valoración de las respuestas obtenidas; se formularon 10 preguntas, de las cuales se obtuvo un 70 % de coincidencias en las respuestas.

Teniendo en cuenta las respuestas obtenidas de estas preguntas y según las opiniones de los entrevistados, se puede decir que en el centro no se hacen pruebas con el objetivo de medir estos parámetros. Además no tienen en cuenta cuántas pruebas le aplican, no llevan un control de las mismas, tampoco realizan operaciones con las respuestas arrojadas, no se tiene conocimiento del nivel de variación de dichas respuestas, no son registrados estos valores en algún repositorio o documentación, solo se observan y mediante esta observación son tomadas las decisiones que respectan a dichas respuestas.

- ✓ Respecto a la **Tolerancia ante fallos, Madurez y Recuperabilidad** que debe tener un producto: refiriéndose estos factores fundamentalmente a los errores, defectos y fallas que puedan presentarse durante el desarrollo del proyecto y en su culminación en si; se realizaron 21 preguntas, coincidiendo las respuestas en un 80%.

Según lo captado una vez hechas las entrevistas a estas personas. Las respuestas de los entrevistados coinciden en que los errores generales más comunes que se presentan en la realización de un software son los de formato y ortografía, errores de validación, de inconsistencia con los tipos de datos, también la omisión de un requisito; y errores que tienen que ver con los temas de integración. Se supo además, que por su parte al grupo de BI se le puede presentar fallas como que el sistema no

muestre todo lo que en los requerimientos se definió que debía mostrar, además de que en ocasiones las interfaces no tienen un buen diseño óptico; estas fallas pueden estar dadas por la falta de control de los datos requeridos por parte del grupo de almacenes. Además no tienen en cuenta el tema de que el software continúe funcionando en caso de alguna falla, simplemente cuando detectan que falló, tratan de localizar el defecto y lo corrigen, ni siquiera llegan a analizar el error. De tal situación se dan cuenta con los casos de prueba que realizan durante la implementación. Por otra parte no registran dichos errores ni defectos, solo las fallas y en la fase de liberación del producto en el documento de No Conformidades. No tienen en cuenta tampoco un control de la cantidad de veces que les fallan los productos. Una vez que estos fallan la recuperación la hacen manual, pues no tienen preparados como se decía anteriormente los software para su recuperación automática. La información de todo lo relacionado con los software la guardan en Backups que poseen los servidores y en caso de fallar algunos de ellos, simplemente vuelven a montar el servidor y recuperan toda la información.

- ✓ En el caso de la **Modularidad**: relacionado con la forma en que está estructurado el desarrollo de los proyectos en el centro, se realizaron 14 preguntas, de las que fueron respondidas aproximadamente el 85.6% del total, por los entrevistados.

En cuanto a este aspecto se puede decir que en el centro desarrollan utilizando las dos formas: por módulos y por componentes, pues algunos de los entrevistados plantearon que por componentes y otros que se tienen en cuenta las dos formas.

Se plantea que el desarrollo se divide por módulos, estos a su vez por componentes y luego se integran para tener cada uno de los módulos listos. No se desarrollan los módulos como si fueran proyectos individuales, el control de si un módulo es más complejo que otro solo lo saben a apreciación y le hacen pruebas a los módulos para ver si está bien su funcionamiento o cumple con lo requerido. También se observó que los módulos los desarrollan en paralelo y separados al mismo tiempo, además de que todos están muy integrados. Creen también que el software sería más confiable si se desarrolla de esta manera y de fallar un módulo, solo afectaría el sistema completo en dependencia de que módulo sea y cuan importante sean sus funcionalidades en el proceso.

En caso de las respuestas de que desarrollaban por componentes, el argumento fue que el proceso de desarrollo del centro está basado en líneas de desarrollo, y las líneas se desarrollan por componentes para después integrar todos esos componentes. Plantean que no son desarrollados los componentes como si fueran proyectos individuales y no los definen ellos, lo que hacen en los componentes son

soluciones específicas que ya vienen definidas. Informando además que estos componentes están fuertemente ligados unos con otros, creen bien esta forma de desarrollar y hacen saber que de fallar uno de los componentes, esto afectaría el sistema completo. Para desarrollar cada uno de los componentes se rigen por las pautas definidas por el grupo y la línea, por ejemplo el grupo de BI de la línea de almacenes de datos utiliza una metodología nombrada Metodología para el Desarrollo de Soluciones de Almacenes de Datos y BI del inglés, la cual es una adaptación de la metodología de Kimball.

- ✓ Para la **Simplicidad, Consistencia y Completitud**: relacionado en gran medida con la implementación de las funciones, las líneas de código; cuan sencillo, compactado y completo llegan a estar el código y las funciones en general; se crearon 14 preguntas las cuales fueron respondidas el 78.6% del total de estas.

De las respuestas de la mayoría de los entrevistados se puede deducir que aunque en el centro no utilizan ningún estándar de codificación, se organizan a la hora de implementar y llegan a acuerdos de cómo hacer las cosas, pero no tienen establecido ninguna pauta al implementar. A lo que si le ven mucha importancia es a la cuestión de la reutilización del código, es por lo que hacen que sus funciones puedan ser reutilizadas, implementándolas de la forma más comprensible y sencilla posible. Además, revisan las tareas de cada uno de los desarrolladores verificando que estén todas las funciones requeridas implementadas y si están bien o no, lo cual no se registra, simplemente verifican y en caso de que no estén todas lo devuelven y el encargado tiene que realizar todo lo que tenía que hacer; llevando con la herramienta de gestión de proyectos que utilizan, el control de este asunto. Por otra parte todos plantearon que el código fuente a parte de estar montado en la aplicación que realizan, se guarda en el servidor de ficheros que tienen en el repositorio donde guardan toda la información del proyecto, y que dicha información solo está accesible, con permisos de lectura y escritura, para los integrantes del grupo de trabajo, y permisos de solo lectura para los integrantes de la línea completa.

- ✓ A cerca de la **Conformidad con la confiabilidad**: lo cual se refiere a la capacidad que adquieren sus productos para adherirse a normas, regulaciones o leyes; se formularon 5 preguntas, las cuales fueron respondidas todas, representando esto el 100% de las preguntas realizadas a los entrevistados.

Según la información que se obtuvo al respecto, se identificó que en el centro le aplican a sus proyectos, como normas, entre otros, los lineamientos de calidad de la UCI, además de que sus

productos cuentan con mucha capacidad para adaptarse a cualquier norma o ley relativa a la confiabilidad. Una de estas leyes es que tratan de hacerlo todo en software libre y adaptándose además a la necesidad de cada cliente. El control de este factor lo llevan mediante las revisiones, auditorías y las revisiones del código como tal. Y creen que este indicador les afectaría, en caso de que los productos no contaran con dicha capacidad, en la calidad en general de los mismos, en el tiempo y costo además.

- ✓ Sobre el tema de las **tecnologías**: se presentaron 25 preguntas, enfocadas a conocer a cerca de todas y cada una de las tecnologías que usan en DATEC, además de por qué y para qué la usan, entre otros aspectos. De estas preguntas exactamente fueron respondidas el 84% por los entrevistados:

De las respuestas dadas por los entrevistados se obtuvo la siguiente información.

En el centro usan los gestores de base de datos MySQL, Oracle y PostgreSQL, pero generalmente usan el último (PostgreSQL), por las facilidades que presenta, de licencia de comercialización, que es libre, más confiable y estable, además porque está definido por la UCI y el centro en específico. Plantean que los software podrían sufrir fallas a causa de la mala configuración, instalación o carga inicial del gestor que estén utilizando y que se les han presentado problemas de rendimiento también por esta causa, y podrían sufrir además, riesgos de pérdida de información también debido a la mala configuración del gestor. En estos temas de los fallos y más relacionados con los gestores no tiene mucha experiencia, pues el centro es muy joven, solo ha desarrollado dos productos completos, los demás están simplemente pensados o en términos de negocio aún. En este caso cuando detectan una falla lo registran como en las situaciones anteriores en el documento de no conformidad, solo si es el proceso de liberación y realizan el mismo procedimiento de analizar cual fue el defecto y tratar de arreglarlo.

Para el control de versiones utilizan el Subversion, con los clientes Tortoise para Windows y RapidSVN para Linux, por política establecida por la universidad y el centro en cuestión. No han tenido ningún tipo de falla a causa de este software y les parece bastante seguro y confiable. La información no la guardan en un solo lugar, pues poseen el servidor general, además de servidores de respaldo y copias de seguridad en cada uno de ellos, además de en las estaciones de trabajo, por lo que no corren riesgo de pérdida de información.

Usan los sistemas operativos Windows XP y Ubuntu de Linux principalmente para los servidores. Han tenido fallos los software por temas de configuración de la impresora, controlando este tipo de fallo con la instalación del controlador de la impresora. Pero se describe que los fallos que dependen del sistema operativo sobre el que estén trabajando no ocurren con mucha frecuencia.

También usan la suite de herramientas que incluye Pentaho, de ella usan para el desarrollo de soluciones de BI y de almacenes de datos Pentaho Business Intelligent Cumunitie Edition 3.5 y Pentaho Data Integration para los temas de extracción, transformación y carga de datos. Para estos mismos temas pero en caso de propietarias usarían para BI, Oracle BI Enterprise Edition Plus y para ETL, Oracle Warehouse Builder. Sobre ésta herramienta plantean que hasta el momento no se han encontrado en la situación de un fallo que haya dependido de dicha herramienta, pero que el software no está exento de fallas cuando precisamente no se lleva ningún control con esto, ni se ha hecho nunca ningún estudio al respecto.

Sobre las metodologías a utilizar para el desarrollo de cada proyecto, los entrevistados plantearon que cada departamento define dentro de él la metodología a utilizar para la construcción de sus productos, o sea no es una especificación del centro usar una determinada, el departamento en específico es quien decide cual usará. Mayormente usan una adaptación de la metodología Kimball dentro de la Línea de Almacenes de Datos, y la metodología OpenUp en las otras líneas. Estas metodologías han sido adaptadas por el equipo de desarrollo de los diferentes departamentos para así poder dar cumplimiento a las necesidades específicas que se plantean. Los software que desarrollan no son afectados directamente a causa de la metodología que utilizan, si el mismo falla es porque se ha introducido un defecto en la etapa de desarrollo, el cual se detectaría y controlaría en la etapa de pruebas, pero esto no es propiciado directamente por las metodologías.

- ✓ Teniendo en cuenta el tema de **Calidad** en el centro y según las respuestas dadas por los entrevistados, se puede decir que dentro de los proyectos del centro son aplicados por el departamento de Calidad un conjunto de criterios y normas generales, por las cuales se rigen en el momento de analizar la calidad de los proyectos que se van desarrollando. En cuanto a la calidad de los modelos de diseño y del código fuente este departamento no realiza revisiones técnicas. Tienen bien definidas las políticas de calidad y las aplican en todas las áreas dentro de las cuales se encuentran: el área de atención al cliente y la de seguridad informática, entre otras. Poseen además, un sistema estructurado de Calidad, dentro del cual se encuentran como puntos fundamentales las políticas y regulaciones donde se incluyen los lineamientos, los procesos donde

se incluyen el expediente de proyecto y la documentación cliente suministrador, además de las funciones y estructura.

### 2.2.3. Resumen de las entrevistas

Dada la situación actual existente en DATEC, se ha apreciado que a pesar de todos los esfuerzos realizados por el centro en función del logro del desarrollo de los productos con la mayor calidad posible, es necesario aplicar algún método científico para la medición de la confiabilidad en dichos productos. Por tanto se ha evidenciado que es preciso desarrollar una herramienta científica probada que permita medir la confiabilidad en estos productos. Dado el análisis realizado de las entrevistas, se determinó que los factores en los que se debe profundizar en el desarrollo del modelo deben ser la exactitud, la precisión, la tolerancia ante fallos, la madurez, la simplicidad, la recuperabilidad, completitud, consistencia modularidad. Estos elementos han sido seleccionados como los más críticos ya que carecen de una forma concisa para su correcta revisión dentro de los proyectos, lo cual puede provocar que carezcan de un nivel elevado de confiabilidad. Las pruebas se realizan solo en la fase de liberación y con el único objetivo de verificar si la respuesta que obtienen es la esperada.

### 2.3. Análisis de los diferentes resultados científicos

Como consecuencia lógica de la investigación se obtienen resultados que son una cuestión relevante que el investigador se propone alcanzar para completar su objetivo y al mismo tiempo dar solución a su problema científico.

Dentro de los diferentes resultados científicos de carácter práctico que se pueden obtener en una investigación se encuentran las propuestas de:

- ✓ metodologías
- ✓ sistemas
- ✓ estrategias
- ✓ otros.
- ✓ modelos

Las **metodologías** van a estar definidas como un conjunto de métodos o procedimientos que se estudian para luego aplicarlos en orden a la consecución de determinados objetivos para perfeccionar o transformar un objeto. (27)

Tendremos como **estrategia** a un proceso de organización que sea coherente, unificado, integrado, sistémico, transformador y flexible que debe partir de un diagnóstico en el que se evidencie un problema a resolver, la proyección y ejecución de acciones inmediatas e intermedias, que de manera progresiva y coherente permita alcanzar los objetivos propuestos. (27)

El **sistema** es aquella combinación de elementos cuya integración sea la que produzca un resultado superior a la aplicación aislada o a las combinaciones parciales de sus componentes. Es una relación entre ellos en lo que cada uno enriquece a sí mismo y acentúa la acción de los demás. (27)

### **Modelo:**

“Por modelo se entiende un sistema concebido mentalmente o realizado en forma material, que, reflejando o reproduciendo el objeto de la investigación, es capaz de sustituirlo de modo que su estudio nos dé nueva información sobre dicho objeto”. (V.A. Shtoff.) (27)

Se requiere de la elaboración de un modelo como resultado científico de la investigación, ya que es necesario conocer la esencia del centro, pero debido a la amplitud, complejidad y diversidad de información contenido en este se necesita buscar un medio auxiliar que posibilite, en un primer momento, su simplificación y que, posteriormente, pueda servir como instrumento para la predicción de acontecimientos que no han sido observados aún. Todo esto siempre en correspondencia con el fin de saber como medir la confiabilidad de sus productos.

### **2.4. El modelo como alternativa a seguir**

Un modelo de medición de software es una descripción de todos los elementos de interés en el proceso de medición de un proyecto software. Dichos elementos incluyen ¿qué medir? (atributos, entidades y categorías de entidades), ¿cómo? (métodos de medición, formas de medir), ¿por qué? (necesidades de información), resultados de las mediciones y los criterios de análisis. (28)

El modelo tiene la posibilidad de otorgar al proceso de trabajo o de investigación la información necesaria y susceptible de comprobación práctica, teórica o experimental. La existencia de relaciones, reglas prácticas, concepciones y fundamentos teóricos precisos para describir así como poder pasar de la información que ofrece el modelo a la información sobre el objeto o proceso modelado y viceversa.

El desarrollo de un modelo posee las siguientes características: (27)

- ✓ Se explica el proceso objeto de estudio con los fundamentos teóricos necesarios y suficientes, surgidos en el proceso de conocimiento, relacionados de forma esencial y simplificada, a lo que contribuye decisivamente su capacidad ilustrativa y traslativa que caracteriza a este proceso.
- ✓ El modelo en estas condiciones sirve de medio para la construcción de una teoría nueva, una concepción o fundamentación según las características del fenómeno estudiado. Tiene un carácter sintético, intensivo ya que no describe una estructura concreta sino, mediante un proceso de abstracción, se aparta de la realidad perceptible y subraya frecuentemente hasta el extremo lógico, cierto atributo importante para la solución del problema.
- ✓ Una vez concretado el modelo, uno de los fundamentos esenciales está en su aplicación práctica.

### **CMMI como guía a seguir**

Para la realización del modelo a proponer se escoge el modelo CMMI como guía, por ser estructuralmente uno de los modelos más completos y conocidos a nivel internacional. Este modelo contiene una serie de prácticas que sirven para determinar el grado de capacidad y madurez en una organización para su proceso de desarrollo de software.

### **2.5. Modelo de Medición de Confiabilidad y análisis de las tecnologías**

Tomando como referencia la estructura del modelo CMMI se decide elaborar el modelo a proponer, con el objetivo de desarrollar un método auxiliar que permita medir la confiabilidad de los productos de DATEC, debido a la situación existente en el centro, ya que no cuentan en el mismo con ninguna herramienta que les permita tener un conocimiento del nivel de confiabilidad que poseen sus productos. En el modelo se tocarán características específicas con las que debe cumplir un software para que sea confiable y de las cuales en el centro no se llevan un control adecuado, además de algunos factores asociados a la confiabilidad que de no ser tenidos en cuenta también dificultarían la confiabilidad del software.

#### **2.5.1. Análisis de las tecnologías**

Antes de presentar el modelo fue necesario realizar un análisis profundo de las tecnologías más importantes utilizadas en el centro. Con el objetivo de delimitar y tener en cuenta cuál de estas afectaría o influiría en la confiabilidad del software que se esté desarrollando. Según las entrevistas a los especialistas del centro y los estudios realizados del tema se puede plantear que los software no

están expuestos a riesgos asociados a la confiabilidad, como consecuencia de las tecnologías que utilizan. Tal análisis se hizo teniendo en cuenta los atributos con los cuales se define la confiabilidad. Cualquier riesgo que corra el software estaría dado por la configuración de estas tecnologías, además de su uso, de los cuales es responsable la persona que trabaje con la misma. Por otra parte existen algunos factores de calidad sin los que tampoco fuera posible un software confiable y con los que están relacionadas de forma más directa éstas tecnologías. Se considera que la utilización de las tecnologías pudiera influir en algunos de los factores, los cuales conforman lo que se conoce como RAS-IS.

A continuación se presenta una tabla donde se relacionan las tecnologías analizadas con los factores del RAS-IS.

Donde: F (Fiabilidad), D (Disponibilidad), Sr (Servicialidad), I (Integridad), Sg (Seguridad).

### Figura 10. Relación de las tecnologías con los factores del RAS-IS

**Linux:** Por ser uno de los sistemas operativos que proporciona más fiabilidad en los entornos de trabajo, es utilizado en el centro para los servidores. Debido a la importancia que se le confieren a los servidores por la información que almacenan se considera que este sistema operativo puede estar ligado a los riesgos que atenten contra la disponibilidad y servicialidad de dicha información.

**Windows:** Es el sistema operativo más empleado a nivel mundial, debido, entre otras cosas, a su entorno amigable con el usuario inexperto. Por otra parte este sistema operativo no es uno de los más seguros. Por lo que con el uso del mismo expondría el desarrollo de los productos a riesgos de seguridad, atentaría contra la integridad del software y de paso lo haría menos confiable.

**Oracle:** Es uno de los sistemas gestores de bases de datos más completos, destacando: soporte de transacciones, estabilidad, escalabilidad y soporte multiplataforma. Ha sido criticada por algunos especialistas la seguridad de la plataforma, y las políticas de suministro de parches de seguridad, aunque los creadores han ido realizando tratamientos de estas vulnerabilidades. Este sistema gestor de base de datos entonces se verá vinculado además a riesgos de Servicialidad por el nivel de relevancia de la información que maneja.

**PostgreSQL:** Este gestor es el más utilizado en el centro para la administración de las bases de datos, por ser un gestor bien probado que contiene más de 5 lenguajes de programación procedurales. Este gestor soporta un número limitado de conexiones concurrentes y rechaza las que superan dichas cifras.

Por tales motivos se considera que los software desarrollados sobre esta herramienta, se pueden ver afectados en los factores de servicialidad, además podrían estar expuestos a riesgos que atenten contra la integridad de los datos, por el continuo manejo de los mismos.

**Suite Pentaho:** Esta suite de herramientas se utiliza fundamentalmente para el manejo de datos en los grupos de BI y ETL. Algunas de estas herramientas son: Pentaho Business Inteligent Cumunitie Edition 3.5, Oracle BI Enterprise Edition Plus, Oracle Warehouse Builder y Pentaho Data Integration. Esta última la utilizan en el grupo de ETL y está enfocada a los temas de extracción, transformación y carga de datos. Es por las características que presentan estas herramientas y para lo que están diseñadas que se considera que los software que la utilizan podrían tener problemas de integridad de los datos, lo cual se corresponde con el factor integridad.

**Symphony:** Es un framework de desarrollo, una biblioteca de clases cohesivas escritas en PHP. Proporciona una arquitectura, componentes y herramientas para que los desarrolladores construyan aplicaciones web complejas con mayor rapidez. Elegir Symphony le permite liberar sus aplicaciones más temprano, acogerlas y escalarlas sin problemas, y mantenerlas por tiempo prolongado sin que le de sorpresas. Según las características y del framework de desarrollo se plantea que los software podrían estar expuestos a riesgos relacionados con la fiabilidad y la disponibilidad del software, pues en esta plataforma es donde mayormente se desarrollan todas las funcionalidades concretas del proyecto.

**Doctrine:** Potente y completo sistema ORM (Object Relational Mapper, Mapeo objetos relacional). Entre muchas otras cosas tiene la posibilidad de exportar una base de datos existente a sus clases correspondientes y también a la inversa, es decir convertir clases (convenientemente creadas siguiendo las pautas del ORM) a tablas de una base de datos. Por otro lado, como la librería es bastante grande ésta tiene un método para ser compilada al pasar a producción. Se considera que en los factores en los que el software podría estar afectado teniendo en cuenta esta herramienta es en los de fiabilidad e integridad, debido al manejo continuo que tiene con los datos, así como con lo de las exportaciones de las bases de datos.

**Slony:** Herramienta para replicar una base de datos en PostgreSQL. Posee una gran ventaja y es que el servidor primario manda las actualizaciones en tiempo real a la base de datos esclava y en caso de que falle el servidor primario, continuamos trabajando con la base de datos esclava. Es por lo que no corren riesgos los software de disponibilidad pero por otra parte se considera que en caso de tener

algún problema con la herramienta en el software se vería afectada la integridad de los datos, así como la seguridad, debido a la transmisión y replicación de la información.

**Pgpool II:** Herramienta de clustering que trabaja entre los servidores de PostgreSQL y un cliente de base de datos PostgreSQL. Por su gran asociación con el tema de los servidores y los servicios para que las bases de datos funcionen, se considera que los software podrían verse afectados de alguna manera en los factores de seguridad y servicialidad. Pues la información manejada en esta base de datos es muy importante y a la que hay que dirigir mucho esfuerzo por mantener segura.

**Nagios:** Herramienta utilizada para monitorear servidores, dispositivos de comunicación y aplicaciones en forma remota desde cualquier estación de trabajo a través de un navegador Web. En definitiva obtendrá un detallado conocimiento de la red. Y podrá evaluar los recursos disponibles de su infraestructura de comunicaciones. Por las características expuestas se plantea que algunos de los riesgos que atenten contra la confiabilidad del producto utilizando esta herramienta podrían estar vinculados al factor de seguridad.

### 2.5.2. Estructura del modelo

En la presente sección se explicará la estructura general del modelo propuesto, explicando además cada una de las partes que lo conforman para luego entrar en los detalles o las descripciones de cada una de ellas.

#### Figura 11. Estructura del Modelo de Medición de Confiabilidad

**Categorías:** Creadas para agrupar los atributos de confiabilidad según sus fines.

**Atributos de Confiabilidad:** Conjunto de atributos que definen la confiabilidad, que se deben tener en cuenta para conseguir el desarrollo de un software confiable.

**Factores Asociados:** Factores de calidad estrechamente vinculados con la confiabilidad, sin ellos tampoco es posible un software confiable.

**Fines Genéricos:** Objetivos generales que establecen lo que un producto debe alcanzar para lograr un nivel de confiabilidad en una categoría determinada.

**Fines Específicos:** Objetivos específicos que se aplican a un atributo de confiabilidad y señalan los elementos particulares que se deben cumplir para satisfacer el propósito de dicho atributo.

**Prácticas:** Actividades que se llevan a cabo para cumplir con los fines específicos a los cuales están asociadas.

**Subprácticas:** Especificaciones requeridas para que se puedan realizar las prácticas.

En el modelo se definen 9 atributos de confiabilidad los cuales se agrupan en 3 categorías atendiendo a su finalidad.

- ✓ Control de Respuestas.
- ✓ Gestión de Fallos.
- ✓ Monitoreo de Implementación.

Categorías	Atributos que incluye	Propósito
Control de Respuestas	<ul style="list-style-type: none"> <li>- Precisión</li> <li>- Exactitud</li> </ul>	Llevar un control de las respuestas resultantes de las pruebas realizadas. Realizar los cálculos pertinentes con la utilización de las métricas.
Gestión de Fallos	<ul style="list-style-type: none"> <li>- Tolerancia ante Fallos</li> <li>- Madurez</li> <li>- Recuperabilidad</li> <li>- Modularidad</li> </ul>	Aplicar las métricas que permitan gestionar las fallas del software.
Monitoreo de Implementación	<ul style="list-style-type: none"> <li>- Simplicidad</li> <li>- Completitud</li> <li>- Consistencia</li> </ul>	Llevar un control de la implementación de las funcionalidades del software.

**Tabla 1. Categorías y Atributos del modelo de medición de confiabilidad**

### Factores asociados a la Confiabilidad

Existen algunos factores de calidad asociados a la confiabilidad que combinados son llamados RAS-IS, sigla dada por el nombre de los cinco factores en ingles: Reliability (Fiabilidad), Availability (Disponibilidad), Serviceability (Servicialidad), Integrity (Integridad) y Security (Seguridad).

### 2.5.3. Descripción del modelo

Antes de comenzar con la descripción de los atributos, junto a las métricas asociadas a cada uno de ellos y el análisis de las mismas, se propone revisar el Anexo #7 donde encontrarán una plantilla en la que se plasma cada atributo y factor asociado a la confiabilidad junto a su definición y el resultado que se obtuvo luego del cálculo y análisis de las métricas. Dicho resultado estará expresado de la siguiente manera: 1 nivel Inicial, 2 nivel Definido y 3 nivel Optimizado. Estos resultados dependerán también del peso que se le asignen a cada uno de los atributos lo cual se explicará más adelante.

#### 2.5.3.1. Fines Genéricos y Específicos de cada atributo

A continuación se presenta una tabla en la que se describen las categorías, incluyendo los atributos que las conforman y los fines específicos para cada uno de ellos.

<b>Control de respuestas</b>	
<b>Fin genérico:</b> Lograr un alto nivel de precisión y exactitud de las respuestas del software.	
<b>Atributos</b>	<b>Fines Específicos</b>
<b>Precisión</b>	<ul style="list-style-type: none"> <li>- Verificar que las respuestas arrojadas por el software se parezcan a la anterior.</li> <li>- Controlar la variación de las respuestas del software.</li> <li>- Realizar los cálculos necesarios que proporcionen el grado de precisión del software.</li> </ul>
<b>Exactitud</b>	<ul style="list-style-type: none"> <li>- Verificar que en las pruebas realizadas al proyecto se obtenga la respuesta esperada.</li> <li>- Llevar un registro de cuanto varían las respuestas obtenidas de las pruebas en relación con la respuesta esperada.</li> </ul>
<b>Gestión de Fallos</b>	
<b>Fin genérico:</b> Llevar a cabo de una manera óptima el seguimiento de los fallos	

que se producen en el software.	
<b>Tolerancia ante fallos</b>	<ul style="list-style-type: none"> <li>- Verificar la capacidad del software de seguir funcionando en caso de un fallo.</li> <li>- Controlar los tipos de fallos encontrados.</li> <li>- Analizar los fallos en búsqueda de corregir los defectos que propiciaron que el software fallara.</li> </ul>
<b>Madurez</b>	<ul style="list-style-type: none"> <li>- Llevar un control de los fallos producidos en el software y sus causas.</li> <li>- Controlar que el software tenga la capacidad de evitar un fallo total.</li> <li>- Prevenir la presencia de fallos en el software.</li> </ul>
<b>Recuperabilidad</b>	<ul style="list-style-type: none"> <li>- Verificar que el software tenga la capacidad de restablecer un nivel de ejecución en caso de fallo.</li> <li>- Controlar que el software tiene la eficiencia de recuperar datos que han sido afectados en un fallo total.</li> </ul>
<b>Modularidad</b>	<ul style="list-style-type: none"> <li>- Verificar que el software tiene una correcta estructura de módulos.</li> <li>- Controlar que los módulos sean lo más independientes posible.</li> </ul>
<b>Monitoreo de Implementación</b>	
<b>Fin Genérico:</b> Lograr una mejor implementación del código, así como del cumplimiento de las funciones requeridas.	
<b>Simplicidad</b>	<ul style="list-style-type: none"> <li>- Lograr que la implementación sea lo más sencilla posible</li> <li>- Aplicar estándares de codificación para una mejor organización del código fuente.</li> <li>- Conseguir que se utilice la</li> </ul>

	reutilización de código para facilitar un desarrollo posterior.
<b>Compleitud</b>	<ul style="list-style-type: none"> <li>- Verificar que estén todas las funciones completamente implementadas.</li> <li>- Lograr que estén implementadas en su totalidad las funciones definidas en los requerimientos.</li> </ul>
<b>Consistencia</b>	<ul style="list-style-type: none"> <li>- Verificar que haya una correcta uniformidad en el trabajo.</li> <li>- Lograr que los requisitos funcionales se cumplan en el diseño y la implementación del producto.</li> </ul>

**Tabla 2. Fines genéricos y específicos para las categorías y los atributos**

Seguidamente aparecerá cada una de las prácticas que se proponen a aplicar que dan cumplimiento a los fines específicos descritos por cada atributo. En estas prácticas lo que se proponen son métricas. Algunas de estas métricas tomadas de bibliografías especificadas al final del documento y otras confeccionadas por las investigadoras. También obtendrán en este paso las subprácticas definidas para cada una de las prácticas. Lo que les ayudará en la realización de cada práctica.

### 2.5.3.2. Prácticas para cada uno de los atributos

#### ✓ Métricas para calcular la Precisión del software

##### 1. Proporción de las respuestas obtenidas

$X=A/B$       Donde:

A = número de respuestas obtenidas que se acercan a la anterior.

B = número total de respuestas obtenidas.

**Observaciones:** Proporciona el grado de precisión que puede tener un software dependiendo de las respuestas arrojadas luego de las pruebas.

#### ✓ Métricas para calcular la Exactitud del software

##### 2.1 Exactitud esperada

$X = 1 - (A/T)$  Donde:

A = número de casos encontrados con diferencias entre los resultados razonablemente esperados y aquellos resultantes más allá de lo permisible.

T = tiempo de operación.

**Observaciones:** Permite hacer un análisis de las diferencias entre los resultados actuales y los razonablemente esperados.

### 2.2 Variación de las respuestas del software

$Y=C/D$  Donde:

C = número de respuestas que se acercan a la respuesta esperada.

D = número total de respuestas obtenidas del software.

**Observaciones:** Proporciona conocimiento a cerca de cuanto varían o podrían variar las respuestas esperadas del software.

✓ **Métricas para calcular la Tolerancia ante fallos del software**

### 3.1 Anulación de fallas.

$X=A/B$  Donde:

A = número de fallas anuladas contra los casos de prueba del modelo de errores que casi causa la falla.

B = número de casos de pruebas ejecutados del modelo de errores que casi causa la falla durante la prueba.

### 3.2 Anulación de operaciones incorrectas (error operacional o proporción de error detectado en la entrada).

$X=A/B$  Donde:

A = número de errores detectados por el usuario en entradas o funciones.

B = número de operaciones o entradas erróneas del usuario durante la operación.

### 3.3 Anulación de interrupciones (proporción de anulación de interrupciones)

$X=1-(A/B)$  Donde:

A = número de interrupciones.

B = número de fallas.

**Observaciones:** La interrupción significa que la tarea ejecutada de cualquier usuario es suspendida hasta que el sistema reiniciado, o su control se pierden hasta que el sistema es forzado a apagarse.

✓ **Métricas para calcular la Madurez del software<sup>1</sup>**

### 4.1 Tiempo promedio entre fallas (TPF).

4.1.1  $X=A/B$  Donde:

A = tiempo de funcionamiento.

B = número de fallas descubiertas.

4.1.2  $Y=C/D$  Donde:

C = suma de intervalos de tiempo entre una falla ocurrida y la siguiente durante el tiempo de funcionamiento observado.

D = número de fallas ocurridas durante el tiempo de funcionamiento observado.

### 4.2. Resolución de fallas. Proporción de solución de fallas.

$X=A/B$  Donde:

A = número de fallas resueltas.

B = total de fallas descubiertas.

### 4.3 Defectos eliminados. Proporción de corrección de errores.

$Y=C/D$  Donde:

---

<sup>1</sup> Estas métricas de Madurez y Tolerancia ante fallos son tomadas del proyecto de grado "Automatización de los procedimientos para la recolección y consolidación de las métricas externas del producto según la norma ISO/IEC 9126:2001" de Andrés Camilo Bustamante Bedoya.

C = número de defectos corregidos.

D = total de defectos descubiertos.

#### 4.4 Intensidad de fallos totales contra casos de prueba

$X=A1/A2$  Donde:

A1 = número total de fallos totales detectados.

A2 = número de casos de pruebas ejecutados.

**Observaciones:** Permite determinar cuántos fallos totales fueron detectados durante un período de pruebas definido.

#### 4.5 Cobertura de las pruebas

$X=A/B$  Donde:

A = número de casos de pruebas que han sido realmente ejecutados, y que representan el escenario de operación durante las pruebas.

B = número de casos de pruebas a ejecutar requeridos para cubrir los requisitos.

**Observaciones:** Permite hacer un análisis de cuántos casos de pruebas requeridos han sido ejecutados detectados durante las pruebas.

#### ✓ Métricas para calcular la Recuperabilidad del software

##### 5.1 Tiempo medio de inactividad

$X=T/N$  Donde:

T = tiempo total de inactividad.

N = número de desastres observados.

**Observaciones:** Permite determinar cuál es el tiempo promedio en que el sistema se mantiene no disponible cuando ocurre un fallo total y antes de la arrancada gradual.

##### 5.2 Tiempo medio de recuperación

$X = \text{SUM}(T) / N$       Donde:

T = tiempo de recuperación de la inactividad en cada n oportunidad.

N = número de oportunidades en que el sistema entró en recuperación.

**Observaciones:** Permite determinar cuál es el tiempo promedio que toma el sistema para completar la recuperación desde el inicio de la recuperación parcial.

### 5.3 Recargabilidad

$X = A / B$       Donde:

A = número de veces que se provocó el reinicio o recarga en el plazo de tiempo previsto en la prueba especificada o en la implantación.

B = número total de veces que se provocó el reinicio o recarga durante la prueba especificada o la implantación.

**Observaciones:** Permite hacer un análisis de cuán frecuentemente el sistema logra recargar luego de una ejecución perdida, proveyendo nuevamente los servicios a los usuarios en el plazo de tiempo previsto.

### 5.4 Restaurabilidad

$X = A / B$       Donde:

A = número de casos de restauración exitosos.

B = número de casos de restauración probados por los requisitos.

**Observaciones:** Permite hacer un análisis de cuán capaz es el producto de auto restaurarse luego de un evento anormal o una solicitud.

### 5.5 Efectividad de la restauración

$X = A / B$       Donde:

A = número de casos de restauración exitosos en el plazo de tiempo previsto.

B = número de casos de restauración ejecutados.

**Observaciones:** Permite hacer un análisis de cuán efectiva es la capacidad de restauración.

✓ **Métricas para calcular la Modularidad del software**

**6. Índice de madurez del software según los módulos. (4)**

$IMS = [MT - (Fc + Fa + Fe)]/MT$  Donde:

IMS = índice de madurez del software.

MT = número de módulos en la versión.

Fc = número de módulos en la versión actual que se han cambiado.

Fa = número de módulos en la versión actual que se han añadido.

Fe = número de módulos en la versión actual que se han eliminado.

**Observaciones:** El IMS puede emplearse también como métrica para la planificación de las actividades de mantenimiento del software.

✓ **Métricas para calcular la Simplicidad del software**

**Complejidad Ciclomática de Thomas McCabe.**

**7.1.**  $V_G = 1 - (A - N + 2)$  Donde:

$V_G$  = complejidad ciclomática.

A = número de aristas del grafo.

N = número de nodos del grafo.

**7.1.**  $V_G = 1 - (N_R + 1)$  Donde:

$N_R$  = número de regiones del grafo.

✓ **Métricas para calcular la Completitud del software**

**8.1 Completitud de la Implementación funcional**

$C = 1 - A / B$  Donde:

A = número de funciones faltantes.

B = número de funciones descritas en la especificación de requerimientos.

**Observaciones:** Esta métrica permite conocer que tan completa está la implementación funcional del software que se desarrolla. Se debe contar las funciones faltantes detectadas mediante una revisión y compararlas con la cantidad de funciones expresadas según los requerimientos funcionales existentes.

### 8.2 Cobertura de la implementación funcional<sup>2</sup>

$X = 1 - A/B$  Donde:

A = número de funciones incorrectamente implementadas o funciones perdidas detectadas.

B = número de funciones descritas en la especificación de requisitos.

**Observaciones:** Esta métrica permite hacer un análisis de cuán correcta ha sido la implementación funcional.

#### ✓ Métricas para calcular la Consistencia del software

### 9.1. Especificidad de los requisitos. (2)

$Q1 = Nui / Nr$  Donde:

Nui = número de requisitos para los que todos los revisores tuvieron interpretaciones idénticas.

Nr = número total de requisitos.

**Observaciones:** Métrica basada en la consistencia de la interpretación de los revisores para cada requisito.

---

<sup>2</sup> Una de las Métricas Internas de la Calidad del Producto de Software recopiladas por el estándar ISO 9126-3.

### 2.5.3.3. Subprácticas para los atributos que definen la confiabilidad

- Para la métrica **1.1** obtener los valores de A y B de donde estén registrados los resultados de las pruebas realizadas.
- Para la métrica **2.1** obtener el valor de A de donde se encuentran registrados los resultados de las pruebas realizadas y el valor de T de donde haya quedado registrado el tiempo de operación.
- Para la métrica **2.2** obtener los valores de C y D, de donde se registren las respuestas obtenidas una vez hechas las pruebas al software.
- Para la métrica **3.1** se debe tener una plantilla donde se registren las fallas que va teniendo el software con las pruebas, además de las anuladas, de aquí se obtendrá el valor de A y para el valor de B se deben tener registrados los casos de pruebas realizados al software.
- Para la métrica **3.2** se debe tener una plantilla donde van a estar registrados los errores encontrados por el usuario, obteniéndose de ésta plantilla el valor de A y para el valor de B se debe tener un registro de las operaciones o entradas erróneas del usuario durante la operación.
- Para la métrica **3.3** se debe tener un registro con las interrupciones que han tenido en el tiempo de operación, de aquí se obtendrá el valor de A y el valor de B se obtendrá del registro de las fallas como para la métrica 3.1.
- Para la métrica **4.1.1** se debe tener registrado el tiempo de funcionamiento y de ese registro se obtendrá el valor de A, y el valor de B se obtendrá del registro de las fallas producidas por el software.
- Para la métrica **4.1.2** se deben tener registrados en la plantilla propuesta para el registro de las fallas, los intervalos de tiempo entre una falla y otra durante el tiempo de funcionamiento en cuestión, y el valor de D se obtendrá del registro de las fallas.
- Para la métrica **4.2** se obtendrán los valores de A y B del registro de las fallas producidas, donde se plasmarán además, las fallas que ya se hayan resuelto.
- Para la métrica **4.3** se obtendrán los valores de C y D del registro de errores producidos por los usuarios, en el que se reflejarán los errores corregidos y el total de errores descubiertos.

- Para la métrica **4.4** se obtendrá el valor de A1 del registro de los fallos producidos por el software, del que se necesita el número total de fallos detectados para asignárselo a A1, y el valor de A2 se obtendrá del registro de los casos de pruebas ejecutados sobre el software.
- Para la métrica **4.5** se obtendrán los valores de A y B del registro de los casos de pruebas ejecutados, en donde aparecerán los casos de pruebas realmente ejecutados y los casos de pruebas a ejecutar requeridos para cumplir los requisitos.
- Para la métrica **5.1** se obtendrá el valor de T del registro de tiempo donde aparecerán representados los tiempos de inactividad y el valor de N del registro de fallos, en el que aparecerán los desastres observados.
- Para la métrica **5.2** se obtendrá el valor de T también del registro de tiempo, en el que aparecerá plasmado el tiempo de recuperación de la inactividad en cada n oportunidad y se obtendrá el valor de N también del registro de tiempo donde se especificará el número de oportunidades en que el sistema entró en recuperación.
- Para la métrica **5.3** se tendrá en el registro de tiempo el número de veces que se provocó el reinicio o recarga en el plazo de tiempo previsto en la prueba especificada y de aquí se obtendrán los valores de A y B.
- Para la métrica **5.4** se tendrá en el registro de tiempo el número de casos de restauración exitosos y el número de casos de restauración probados por los requisitos, de donde se obtendrán los valores de A y B respectivamente.
- Para la métrica **5.5** se obtendrán los valores de A y B del registro de tiempo donde se encontrarán el número de casos de restauración exitosos en el plazo de tiempo previsto y el número de casos de restauración ejecutados.
- Para la métrica **6.1** se tendrá una plantilla para el registro de los módulos donde se plasmará el número de módulos por versiones, así como los módulos cambiados por versiones también, los módulos que se añaden en cada versión, así como los eliminados. De esta plantilla se obtendrán los valores de MT, Fc, Fa y Fe respectivamente.

- Para la métrica **7.1** en general se debe realizar el grafo derivado del fragmento de código que se esté analizando. En los anexos se les presenta un ejemplo de líneas de código y el grafo derivado de ellas, a partir del cual se calculará la complejidad ciclomática según McCabe.
- Para la métrica **8.1** se tendrá una plantilla con el registro de las funciones implementadas y descritas en la especificación de requisitos, así como las funciones faltantes, de la cual se obtendrán los valores de A y B.
- Para la métrica **8.2** se obtendrán los valores de A y B de la plantilla con el registro de las funciones implementadas y descritas en la especificación de requisitos, en la que aparecerá también el número de funciones incorrectamente implementadas o funciones perdidas detectadas.
- Para la métrica **9.1** se tendrá una plantilla donde se reflejen los requisitos totales luego de las revisiones.

#### 2.5.3.4. Fines Genéricos y Específicos de cada factor del RAS-IS

Seguidamente observarán una tabla en la que se refleja los fines específicos para cada uno de los factores que componen el RAS-IS, así como el fin genérico para el conjunto de estos factores.

RASIS	
<b>Fin genérico:</b> Lograr la accesibilidad del sistema en todo momento y evitar la corrupción y mal uso de los datos.	
Factores	Fines Específicos
<b>Fiabilidad (Reliability)</b>	<ul style="list-style-type: none"> <li>- Controlar la ocurrencia de fallas del sistema en el momento en que se esté operando en el mismo.</li> <li>- Lograr minimizar la ocurrencia de éstas fallas.</li> <li>- Llevar un registro de las fallas del sistema, surgidas durante la operación.</li> </ul>
<b>Disponibilidad (Availability)</b>	<ul style="list-style-type: none"> <li>- Verificar que se realice una operación continua del sistema tan larga como sea posible sin interrupciones, aun si ocurre un fallo.</li> <li>- Mantener la operación continua del</li> </ul>

	<p>sistema.</p> <ul style="list-style-type: none"> <li>- Controlar el tiempo de operación continua del sistema, y la cantidad de interrupciones que pueden ocurrir.</li> </ul>
<b>Servicialidad</b>	<ul style="list-style-type: none"> <li>- Controlar que el usuario pueda usar el sistema completamente, y así satisfacer sus requerimientos.</li> <li>- Lograr que el procesamiento se realice en el tiempo de respuesta requerido por el usuario.</li> <li>- Detectar de forma temprana las fallas, y repararlas.</li> </ul>
<b>Integridad (Integrity)</b>	<ul style="list-style-type: none"> <li>- Controlar que el usuario pueda usar el sistema completamente, y así satisfacer sus requerimientos.</li> <li>- Lograr que el procesamiento se realice en el tiempo de respuesta requerido por el usuario.</li> <li>- Detectar de forma temprana las fallas, y repararlas.</li> </ul>
<b>Seguridad (Security)</b>	<ul style="list-style-type: none"> <li>- Controlar la seguridad del sistema.</li> <li>- Verificar una correcta autenticación de los usuarios.</li> <li>- Lograr que la información este protegida.</li> </ul>

**Tabla 3. Descripción de los factores asociados**

### 2.5.3.5. Prácticas para cada uno de los factores del RAS-IS

- ✓ Métrica para calcular la Fiabilidad del software

### 10. MTBF: Tiempo medio entre fallas.

#### Figura 12. Representación de la métrica para calcular la Fiabilidad

Donde:

T1 = tiempo en funcionamiento.

$t_1$  = tiempo en falla.

**Observaciones:** MTBF (Tiempo medio entre fallas), dice qué paradas son las más frecuentes para un proceso. A mayor valor de MTBF, la fiabilidad será más alta.

✓ **Métrica para calcular la Servicialidad del software**

### 11. MTTR: Valor medio de Reparación

#### Figura 13. Representación de la métrica para calcular la Servicialidad

Donde:

$T_1$  = tiempo en funcionamiento.

$t_1$  = tiempo en falla.

**Observaciones:** Esta métrica permite una detección temprana y reparación rápida de fallas para su fácil mantenimiento. MTTR dice cuáles fallas son las más graves. La condición donde no sólo opera el sistema, sino también donde los requerimientos del usuario son satisfecho. El valor más pequeño es, el período más bajo de reparación. Por consiguiente, mientras más pequeño sea éste, mejor. A menor MTTR, mayor servicialidad.

✓ **Métrica para calcular la Disponibilidad del software**

### 12. Disponibilidad

#### Figura 14. Representación de la métrica para calcular la Disponibilidad

Donde:

MTBF = Tiempo medio entre fallas.

MTTR = Valor medio de Reparación.

**Observaciones:** Permite obtener el grado de disponibilidad del software, el resultado estará entre 0 y 1. Mientras mas se acerque a uno el software tendrá una mayor disponibilidad.

✓ **Prácticas para medir la integridad del software**

### 13. Aplicar las siguientes pruebas

**Pruebas de estrés:** Estas pruebas se realizan enviando muchas solicitudes al sistema con el objetivo de forzar la caída del servicio mientras dos o más clientes están intercambiando puntos.

**Pruebas de volumen:** Estas pruebas se realizan enviando gran volumen de datos para comprobar que el sistema logra gestionarlos todos sin inconvenientes, incluso, durante la prueba también se envía un volumen de datos mayor que lo especificado para el sistema con el fin de chequear que éste no colapsa ante un número de datos mayor que la cifra para el que fue diseñado.

**Observaciones:** Para evaluar la integridad de un sistema, se realizan pruebas de estrés o rendimiento y de volumen, con el propósito de mantener los datos consistentes.

Por ejemplo, la contradicción de datos puede ocurrir cuando son ejecutadas múltiples transacciones en un mismo tiempo. En este caso, el control exclusivo de la base de datos es usado para mantenimiento de la consistencia de los datos.

✓ **Prácticas para medir la Seguridad del software**

### 14. Aplicar las siguientes pruebas

#### Prueba de Seguridad y Control de Acceso

Se enfoca en dos áreas de seguridad:

- **Seguridad en el ámbito de aplicación**, incluyendo el acceso a los datos y a las funciones de negocios.
- **Seguridad en el ámbito de sistema**, incluyendo conexión, o acceso remoto al sistema.

**La seguridad en el ámbito de aplicación** asegura que, basado en la seguridad deseada los actores están restringidos a funciones o casos de uso específicos o limitados en los datos que están disponibles para ellos.

**La seguridad en el ámbito de sistema** asegura que, solo los usuarios con derecho a acceder al sistema son capaces de acceder a las aplicaciones y solo a través de los puntos de ingresos apropiados.

### Objetivo de la prueba

**Seguridad en el ámbito de aplicación:** Verificar que un actor pueda acceder solo a las funciones o datos para los cuales su tipo de usuario tiene permiso.

**Seguridad en el ámbito de sistema:** Verificar que solo los actores con acceso al sistema y a las aplicaciones, puedan acceder a ellos.

#### 2.5.3.6. Subprácticas para los factores asociados a la confiabilidad

- Para la métrica **10** se debe llevar un control de cada momento en el que falle el software, el tiempo que está funcionando sin interrupciones y el tiempo en el que se encuentra bajo una falla. Todo esto se debe archivar en el registro de tiempo del software.
- Para la métrica **11** se obtendrán los valores del registro de tiempo, donde aparecerán los tiempos de duración de cada falla, además de los tiempos de reparación. Como en la métrica MTBF.
- Para la métrica **12** se obtendrán los valores del resultado de las métricas de los factores fiabilidad y servicialidad.
- Para las prácticas de **integridad** se sugiere estudiar las pruebas de volumen y estrés y aplicarlas luego con el objetivo de medir la integridad del software en cuestión.
- Para las prácticas de **seguridad** se sugiere estudiar y aplicar las pruebas de seguridad y control de acceso propuestas anteriormente.

#### 2.5.4. Obtención de los resultados

##### 2.5.4.1. Asignación de pesos a los atributos de confiabilidad

Una vez hecha la recopilación de las métricas que se proponen, es necesario la asignación de los pesos que posee cada atributo de la confiabilidad en el producto software que se va a medir.

Para la asignación de los pesos a los atributos fue necesario aplicar el método empírico la entrevista, la misma se muestra en el Anexo #1. Dicha entrevista se aplica con el objetivo de tener una mayor seguridad y precisión en la asignación de los pesos, desde el punto de vista de diferentes roles del centro DATEC.

Además las investigadoras aseguran que los pesos de los atributos deben quedar claros antes de realizar el cálculo de las métricas, en dependencia del grado de importancia que le confirieren a cada atributo sobre el software. Se colocarán los atributos por orden de importancia y los pesos irán disminuyendo a partir de 1 según este mismo orden, de tal forma que cada persona que utilice el modelo pueda variar el orden de los atributos según la importancia que le confieran, pero se mantengan los pesos para ese orden.

De las entrevistas se obtuvieron los siguientes pesos para cada atributo.

Atributos	Peso
Precisión	1
Exactitud	0.9
Tolerancia ante fallos	0.8
Madurez	0.7
Recuperabilidad	0.6
Modularidad	0.5
Simplicidad	0.4
Compleitud	0.3
Consistencia	0.2

**Tabla 4. Asignación de pesos a los atributos**

#### 2.5.4.2. Puntuación final para cada atributo y factor

La puntuación final para cada atributo se dará una vez realizado el cálculo de la métrica y la ubicación en la tabla de escala. En el caso de que a un atributo se le evalúe más de una métrica entonces el resultado del cálculo de las métricas evaluadas se ubica en las Tablas de Escalas, según su valor requerido, se halla el promedio entre los resultados obtenidos después de haber llevado a la escala y este valor del promedio será la puntuación que se le dará a ese atributo.

Se puede utilizar la siguiente fórmula:

$$Ra = (R1 + R2 + Rn) / n$$

Donde **Ra** es el resultado final del atributo que se está midiendo, **R** es el resultado del cálculo de las métricas y **n** la cantidad de métricas con las que se mida ese atributo.

### 2.5.4.3. Escala de resultados

De ésta forma se obtendría el resultado final de la medición de cada atributo. Este resultado es necesario ubicarlo en la tabla de escalas que se muestra a continuación con el fin de conocer en que nivel de confiabilidad se encuentra el atributo que se está midiendo.

<u>INTERVALO</u>	<u>NIVEL</u>	<u>EVALUACIÓN</u>
<b>0 – 0.33</b>	<b>1</b>	<b>Inicial</b>
<b>0.34 – 0.67</b>	<b>2</b>	<b>Definido</b>
<b>0.68 – 1</b>	<b>3</b>	<b>Optimizado</b>

**Tabla 5. Escala de resultados**

Es necesario, por otra parte, realizar el cálculo de las métricas propuestas para los factores de calidad asociados a la confiabilidad y obtener un único valor.

### 2.1.1.1. Fórmulas para la confiabilidad de los atributos y factores

Una vez medidos cada atributo por separado, además de los factores asociados propuestos es importante llegar a un resultado final de la medición de la confiabilidad. Para lo cual se usarán las siguientes métricas:

$$C = ((Ra1 * Pa1) + (Ra2 * Pa2) + \dots + (Ran * Pan)) / n$$

Donde **C** será el resultado de la medición de la confiabilidad luego de medir los atributos que la definen, **Ra** es el resultado de cada atributo por separado, **Pa** es el peso asociado a cada atributo y **n** es la cantidad de atributos que se están midiendo.

$$CF = (Rf1 + Rf2 + Rf3) / 3$$

Donde **CF** va a ser el resultado de la confiabilidad según los factores de calidad asociados y **Rf** el resultado de cada factor por separado.

Se divide entre 3 pues es la cantidad de factores de los cuales se tienen métricas para su medición.

Estos valores de confiabilidad también deben ubicarse en la tabla de escalas antes expuesta.

### 2.1.1.2. Valor de confiabilidad afectado

Una vez que se hayan calculados estos valores se puede plantear que la confiabilidad en general se vería afectada por los definidos factores de calidad asociados de la siguiente manera:

$$CA = C * CF$$

Donde CA va a ser la confiabilidad afectada por los factores.

Luego se ubica este resultado final de confiabilidad en la tabla de escalas y se observará el nivel de confiabilidad con el que cuenta el producto una vez aplicado el modelo de medición.

Por otra parte también se pueden observar los resultados graficando en cualquier herramienta estadística cada uno de los valores obtenidos de la medición de los atributos y los factores asociados como se muestra en las figuras X y Y.

Por otra parte se les presenta en el Anexo # 2 una plantilla, en la cual aparecen las métricas propuestas por cada atributo y factor asociado a la confiabilidad, con su nivel requerido. Dicha plantilla se propone con el fin de que puedan ubicar los resultados luego del cálculo de cada métrica.

A continuación se presenta una plantilla donde quedan plasmados cada uno de los atributos con las métricas propuestas para su medición, además del valor requerido de los resultados y un campo en el que se ubicarán los resultados reales luego de aplicar la métrica.

### 2.1. Conclusiones del capítulo

En este capítulo, luego de realizar un estudio de la situación existente en DATEC respecto a la calidad de las soluciones que proponen, así como de las tecnologías que usan, se lleva a cabo la presentación del resultado científico que se propone. Dicho resultado es un modelo de medición de confiabilidad del software. También se detalla en el capítulo la estructura del modelo, describiendo cada una de sus partes y explicando además el proceso de obtención de los resultados de las métricas propuestas.

### 3. CAPÍTULO 3. VALIDACIÓN Y APLICACIÓN DE LA PROPUESTA.

#### 3.1. Introducción

En este capítulo se describe la validación de la propuesta por el método de multicriterio de expertos, así como una guía para llevar a cabo esta validación. También se muestran los resultados obtenidos al aplicar el modelo a una muestra de los productos del centro. Además se hace un análisis en dichos resultados para ver el nivel de confiabilidad de dichos productos, y de esta manera se da un resultado final de la validación.

La validación de la propuesta del Modelo se realiza a partir de la valoración de dos puntos fundamentales:

- ✓ Aplicación de la evaluación por multicriterio de expertos a través de un método capaz de evaluar no solamente los resultados propuestos sino también de caracterizar la calidad de la investigación.
- ✓ Valoración de los resultados obtenidos de la aplicación del Modelo de medición de la confiabilidad en el proyecto desarrollado por DATEC "Generador de Reportes".

#### 3.2. Validación de la propuesta por el método de Expertos.

La obtención de pronósticos se apoya en dos tipos generales de métodos: los de base objetiva y los de base subjetiva. Los métodos objetivos utilizan técnicas matemáticas bien fundamentadas para procesar la información disponible; pero estas técnicas resultan impotentes para captar la evolución futura de situaciones con alto grado de incertidumbre como la que se investiga. De aquí que se hace necesario la utilización de métodos que estén estructurados a partir de la aceptación de la intuición como una comprensión sinóptica de la realidad social, y basados en la experiencia y conocimiento de personas considerados expertos en la materia en cuestión. Estos métodos denominados subjetivos son conocidos como métodos de consulta o evaluación de expertos.

Se le denomina experto, tanto al individuo en sí como a un grupo de personas u organizaciones capaces de ofrecer valoraciones conclusivas de un problema en cuestión y hacer recomendaciones respecto a sus momentos fundamentales con un máximo de competencia.

El método de evaluación de expertos se emplea para comprobar la calidad y efectividad de los resultados de las investigaciones, tanto en su concepción teórica como de su aplicación en la práctica social, es decir, el impacto que se espera obtener con la aplicación de los resultados teóricos de la

investigación en la práctica, cuando resulta imposible o muy difícil realizar las mediciones por métodos más precisos. (29)

Generalmente la evaluación multicriterio de expertos se encuentra asociada a los conceptos: (30)

- ✓ Decisión: Elección de una de las alternativas posibles para solucionar un problema.
- ✓ Alternativas: Cada una de las soluciones posibles a un problema, dotadas de ventajas e inconvenientes diferentes.
- ✓ Criterios: Distintos aspectos de la realidad que inciden de alguna manera en las ventajas o inconvenientes de las alternativas disponibles como soluciones al problema.

### 3.2.1. Guía para la validación de la propuesta

Para la validación de la propuesta se expone a continuación la guía a seguir.

1. Hacer la selección de los expertos que participarán en la validación.

Para la selección de los expertos se tuvo en cuenta su experiencia en la informática y la efectividad de su labor como profesional. Además deben contar con conocimientos profundos de la Gestión de proyectos y Calidad de Software. También deben tener conocimientos de métodos de medición de software, además de haber obtenido resultados satisfactorios en la producción o la investigación. Lo que proporciona que las respuestas tengan la calidad requerida, puesto que las opiniones necesitan ser confiables y contar con un alto grado de validez.

Se hace necesario que los expertos cuenten con una serie de características, como son:

- |  |  |
|--|--|
| ✓ Competencia.                             | ✓ Conformidad.                             |
| ✓ Creatividad.                             | ✓ Capacidad de análisis.                   |
| ✓ Disposición a participar en la encuesta. | ✓ Espíritu colectivista y autocrítico.     |
|  | ✓ Efectividad de su actividad profesional. |

Grupo de expertos seleccionados:

ID	Expertos
E <sub>1</sub>	Asesor de Calidad de DATEC. <b>Michael González Jorin</b>

<b>E<sub>2</sub></b>	Esp. Grupo de Normalización y Métricas. <b>Ing. Lissette Rodríguez Verdecia.</b>
<b>E<sub>3</sub></b>	Esp. Grupo de Normalización y Métricas. <b>Ing. Dayami Rodríguez Brito.</b>
<b>E<sub>4</sub></b>	Esp. Grupo de Normalización y Métricas. <b>Ing. José Alejandro Lugo García.</b>
<b>E<sub>5</sub></b>	Dpto. Gestión de Calidad de DATEC. <b>Ing. Daimi Bretones Lorenzo</b>
<b>E<sub>6</sub></b>	Esp. Grupo de Normalización y Métricas: <b>Ing. Kariné Ramos Blanco</b>
<b>E<sub>7</sub></b>	J' de Línea de DATEC. <b>Ing. Asnioby Hernández López</b>

**Tabla 6. Expertos e Identificadores del Proceso de Evaluación de Expertos**

2. Enviar la documentación necesaria a los expertos que participarán en la evaluación y validación del modelo propuesto.

Para este punto se les envió a los expertos un resumen del Trabajo de Diploma para que estudien el tema y logren ofrecer una razón precisa del mismo.

Es de notar que en este método se parte, precisamente, de buscar las diferentes alternativas que se pretenden valorar. Luego de haberlas determinado, se procede con el proceso de consenso y selección de la mejor alternativa. Una vez determinadas las posibles acciones se indaga entre los expertos si consideran que esas son las que conformarán la lista final, o en caso contrario, si existen desacuerdos al respecto.

Las alternativas que se proponen para valorar la propuesta están basadas en cuatro criterios fundamentales:

**Criterio del Método Científico:** referente al nivel de calidad de la fundamentación teórica

**Criterios de Implantación:** encierra los aspectos relacionados con usabilidad del modelo.

**Criterios de Generalización:** recoge las acotaciones concernientes al nivel de comprensión de la investigación y la propuesta.

**Criterios de Impacto:** posibilidades de trascendencia de la propuesta.

Estos criterios se especifican en las siguientes alternativas:

Indicadores	Alternativas
A <sub>1</sub>	Nivel de calidad de la investigación.
A <sub>2</sub>	Aportes científicos novedosos.
A <sub>3</sub>	Novedad científica de la investigación.
A <sub>4</sub>	Necesidad de uso del modelo.
A <sub>5</sub>	Satisfacción de las necesidades de la producción.
A <sub>6</sub>	Garantía de principios básicos de la medición de la confiabilidad del software.
A <sub>7</sub>	Nivel de comprensión del modelo.
A <sub>8</sub>	Facilidades de uso.
A <sub>9</sub>	Impacto en el área para la cual está destinado el modelo.
A <sub>10</sub>	Contribución al proceso de desarrollo de software.
A <sub>11</sub>	Contribución a la gestión de proyectos.
A <sub>12</sub>	Posibilidades de aplicación

**Tabla 7. Alternativas e Indicadores del Proceso de Evaluación de Expertos**

### Asignación de pesos a las alternativas

Los expertos deberán asignar un peso o ponderación a cada alternativa, con el fin de ordenarlas según su influencia o importancia en el tema en cuestión. Los pesos estarán en un intervalo de 1 a 5, lo que significará la opinión que brinda el experto sobre cada alternativa.

5- Excelente.

4- Aceptable, sin embargo se pudo haber perfeccionado.

3- Buena.

2- Insuficiente.

1- Se recomienda que no sea considerada una alternativa a valorar, pues no se cumple en la investigación con los objetivos de la alternativa.

Los resultados concernientes a la recogida de los pesos de cada alternativa se muestran en la Tabla 11.

**E:** el experto que realiza la validación.

**A:** alternativa evaluada.

**E<sub>p</sub>:** promedio del peso asignado a cada alternativa por los expertos.

**P<sub>r</sub>:** peso relativo de cada alternativa.

Resultado de la asignación de pesos para cada alternativa por los expertos.

Alternativas	E <sub>1</sub>	E <sub>2</sub>	E <sub>3</sub>	E <sub>4</sub>	E <sub>5</sub>	E <sub>6</sub>	E <sub>7</sub>	E <sub>p</sub>	E <sub>r</sub>
<b>A<sub>1</sub></b>	4	4	5	5	4	4	3	4.14	0.08
<b>A<sub>2</sub></b>	5	4	4	5	3	4	4	4.14	0.08
<b>A<sub>3</sub></b>	4	3	5	4	4	3	3	3.71	0.07
<b>A<sub>4</sub></b>	5	5	5	5	4	4	5	4.71	0.1
<b>A<sub>5</sub></b>	4	4	3	3	4	5	4	3.85	0.08
<b>A<sub>6</sub></b>	5	4	4	4	3	3	3	3.71	0.07
<b>A<sub>7</sub></b>	4	5	5	5	4	4	3	4.29	0.09
<b>A<sub>8</sub></b>	5	5	5	5	4	4	4	4.57	0.09
<b>A<sub>9</sub></b>	4	5	4	3	5	4	3	4	0.08
<b>A<sub>10</sub></b>	5	4	5	3	4	4	4	4.14	0.08
<b>A<sub>11</sub></b>	4	5	5	5	4	4	3	4.29	0.09
<b>A<sub>12</sub></b>	3	3	4	4	5	5	4	4	0.08
<b>Total</b>	-	-	-	-	-	-		49.55	0.99

**Tabla 8. Asignación de pesos a las alternativas por los expertos**

Los resultados referidos a la recogida de la evaluación de cada alternativa se muestran en la Tabla 12:

**E<sub>a</sub>:** Evaluación promedio de cada experto.

Alternativas	E <sub>1</sub>	E <sub>2</sub>	E <sub>3</sub>	E <sub>4</sub>	E <sub>5</sub>	E <sub>6</sub>	E <sub>7</sub>	E <sub>a</sub>
A <sub>1</sub>	5	5	5	5	5	5	4	4.86
A <sub>2</sub>	5	5	4	4	5	4	4	4.43
A <sub>3</sub>	4	5	5	4	5	3	4	4.29
A <sub>4</sub>	5	3	3	4	3	4	3	3.57
A <sub>5</sub>	4	5	3	3	4	5	3	3.86
A <sub>6</sub>	5	5	5	3	4	3	5	4.29
A <sub>7</sub>	5	5	4	4	3	5	4	4.29
A <sub>8</sub>	4	4	5	5	4	4	3	4.14
A <sub>9</sub>	5	5	5	4	5	4	4	4.57
A <sub>10</sub>	5	4	5	3	4	4	5	4.29
A <sub>11</sub>	5	5	3	3	3	3	3	3.57
A <sub>12</sub>	4	5	5	5	3	5	3	4.29

**Tabla 9. Resultado de la encuesta para la recogida de la evaluación de cada experto**

3. Conociendo el peso relativo ( $P_r$ ) y la calificación promedio ( $E_a$ ) de cada alternativa se puede obtener el valor de  $P_r \times E_a$ , en la Tabla # se muestran los resultados.

Calificación promedio de cada alternativa.

Alternativas	E <sub>a</sub>	P <sub>r</sub>	P <sub>r</sub> × E <sub>a</sub>
A <sub>1</sub>	4.86	0.08	0.39
A <sub>2</sub>	4.43	0.08	0.35
A <sub>3</sub>	4.29	0.07	0.30
A <sub>4</sub>	3.57	0.1	0.36
A <sub>5</sub>	3.86	0.08	0.31

<b>A<sub>6</sub></b>	4.29	0.07	0.30
<b>A<sub>7</sub></b>	4.29	0.09	0.39
<b>A<sub>8</sub></b>	4.14	0.09	0.37
<b>A<sub>9</sub></b>	4.57	0.08	0.37
<b>A<sub>10</sub></b>	4.29	0.08	0.34
<b>A<sub>11</sub></b>	3.57	0.09	0.32
<b>A<sub>12</sub></b>	4.29	0.08	0.34
<b>Σ (P<sub>r</sub> x E<sub>a</sub>)</b>	-	-	<b>4.14</b>

**Tabla 10. Calificación promedio de cada alternativa**

4. Se calcula el Índice de Aceptación (IA).

$$IA = \Sigma (Pr \times Ea) / 5$$

$$IA = 4.14 / 5$$

$$IA = \mathbf{0.83}$$

5. Se ubica el Índice de Aceptación en rangos ya predefinidos, determinando así la probabilidad de éxito.

### Rangos predefinidos del Índice de Aceptación

$IA > 0,7$  Existe alta probabilidad de éxito

$0,7 > IA > 0,5$  Existe probabilidad media de éxito

$0,5 > IA > 0,3$  Probabilidad de éxito baja

$0,3 > IA$  Fracaso seguro

### 3.2.2. Resultados de la evaluación del Modelo

Según la valoración de los expertos:

La investigación cuenta con excelentes niveles de calidad, y de aportes novedosos. La propuesta tiene un buen nivel de comprensión y facilidad de uso. Brinda la posibilidad de ser aplicada en los proyectos de desarrollo de software de la UCI y cualquier otra empresa desarrolladora de software.

De acuerdo al Índice de Aceptación calculado se demuestra que la propuesta tiene alta probabilidad de éxito.

### 3.3. Aplicación de la Propuesta en el proyecto Generador de Reportes.

#### ¿Qué es el "Generador de Reportes"?

Plataforma como un sistema independiente que conecta a Bases de Datos para la confección de los Reportes (Oracle, MySQL, PostgreSQL).

#### División por módulos de la plataforma

#### Figura 15. División por módulos del Generador de Reportes

#### 3.3.1. Efectividad del Modelo de medición de la confiabilidad en el proyecto "Generador de Reportes".

Durante el proceso de evaluación en el proyecto "Generador de Reportes", se aplicó el 50% de las métricas propuestas en el modelo. Fueron seleccionadas a consideración de las investigadoras de acuerdo con la documentación registrada del proyecto.

A continuación se plantea todo el proceso de aplicación de las métricas, así como sus resultados:

#### ✓ Resultado de las métricas para calcular la Precisión

##### 1.1 Proporción de las respuestas obtenidas

Esta métrica se aplicó en el módulo Visor de Reportes, pues era el que tenía la documentación requerida. Estaban registrados los datos de la versión 1.0 y 2.0 del proyecto.

Para esta métrica  $X=1$ , por lo que el valor de precisión es de 1.

#### ✓ Resultado de las métricas para calcular la Exactitud

##### 2.1 Exactitud esperada

Esta métrica fue aplicada en los módulos: Acciones Generales, Diseñador de Modelos, Diseñador de Reportes y Visor de Reportes. Estos son los que tenían un registro de los Casos de Prueba probados.

Para esta métrica  $X=0.7$

### **2.2 Variación de las respuestas del software**

Esta métrica fue aplicada en los módulos: Acciones Generales, Diseñador de Modelo, Diseñador de Reportes y Visor de Reportes. Estos son los que tenían un registro de los Casos de Prueba probados.

Para esta métrica  $Y=0.92$

Por lo que el valor de exactitud luego de aplicar las formulas anteriormente expuestas para este caso es de 0.81.

### **✓ Resultado de las métricas para calcular la Tolerancia ante fallos**

#### **3.1 Anulación de fallas.**

Esta métrica se aplicó en el módulo Visor de Reportes. Se tuvo en cuenta los registros de la versión 1.0 y versión 2.0 del modulo.

Para la misma el valor de  $X=0$

#### **3.2 Anulación de operaciones incorrectas (error operacional o proporción de error detectado en la entrada).**

Esta métrica no fue aplicada, ya que se carecía de los registros de las entradas o funciones hechas por usuario durante la operación del proyecto.

Por lo que el valor de la medición de la tolerancia según las métricas aplicadas es de 0.

### **✓ Resultados de las métricas para calcular la Madurez**

#### **4.1 Tiempo promedio entre fallas (TPF).**

Esta métrica no fue aplicada en el proyecto pues no se tenía la documentación de los LOG de tiempo, que es una información necesaria para el cálculo.

#### **4.2. Resolución de fallas.**

Esta métrica fue aplicada en el Módulo Visor de Reportes, que era el que poseía la información de la versión 1.0 y 2.0.

Luego de aplicar esta métrica el valor de  $X$  es 0.

### **4.3 Defectos eliminados.**

Esta métrica fue aplicada en el Módulo Visor de Reportes, que era el que poseía la información de la versión 1.0 y 2.0.

Para esta métrica el valor de  $Y$  es 0.

### **4.4 Intensidad de fallos totales contra casos de prueba.**

Esta métrica fue aplicada en los módulos: Acciones Generales, Diseñador de Modelo, Diseñador de Reportes y Visor de Reportes. Estos son los que tenían un registro de los Casos de Prueba probados.

El resultado de  $X$  para esta métrica fue de 0.2.

### **4.5 Cobertura de las pruebas**

Esta métrica no fue aplicada ya que no se cuenta con los registros de los casos de prueba necesarios para el cumplimiento de todos los requisitos del proyecto.

### **4.6 Madurez de las pruebas**

Esta métrica no fue aplicada ya que no se cuenta con los registros de los casos de prueba necesarios para el cumplimiento de todos los requisitos del proyecto.

Por lo que el resultado del atributo madurez luego de aplicar las métricas propuestas es de 0.2.

#### **✓ Resultados de las métricas para calcular la Recuperabilidad**

Las métricas de recuperabilidad no fueron aplicadas ya que no se cuenta con los registros de tiempo necesarios para el cálculo.

#### **✓ Resultados de las métricas para calcular la Modularidad**

### **6.1. Índice de madurez del software según los módulos.**

Para esta métrica se hizo un análisis de los módulos registrados en las 2 iteraciones, de las cuales se tiene información.

El índice de madurez del software según los cálculos es de 1.

Por lo que el atributo modularidad tendrá un resultado final de 1.

✓ **Resultados de las métricas para calcular la Simplicidad**

### 7.1. Complejidad Ciclomática

Esta métrica no pudo ser aplicada ya que no se tienen los registros del proceso de implementación del proyecto.

✓ **Resultados de las métricas para calcular la Completitud**

Estas métricas no fueron aplicadas, ya que no se posee la información de los requisitos del proyecto.

✓ **Resultados de las métricas para calcular la Consistencia**

### 9.1. Especificidad de los requisitos.

Esta métrica no se aplicó pues se carece de la documentación en la cual se plasman los requisitos especificados en el proyecto.

✓ **Resultados de las métricas para calcular los factores del RAS-IS**

Las métricas propuestas para medir los elementos pertenecientes al RAS-IS que son fiabilidad, disponibilidad y servicialidad no fueron aplicadas, pues no se obtuvo el registro de los LOG de tiempo. En el caso de los elementos de integridad y seguridad se propone que sean evaluados de forma cualitativa a través de las pruebas propuestas. Se evidenció en la documentación obtenida, que los casos de prueba de estrés fueron aplicados y no fueron detectadas no conformidades. Por tanto se puede decir que las pruebas de estrés aplicadas tuvieron una respuesta satisfactoria.

### 3.3.2. Resultado final de confiabilidad

Para obtener un resultado final de la medición de la confiabilidad se aplicó la siguiente fórmula:

$$C = ((Ra_1 * Pa_1) + (Ra_2 * Pa_2) + \dots + (Ra_n * Pa_n)) / n$$

Obteniéndose según los cálculos un valor de **C = 0.47**

Dicho valor fue representado en la tabla de escalas propuesta en el capítulo 2. Por tanto se obtuvo que el proyecto “Generador de Reportes” se enmarca en el nivel 2 de la escala, lo que lleva a la conclusión de que se encuentra en un nivel de confiabilidad definido.

Dichos resultados se verán representados en la gráfica que aparece a continuación.

### **Figura 16. Resultados de confiabilidad de los atributos medidos**

#### **3.4. Conclusiones del capítulo**

El capítulo se centró en la Valoración de la propuesta, tanto en la concepción teórica como en la práctica.

La valoración se encausó hacia los resultados obtenidos durante:

- La evaluación por el multicriterio de expertos.
- La aplicación en el proyecto “Generador de Reportes”.

La valoración de la propuesta mostró excelentes resultados, según el método teórico aplicado, que de forma general avalan el nivel de calidad, funcionalidad y aplicabilidad, de la propuesta.

Por otra parte según la valoración de la propuesta por el método práctico mediante la aplicación de la misma en el proyecto “Generador de Reportes”, se obtuvo un resultado parcial de confiabilidad debido a que no se pudo aplicar la totalidad de las métricas por algunos inconvenientes en cuanto a la documentación facilitada por el centro.

### CONCLUSIONES

El Modelo de Medición de la Confiabilidad y el análisis de las tecnologías de los productos de DATEC surge luego de analizar las tendencias actuales de los modelos de calidad existentes, las particularidades y las necesidades de medición de la confiabilidad de los productos, además de analizar las tecnologías usadas en el centro para el desarrollo de sus productos y comprender cuán importante resulta su concepción y aplicación para la monitorización, valoración y control de los atributos de confiabilidad definidos en el modelo propuesto, así como de los factores asociados a la misma. Se puede concluir:

- ✓ La propuesta está enmarcada en el modelo de Medición de la Confiabilidad, pues el mismo recoge prácticas adecuadas a las peculiaridades del proceso productivo en DATEC.
- ✓ Las métricas definidas cubren los atributos y factores asociados a la confiabilidad: Precisión, Exactitud, Tolerancia ante fallos, Madurez, Recuperabilidad, Modularidad, Simplicidad, Completitud, Consistencia, Fiabilidad, Disponibilidad, Servicialidad, Integridad y Seguridad del software.
- ✓ Los expertos concuerdan en su efectividad, tanto en la concepción teórica como en los resultados que se obtendrán con su aplicación.
- ✓ Los resultados obtenidos durante la aplicación del modelo en el proyecto "Generador de Reportes", reportaron buenos resultados de forma parcial, que avalan su eficacia.

Basado en los aspectos anteriormente expuestos, se consideran abordados y cumplidos los objetivos propuestos, logrando de esta manera solucionar el problema planteado con anterioridad. Durante la realización del trabajo fueron consultadas diferentes bibliografías actualizadas para la obtención de información confiable y actual.

### RECOMENDACIONES

Este trabajo de diploma propone un Modelo de Medición de la confiabilidad y análisis de las tecnologías en los productos de DATEC, en función de implementar acciones que socialicen el conocimiento aportado, se recomienda:

- ✓ Publicar los resultados de este trabajo de diploma para poner a disposición del centro de DATEC el modelo de medición propuesto.
- ✓ Proponer, tras comprobar un resultado exitoso, la utilización del modelo en los Proyectos de Desarrollo de Software de la UCI.
- ✓ Elaborar la estrategia de implementación del Modelo.

Recomendaciones enfocadas a la mejora de la concepción de modelo:

- ✓ Como complemento de la propuesta se puede realizar la automatización de la misma, en aras de facilitar el trabajo del consultor de la propuesta.
- ✓ Crear un sistema de recolección de los datos dividida en dos vertientes:
  - Proveer: Colecciona los datos que se utilizarán posteriormente a la hora de efectuar el cálculo matemático de las métricas,

Recoger y Retroalimentar: Una vez efectuado el cálculo de las métricas el resultado obtenido sirva para la reutilización y retroalimentación de datos.

### REFERENCIAS BIBLIOGRÁFICAS

1. dejameser.wordpress.com. [En línea] 28 de 5 de 2007. <http://dejameser.wordpress.com/pm-bookmarks-9-reas-de-conocimiento-pmbok>.
2. **Pressman, Roger**. *Ingeniería de Software. Un Enfoque Práctico*. s.l. : Davis, 1998.
3. **66-001-92), ISO 8402 (UNE**. [En línea]
4. **982.1-1988, Estándar IEEE**.
5. mgar.net. *mgar.net*. [En línea] <http://mgar.net/soc/isointro.htm>.
6. *Gestión, Control y Garantía de la calidad del software*. s.l. : ITEC "Leonardo DaVinci", 2006.
7. rincondelvago.com. [En línea] 24 de 3 de 2010. [http://html.rincondelvago.com/control-de-calidad\\_15.html](http://html.rincondelvago.com/control-de-calidad_15.html).
8. **Figueroa\*, María Antonieta Abud**. Calidad en la Industria del Software. . [En línea] [Citado el: 20 de 3 de 2010.] <http://www.revistaupiicsa.20m.com/Emilia/RevEneAbr04/Antonieta1.pdf>.
9. *La implantación de ISO 9001 en el desarrollo de software*. **Osorio, Gloria Quintanilla**. 9 de 1999, Soluciones Avanzadas.
10. **Sanders, Joc y Eugene Curran**. *Software Quality*. Addison Wesley : s.n. A Framework for Success in Software Development and Support.
11. International Organization for Standardization. Norma de gestión de la calidad, parte 3. [En línea] <http://alarcos.inf-cr.uclm.es/doc/calidad/ISO%209000'3.doc>.
12. **Mohanty**. 1979.
13. **Litlewood**. *Strigini*. 2000.
14. **Wellings, Buns y**. 1990.
15. catarina.udlap.mx. *catarina.udlap.mx*. [En línea] 2007. [catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/.../capítulo2.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/.../capítulo2.pdf).
16. **Paredes, Cacia Oro Rivera y Leidys de la Luz**. *Propuesta de Métricas para Evaluar el Desarrollo del Proyecto IMAC*. 2009.
17. **Leticia Dávila Nicanor, Pedro Mejías Alvarez**. delta.cs.cinvestav.mx. [En línea] [Citado el: 15 de 4 de 2010.] <http://delta.cs.cinvestav.mx/~pmaalvarez/davila-mejia.pdf>.
18. **Michael**. [En línea] 1999.
19. **Pressman, Roger**. *Ingeniería de Software. Un enfoque práctico*. [ed.] McGraw Hill. 2002.

20. **Fillottrani, Pablo R.** Calidad en el Desarrollo de Software. *Calidad en el Desarrollo de Software*. [En línea] 2007. <http://www.cs.uns.edu.ar/~prf/teaching/SQ07/clase6.pdf>.
21. **Palazzolo, Lic. Cecilia.** noqualityinside.com. *noqualityinside.com*. [En línea] 2005. [http://noqualityinside.com/nqi/nqifiles/CalidadDeSW\\_diap.pdf](http://noqualityinside.com/nqi/nqifiles/CalidadDeSW_diap.pdf).
22. **Instituto de Investigaciones en Normalización. Ministerio de Ciencia, Tecnología y Medio Ambiente.** cubaindustria.cu. [En línea] 2007. <http://www.inin.cubaindustria.cu/products/rev-2007-23.htm>. 3.
23. **Melvin.** Modelos de Gestión de la Calidad del Softwar. . [En línea] 2008. <http://modelosdegestiondelacalidad.blogspot.com/>.
24. **Deronceré, Alain Fernández.** fing.edu. [En línea] 2008. <http://www.fing.edu.uy>.
25. **M., Prof. Luis Eduardo Mendoza.** Sistemas de Información III. Teoría. [En línea] 2000.
26. CALISOFT. Centro de Calidad para soluciones tecnológicas. [En línea] UCI, 2008-2009. <http://calisoft.uci.cu/>.
27. **Rodríguez, Dres. Rogelio Bermúdez y Marisela.** www.monografias.com. [En línea] <http://www.monografias.com/trabajos64/resultados-cientificos-investigacion-educacional/resultados-cientificos-investigacion-educacional2.shtml>.
28. **Ferreira, M., García, F., Ruiz F. y otros.** Medición del Software. [En línea] 11 de 2006. <http://tsi.uclm.es>.
29. **Veliz, I.Y.Z.** *Modelo de Gestión de Riesgos en Proyectos de Desarrollo de Software*. Ciudad de la Habana : UCI, 2007.
30. **Sendra, J.B., SIG y evaluación multicriterio. 2006, Universidad de Alcalá: España.].** *SIG y evaluación multicriterio*. España : Universidad de Alcalá, 2006.
31. softqm.blogspot.com. *softqm.blogspot.com*. [En línea] 11 de 2006. <http://softqm.blogspot.com/2006/11/gestin-de-la-calidad-del-software.html>.
32. **Wellings, Buns y.** 1990.
33. **Sagarra, Yisel Nadereau.** *Propuesta de automatización para el cálculo de las Métricas de Calidad generadas en los proyectos productivos de la Facultad 9*. Ciudad de la Habana : s.n., 2009.
34. eumed.net. [En línea] 2009. <http://www.eumed.net/ce/2009b/tjm.htm>.
35. **Bedoya, Camilo Bustamante.** *Automatización de los procedimientos para la recolección y consolidación de las métricas externas del producto según la norma ISO/IEC 9126:2001*.
36. catarina.udlap.mx. [En línea] [Citado el: 10 de 4 de 2010.] [catarina.udlap.mx/u\\_dl\\_a/tales/documentos/list/.../capitulo2.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/list/.../capitulo2.pdf).

## BIBLIOGRAFÍA

1. dejameser.wordpress.com. [En línea] 28 de 5 de 2007. <http://dejameser.wordpress.com/pm-bookmarks-9-reas-de-conocimiento-pmbok>.
2. **Pressman, Roger.** *Ingeniería de Software. Un Enfoque Práctico.* s.l. : Davis, 1998.
3. **66-001-92), ISO 8402 (UNE.** [En línea]
4. **982.1-1988, Estándar IEEE.**
5. mgar.net. *mgar.net.* [En línea] <http://mgar.net/soc/isointro.htm>.
6. *Gestión, Control y Garantía de la calidad del software.* s.l. : ITEC "Leonardo DaVinci", 2006.
7. rincondelvago.com. [En línea] 24 de 3 de 2010. [http://html.rincondelvago.com/control-de-calidad\\_15.html](http://html.rincondelvago.com/control-de-calidad_15.html).
8. **Figuroa\*, María Antonieta Abud.** Calidad en la Industria del Software. . [En línea] [Citado el: 20 de 3 de 2010.] <http://www.revistaupiicsa.20m.com/Emilia/RevEneAbr04/Antonieta1.pdf>.
9. *La implantación de ISO 9001 en el desarrollo de software.* **Osorio, Gloria Quintanilla.** 9 de 1999, Soluciones Avanzadas.
10. **Sanders, Joc y Eugene Curran.** *Software Quality.* Addison Wesley : s.n. A Framework for Success in Software Development and Support.
11. International Organization for Standardization. Norma de gestión de la calidad, parte 3. [En línea] <http://alarcos.inf-cr.uclm.es/doc/calidad/ISO%209000'3.doc>.
12. **Mohanty.** 1979.
13. **Litlewood.** *Strigini.* 2000.
14. **Wellings, Buns y.** 1990.
15. catarina.udlap.mx. *catarina.udlap.mx.* [En línea] 2007. [catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/.../capitulo2.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/.../capitulo2.pdf).
16. **Paredes, Cacia Oro Rivera y Leidys de la Luz.** *Propuesta de Métricas para Evaluar el Desarrollo del Proyecto IMAC.* 2009.
17. **Leticia Dávila Nicanor, Pedro Mejías Alvarez.** delta.cs.cinvestav.mx. [En línea] [Citado el: 15 de 4 de 2010.] <http://delta.cs.cinvestav.mx/~pmaalvarez/davila-mejia.pdf>.

18. **Michael.** [En línea] 1999.
19. **Pressman, Roger.** *Ingeniería de Software. Un enfoque práctico.* [ed.] McGraw Hill. 2002.
20. **Fillottrani, Pablo R.** Calidad en el Desarrollo de Software. *Calidad en el Desarrollo de Software.* [En línea] 2007. <http://www.cs.uns.edu.ar/~prf/teaching/SQ07/clase6.pdf>.
21. **Palazzolo, Lic. Cecilia.** noqualityinside.com. *noqualityinside.com.* [En línea] 2005. [http://noqualityinside.com/nqi/nqifiles/CalidadDeSW\\_diap.pdf](http://noqualityinside.com/nqi/nqifiles/CalidadDeSW_diap.pdf).
22. **Instituto de Investigaciones en Normalización. Ministerio de Ciencia, Tecnología y Medio Ambiente.** cubaindustria.cu. [En línea] 2007. <http://www.inin.cubaindustria.cu/products/rev-2007-23.htm>. 3.
23. **Melvin.** Modelos de Gestión de la Calidad del Software. . [En línea] 2008. <http://modelosdegestiondelacalidad.blogspot.com/>.
24. **Deronceré, Alain Fernández.** fing.edu. [En línea] 2008. <http://www.fing.edu.uy>.
25. **M., Prof. Luis Eduardo Mendoza.** Sistemas de Información III. Teoría. [En línea] 2000.
26. CALISOFT. Centro de Calidad para soluciones tecnológicas. [En línea] UCI, 2008-2009. <http://calisoft.uci.cu/>.
27. **Rodríguez, Dres. Rogelio Bermúdez y Marisela.** www.monografias.com. [En línea] <http://www.monografias.com/trabajos64/resultados-cientificos-investigacion-educacional/resultados-cientificos-investigacion-educacional2.shtml>.
28. **Ferreira, M., García, F., Ruiz F. y otros.** Medición del Software. [En línea] 11 de 2006. <http://tsi.uclm.es>.
29. **Veliz, I.Y.Z.** *Modelo de Gestión de Riesgos en Proyectos de Desarrollo de Software.* Ciudad de la Habana : UCI, 2007.
30. **Sendra, J.B., SIG y evaluación multicriterio. 2006, Universidad de Alcalá: España.].** *SIG y evaluación multicriterio.* España : Universidad de Alcalá, 2006.
31. softqm.blogspot.com. *softqm.blogspot.com.* [En línea] 11 de 2006. <http://softqm.blogspot.com/2006/11/gestin-de-la-calidad-del-software.html>.
32. **Wellings, Buns y.** 1990.
33. **Sagarra, Yisel Nadereau.** *Propuesta de automatización para el cálculo de las Métricas de Calidad generadas en los proyectos productivos de la Facultad 9.* Ciudad de la Habana : s.n., 2009.
34. eumed.net. [En línea] 2009. <http://www.eumed.net/ce/2009b/tjm.htm>.
35. **Bedoya, Camilo Bustamante.** *Automatización de los procedimientos para la recolección y consolidación de las métricas externas del producto según la norma ISO/IEC 9126:2001.*

36. catarina.udlap.mx. [En línea] [Citado el: 10 de 4 de 2010.]  
[catarina.udlap.mx/u\\_dl\\_a/tales/documentos/list/.../capitulo2.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/list/.../capitulo2.pdf).