



*Universidad de las Ciencias Informáticas  
Facultad # 2*



*“Propuesta de Métricas para Evaluar el Desarrollo del Proyecto  
IMAC”.*

*Trabajo para Optar por el Título de:  
Ingeniero en Ciencias Informáticas*

***Autores:**  
Cecilia Oro Rivera  
Leidys de la Luz Paredes*

***Tutor:**  
Aymara Marín Díaz*

*Ciudad de la Habana 2009 “Año del 50 Aniversario del Triunfo de la Revolución”.*

## *Frase:*

*“ Cuando pueda medir lo que está diciendo y expresarlo con números, ya conoce algo sobre ello; cuando no pueda medir, cuando no pueda expresar lo que dice con números, su conocimiento es precario y deficiente: puede ser el comienzo del conocimiento, pero en tus pensamientos apenas estás avanzando hacia el escenario de la ciencia ”.*

*Lord Kelvin.*

## *Declaración de Autoría:*

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 2 de la Universidad de las Ciencias Informáticas (UCI) a hacer uso del mismo en su beneficio. Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_ del año \_\_\_\_.

---

**Firma del Autor**  
Cecilia Oro Rivera

---

**Firma del Autor**  
Leidys de la Luz Paredes

---

**Firma del Tutor**  
Ing. Aymara Marín Díaz

## Agradecimientos:

*Ante todo quiero agradecer a mis padres, quienes han estado siempre a mi lado siendo la mano enérgica y a la vez amiga que me ha salvado de las caídas, y los que siempre como Pepe Grillo han intentado ser la fuerte conciencia de su pequeña rebelde.*

*A mi hermano por no dejarme escoger nunca el primer camino sin pensar antes en millones de alternativas. Sorry por no hacerte mucho caso antes.*

*A Yanila Suárez Llopiz, sin tus esfuerzos y desvelos junto a los de mi herma, no hubiese llegado aquí. Gracias mil a los dos.*

*A los amigos adquiridos durante todos estos años de estancia en la UCA, que me han hecho sentir como en casa.*

*A mi tutora Aymara Marín Díaz por confiar en mí y dejarme ser.*

*Cecilia Oro Rivera*

*Gracias le doy primeramente a Dios por permitirme culminar con éxito este sueño, así como por las grandes bendiciones que de su mano he recibido a lo largo de mi vida. Se que sin su ayuda nada hubiese podido conseguir.*

*Agradezco a mi mamá por su infinito amor, por su paciencia y apoyo incondicional, sin los cuales gran parte de lo que soy no sería posible. Gracias mamá por estar siempre a mi lado y por el aliento constante en los momentos más difíciles, por tu disposición a escuchar mis problemas, frustraciones, sueños, anhelos, etc. Te agradezco por escuchar mis pláticas sobre los problemas que tenía que resolver, aunque no entendieras de lo que te estaba hablando. Te doy las gracias por las palabras de motivación, de orientación y de confianza sin las cuales mucho de esto no sería posible. Pero por sobre todo, te doy las gracias por traerme al mundo, aun cuando casi te costó la vida y porque cada día me entregas una parte importante de tu ser. Esta tesis es para ti, porque sé que es la culminación del excelente trabajo que llevas realizando por más de 23 años. Gracias, mil gracias.*

*Gracias a mi familia adorada por su apoyo y paciencia en los momentos difíciles, por tener siempre la frase adecuada para hacerme reír. Gracias por ser como son y por todo el amor que me brindan. Los quiero mucho, no lo olviden.*

*Agradezco a mi amigo Ernesto por su apoyo incondicional durante este tiempo y porque ha traído una nueva luz a mi vida. Gracias por todos los bellos momentos que hemos vivido juntos y por las constantes palabras de ánimo que me motivaron a seguir adelante.*

*Una gratitud especial a la Revolución y a mis profesores en la UCI, cuyo conocimiento contribuyó a ampliar y consolidar mi visión sobre las ciencias de la computación. Sin duda, no podré pagar sus valiosas enseñanzas.*

*No puedo olvidar a mis compañeros y amigos, los cuales también contribuyeron para hacer más agradable y sencilla mi estancia en la UCI. Gracias a cada uno de ellos por su ayuda, orientación y amistad.*

*Leidys de la Luz Paredes*

## *Dedicatoria:*

*A mis padres quienes con su ejemplo me han guiado por el camino correcto que se ha de seguir en la vida y que confiaron en mí todo el tiempo.*

*A mi hermano por ser mi mayor crítico y mi bastón.*

*A mis bebés: Dahiana y Juan Diego Oro*

*A los amigos que han estado para mí en los momentos difíciles y buenos de mi vida.*

*Cecilia Oro Rivera*

*A Dios sea la gloria. Gracias te doy Señor por permitirme ver realizado este sueño.*

*A mi madre adorada, por su gran e inmenso amor, que sin duda jamás podre pagar, por ser simplemente mi mejor amiga y la persona que más quiero en este mundo. A mi familia preciosa, por su gran cariño y amor, y por todos los grandes momentos que hemos vivido juntos.*

*A Ernesto, mi gran amigo, por toda su comprensión y apoyo que han hecho de mi vida algo maravilloso.*

*Leidys de la Luz Paredes*

## ***Resumen:***

La industria del software es un sector donde el concepto de calidad ha tenido un gran impacto. En primer lugar, el desarrollo de software es una actividad relativamente nueva con respecto a otras actividades productivas o industriales en donde la calidad ya se ha estudiado. En segundo lugar, la demanda de software y lo complejo del producto en sí, parecen crecer a mayor velocidad que las metodologías para su desarrollo, el personal capacitado y las herramientas de apoyo; lo que convierte a la evaluación y obtención de la calidad en el mismo, desde los años 70, en la mayor preocupación de los ingenieros, especialistas y comercializadores del sector. Por lo que en el intento de darle cumplimiento de alguna forma a estos dos retos algunas empresas dedicadas a su fabricación han establecido procesos de medición como elemento estratégico.

La definición de un proceso de medición implica seleccionar métricas adecuadas que te permitan obtener un resultado cuantitativo, real y objetivo del elemento o criterio definido a medir, de manera que permita controlarlo y mejorarlo, lo que contribuye favorablemente al logro de la calidad.

Esta es la razón que le dio surgimiento a esta investigación, la propuesta de un sistema de métricas para el control del proyecto IMAC que perfeccionará la gestión de la calidad de los procesos de desarrollo en el software SIGAC durante todo su ciclo de vida.

### **Palabras Clave:**

métricas, calidad, proceso de medición, software.

## *Summary:*

The industry of the software is a sector where the concept of quality has had a great impact. In the first place, the software development is a relatively new activity with regard to other productive or industrial activities where the quality has already been studied. In second place, the software demand and the complexity of the product seem to grow faster than the methodologies for their development, the trained personnel and the support tools, it transforms to the evaluation and the obtaining of the quality into the same in the biggest concern for engineers, specialists and men of business of the sector. For this reason some companies dedicated to their production have established mensuration processes like strategic elements in the intent of giving execution in some way to these two challenges.

The definition of a mensuration process implies to select metric appropriate that allow you to obtain a quantitative, real and objective result of the element or defined approach to measure, so that you can control it and improve it, contributing favorably to the achievement of the quality.

This is the reason that gave emergence to this investigation, the proposal of a system of metric for the control of the project IMAC that will perfect the management of the quality of the development processes in the software SIGAC during all their cycle of life.

### **Keywords:**

metrics, quality, mensuration process, software

## Índice de Contenido:

<i>Frase:</i> .....	2
<i>Declaración de Autoría:</i> .....	3
<i>Agradecimientos:</i> .....	4
<i>Dedicatoria:</i> .....	7
<i>Resumen:</i> .....	8
<i>Summary:</i> .....	9
<i>Introducción:</i> .....	17
<i>Capítulo 1: Fundamentación teórica</i> .....	23
<i>1.1 Introducción</i> .....	23
<i>1.2 Calidad del Software, Objetivo Importante.</i> .....	23
1.2.1 ¿Existen diferencias entre la calidad de un producto de software y otro de fabricación industrial? .....	23
1.2.2 Gestión de la Calidad. ....	26
1.2.3 Sistemas de gestión de la calidad. ....	26
1.2.4 Control de la Calidad. ....	27
1.2.5 Aseguramiento de la Calidad. ....	27
1.2.6 Factores que determinan la calidad. ....	28
1.2.7 Criterios que determinan calidad. ....	30
1.2.8 Métricas, Medida e Indicadores. La Medición como acción. ....	31
<i>1.3 Métricas.</i> .....	33
1.3.1 Clasificación de las Métricas. ....	33
1.3.2 ¿A qué aplicarle métricas durante el desarrollo del software? .....	34
1.3.3 Reglas básicas para evitar el problema del factor humano. ....	38
1.3.4 Características que debe poseer una métrica del software. ....	39
1.3.5 Análisis del comportamiento de las métricas para el software en el mundo. ....	41

1.3.6 Las métricas del software en la UCI.....	42
<b>1.4 ¿Qué son las Normas o Estándares?</b> .....	42
1.4.1 ¿Qué es la IEEE?.....	43
1.4.2 ¿Qué es la ISO? .....	43
<b>1.5 CMMI</b> .....	44
1.5.1 Áreas de Proceso de Nivel 2 .....	44
1.5.2 CMM-CMMI: Gestión de Requisitos o Requerimientos .....	45
1.5.3 CMM-CMMI: Planificación de proyectos.....	46
1.5.4 CMM-CMMI: Monitorización y Control de proyectos.....	46
5.5 CMM-CMMI: Medición y Análisis.....	47
1.5.6 CMM-CMMI: Aseguramiento de la calidad .....	47
1.5.7 CMM-CMMI: Gestión de la configuración.....	48
<b>1.6 El proceso de la investigación científica</b> .....	48
1.6.1 La Recolección de los datos.....	48
1.6.2 Algunos conceptos fundamentales para continuar.....	50
1.6.3 ¿Qué necesito para realizar la encuesta?.....	51
1.6.4 Método Delphi: Método de Expertos.....	52
1.6.5 Técnica de Pareto: (La regla del 80-20). .....	54
<b>Capítulo 2: Métricas para el Software</b> .....	55
<b>2.1 Introducción</b> .....	55
<b>2.2 La Encuesta: Análisis y Procesamiento de los Datos</b> .....	56
2.2.1 Selección de la Población y la Muestra. ....	56
2.2.2 ¿Por qué realizarle la encuesta al administrador de calidad?.....	57
2.2.3 La encuesta .....	58
2.2.4 Análisis de los Datos de la Encuesta.....	61
2.2.4.1 Experiencia en el uso de métricas de software en los proyectos.....	61
2.2.4.2 Técnicas y Herramientas para gestión de las métricas. ....	63

2.2.4.3 Grado de importancia conferido al conjunto de métricas propuesto.....	63
<b>2.3 Definición de la Propuesta de Métricas .....</b>	<b>68</b>
2.3.1 Tiempo Real .....	68
2.3.2 Madurez de las pruebas .....	69
2.3.3 Magnitud del Retraso .....	70
2.3.4 Identificación de Riesgos .....	71
2.3.5 Cobertura de las pruebas .....	72
2.3.6 Complejidad Estimada.....	73
2.3.7 Integridad de la Implementación Funcional.....	75
2.3.8 Porcentaje de Defectos por tipo .....	76
2.3.9 Esfuerzo de desarrollo del software (personas-mes).....	77
2.3.10 Efectividad en la Mitigación de Riesgos .....	78
2.3.11 Adecuación funcional .....	79
2.3.12 Restaurabilidad.....	79
2.3.13 Productividad.....	80
2.3.14 Defectos Removidos .....	82
2.3.15 Índice de Defectos .....	83
<b>2.4 Conclusiones .....</b>	<b>85</b>
<b>Capítulo 3: Estudio del Caso Real (IMAC):.....</b>	<b>87</b>
<b>3.1 Introducción .....</b>	<b>87</b>
<b>3.2 Proceso de Evaluación. ....</b>	<b>88</b>
3.2.1 Resultados de la Aplicación de la Métrica Complejidad Estimada.....	88
3.2.2 Resultados de la Aplicación de la Métrica Esfuerzo de desarrollo de (personas-mes).....	89
3.2.4 Resultados de la Aplicación de las Identificación de los Riesgos, Efectividad de la mitigación de riesgos..	91
3.2.5 Resultados de Aplicación de la Métrica de Productividad. ....	98
3.2.6 Resultados de Aplicación de las Métricas de Cobertura de las Pruebas, Madurez de las Pruebas y Restaurabilidad.....	103

3.2.7 Resultados de Aplicación de las Métricas de Adecuación Funcional y Integridad de la Implementación Funcional. ....	104
3.2.8 Veredicto conclusivo .....	105
<b>3.3 Conclusiones</b> .....	107
<b>Conclusiones Generales</b> .....	108
<b>Recomendaciones</b> .....	109
<b>Referencias Bibliográficas</b> .....	110
<b>Bibliografía</b> .....	112
<b>Glosario de Términos</b> .....	114
<b>Anexos</b> .....	117

## *Índice de Tablas:*

Tabla 1: Factor 1 .....	59
Tabla 2: Factor 2 .....	59
Tabla 3 Factor 3 .....	60
Tabla 4: Experiencia de los Administradores de Calidad.....	61
Tabla 5: Importancia que le confieren los expertos a Selección de Métricas Inicial y Aplicación de dichas Métricas en los Proyectos .....	65
Tabla 6: Cálculo de Pareto .....	66
Tabla 7: Rango de Análisis de la Complejidad Estimada de un Sistema.....	74
Tabla 8: Rango de Control de Riesgos. ....	78
Tabla 9: Cálculo de la Complejidad Estimada .....	88
Tabla 10: Cálculo del Esfuerzo de desarrollo de (personas-mes).....	89
Tabla 11: Defectos Encontrados en el IMAC .....	90
Tabla 12: Listado de Riesgos.....	94
Tabla 13: Soluciones Propuestas para la Mitigación de los Riesgos.....	96
Tabla 14: Efectividad de las Soluciones Propuestas para los Riesgos. ....	97
Tabla 15: Comportamiento de la Productividad en el Mes de Febrero. ....	98
Tabla 16: Comportamiento de la Productividad en el Mes de Marzo. ....	99
Tabla 17: Comportamiento de la Productividad en el Mes de Abril .....	99
Tabla 18: Comportamiento de la Productividad en el Mes de Mayo.....	100
Tabla 19: Resumen General de las Tareas del Proyecto en el Mes de Febrero. ....	100
Tabla 20: Resumen General de las Tareas del Proyecto en el Mes de Marzo.....	100
Tabla 21: Resumen General de las Tareas del Proyecto en el Mes de Abril. ....	101
Tabla 22: Resumen General de las Tareas del Proyecto en el Mes de Mayo.....	101

Tabla 23: Contenedora de los Resultados de las Métricas de Cobertura de las Pruebas, Madurez de las Pruebas y Restaurabilidad.....104

Tabla 24: Contenedora de los Resultados de las Métricas de Adecuación Funcional y Integridad de la Implementación Funcional.....105

## *Índice de Figuras:*

Figura 1 : Factores que determinan la Calidad (Modelo de McCall) .....	30
Figura 2: Relación entre Medición, Medida y Métrica .....	33
Figura 3: Total de Proyectos en la UCI.....	56
Figura 4: Estado de Familiarización con las Métricas. ....	62
Figura 5: Aplicación de las Métricas en los proyectos.....	62
Figura 6: Estado de Empleo de Técnicas y Herramientas para Gestionar las Métricas en los Proyectos. ....	63
Figura 7: Gráfica de Pareto. ....	67
Figura 8: Curva Riesgo Aceptable .....	91
Figura 9: Gráfica de Productividad del cuatrimestre (febrero-mayo). ....	101
Figura 10: Tiempo Estimado y Real. ....	102

## ***Introducción:***

Uno de los problemas que se afrontan actualmente en la esfera computacional es la calidad del software. Desde la década del 70, este tema ha sido motivo de preocupación para especialistas, ingenieros, investigadores y comercializadores de software, los cuales han realizado gran cantidad de investigaciones al respecto con dos objetivos fundamentales:

- Obtener un software con calidad.
- Evaluar la calidad del software.

La Calidad del Software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia, la cual plantea un adecuado balanceo de eficiencia, confiabilidad, facilidad de mantenimiento, portabilidad, facilidad de uso, seguridad e integridad.

Sobre lo que es calidad de software se han escrito numerosas definiciones sin llegar a un concepto único. Siendo el más cercano a definirla el ingeniero de software Roger S. Pressman, en su libro "Ingeniería de Software. Un Enfoque Práctico", que define a la calidad del software desde el punto de vista ingenieril como:

*"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"* (S.Pressman, 2005)

La Calidad del Software (CS) es una disciplina más dentro de la Ingeniería del Software, que implica la utilización de metodologías o procedimientos y definir normas o estándares genéricos para el desarrollo

del software. Los procedimientos pueden variar en cada proyecto, pero lo importante es que estén escritos, personalizados, adaptados a los procesos del mismo y que sean cumplidos; de manera que se unifique la filosofía de trabajo en aras de producir software de la mejor calidad posible, que cumpla, o supere las expectativas de los usuarios.

Los procesos para el control de la calidad del software en los proyectos implican definir los parámetros, indicadores o criterios de medición. Estas mediciones se realizan para garantizar que las restricciones y objetivos del proyecto puedan ser satisfechas ya que como bien dijese Tom De Marco: "*Usted no puede controlar lo que no se puede medir*"(Marco, 1995).

En el software se miden los atributos propios del mismo, descomponiéndose un atributo general en otros más simples de medir. Estos atributos medibles son las bases para la calidad, el control y el perfeccionamiento de la productividad.

En general, la medición persigue tres objetivos fundamentales: "*ayudar a entender qué ocurre durante el desarrollo y el mantenimiento, permitirnos controlar qué es lo que ocurre en nuestros proyectos y poder mejorar nuestros procesos y nuestros productos*" (S.Pfleeger, 1997)

Es válido y necesario decir entonces que el proceso de medición por su importancia, no puede ser tomado a la ligera. Es vital su análisis y la adecuada selección de las métricas que se van a utilizar para su realización. Siendo las métricas medidas que aportan información vital del ciclo de vida del proyecto, como por ejemplo: fases en que se encuentra el proyecto, el estado en que está el mismo, el producto, el proceso y los recursos. Permitiendo la retroalimentación tanto para los desarrolladores, como para el desarrollo del sistema, ya que proporciona la oportunidad de realizar acciones correctivas basándose en los nuevos datos aportados. Igualmente la obtención de estos datos es valioso para la realización de futuros proyectos, dado la importancia que tienen para ellos el esfuerzo, tiempo de desarrollo, costo, posibles errores, recursos necesarios y tamaño para la conceptualización de los mismos.

Las métricas a emplear durante el ciclo de vida del software deben tener en cuenta el entorno o contexto en el que serán aplicadas, siendo importante que sean seleccionadas con cuidado y que su grado de

complejidad no sea elevado, pues puede conducir a cierto nivel de ambigüedad en las definiciones, propiedades y asunciones de las mismas, haciendo que su uso sea difícil, la interpretación peligrosa y los resultados contradictorios. Para evitarlo es necesario contar con un método de definición de métricas y con una base para su formalización.

La Universidad de las Ciencias Informáticas (UCI) es la primera universidad creada al calor de la Batalla de Ideas, con la misión de formar un profesional atemperado a los tiempos nuevos y altamente calificado en ciencias informáticas. Desde sus inicios esta universidad ha vinculado la docencia a la producción de software; y su objetivo ha sido siempre obtener productos de software con la mayor calidad posible, por lo que ha realizado algunos estudios sobre las métricas. Pero aún no ha logrado definir métricas específicas para aplicarlas en todos los proyectos productivos que en ella existen, dado que este es un tema relativamente nuevo que actualmente se encuentra en investigación por el centro de calidad de la UCI y abarca un área relativamente extensa, ya que cada proyecto posee características diferentes. Para darle solución a esta situación cada proyecto existente en la universidad investiga individualmente sobre las métricas posibles a aplicar en su marco de trabajo.

Uno de estos proyectos es el de Informatización del Ministerio de Auditoría y Control (IMAC), el cual aún no ha logrado definir las métricas correctas a emplear. Pues aunque se realizó un estudio previo titulado Propuesta de Métricas para el proyecto SIGAC, en la cual se propusieron un conjunto de métricas, sólo se aplicó una primera iteración de la misma, no dándole continuidad al ciclo. Las razones fueron las siguientes:

- Cambio de Línea de Trabajo en el Proyecto.
- Cambio de la Concepción Inicial.
- Cambio del Equipo de Trabajo.
- La propuesta no logró abarcar todos los aspectos medibles del IMAC.

La no aplicación de métricas en el proyecto ha generado que exista en el mismo:

- Desconocimiento total del estado del proyecto por parte del equipo de trabajo.

- No se puede administrar el proyecto.
- No se puede caracterizar, evaluar, predecir su desarrollo y mejorarlo.
- La administración no puede comprender el estado de su trabajo.

Motivos que hacen necesaria la propuesta de un sistema de métricas capaz de darle solución a la situación imperante en el IMAC. El cual ante el presente estado en que se encuentra, solicita una investigación sobre el tema. Dando lugar al presente trabajo investigativo, que tiene como problema científico: ¿cómo evaluar el desarrollo del proyecto IMAC mediante el uso de métricas del software?; como objeto de estudio: métricas establecidas para el proceso de desarrollo de software y como campo de acción: métricas establecidas para el proceso de desarrollo del software que se ajusten a los criterios de calidad definidos en el proyecto.

Para dar solución al problema científico antes expuesto se plantea el siguiente objetivo general de la investigación: establecer métricas que permitan la evaluación de la calidad del proyecto IMAC, cuyo cumplimiento se basará en la realización de las siguientes tareas de la investigación:

- Realizar un análisis de las características del proyecto.
- Realizar estudio sobre las métricas de software.
- Entrevistar a trabajadores del departamento de Calidad Central UCI.
- Realizar encuesta a los administradores de calidad de los proyectos UCI.
- Procesar los datos de la encuesta
- Realizar un estudio exhaustivo de las métricas seleccionadas por los expertos.
- Aplicar las métricas en el proyecto.
- Valorar los resultados obtenidos en la aplicación de las métricas en el proyecto.

Se define entonces como idea a defender la siguiente: La aplicación de métricas para evaluar el desarrollo del proyecto IMAC permitirá la toma de acciones correctivas pertinentes que dará como resultado un producto de mayor calidad.

Para la realización de esta investigación se utilizaron los métodos empíricos, métodos teóricos y métodos estadísticos matemáticos.

#### Métodos teóricos

**Análisis y Síntesis:** A través de la investigación sobre las métricas y las herramientas que facilitan el uso de las mismas, se extrajeron los elementos más importantes relacionados con el campo de acción.

**Histórico Lógico:** Permitted constatar teóricamente como ha evolucionado la utilización de las métricas en la UCI.

**Inductivo – deductivo:** Permitted llegar a un grupo de conocimientos generalizadores, tanto desde el análisis de lo particular a lo general, como desde el análisis de elementos generalizadores a uno de menor nivel de generalización.

#### Métodos empíricos

**La encuesta:** Se realizaron encuestas tipo cuestionario a los administradores de calidad pertenecientes a la muestra y encuestas tipo entrevista a los trabajadores del departamento de Calidad Central UCI que atienden Métricas, con el objetivo de obtener información acerca de las métricas que pueden ser utilizadas en los proyectos UCI y el porqué de estas.

#### Métodos estadísticos matemáticos

**Los métodos estadísticos matemáticos:** fueron los que nos permitieron evaluar los resultados de los datos recopilados por las entrevistas.

Este trabajo de tesis está estructurado en tres capítulos, conclusiones generales, recomendaciones, bibliografía y anexos, con un total de 120 páginas, donde los capítulos están dispuestos de la siguiente manera:

Capítulo 1: Fundamentación teórica: durante el desarrollo de este capítulo se hace una síntesis del estado del arte del objeto de estudio, dándole a conocer al lector en un lenguaje lo más claro posible los conceptos y aspectos relevantes asociados al tema de la investigación.

Capítulo 2: Métricas para el Software: durante el desarrollo de este capítulo se realizará un estudio y análisis de los resultados de la encuesta tipo cuestionario aplicada a 30 de los administradores de la calidad de 30 proyectos y la propuesta de métricas a aplicar en el IMAC con su respectiva fundamentación.

Capítulo 3: Estudio del Caso Real (IMAC): durante el desarrollo de este capítulo se realizará un análisis de los resultados de la aplicación de las métricas en el IMAC, dando finalmente una valoración del estado del proyecto.

# ***Capítulo 1: Fundamentación teórica***

## ***1.1 Introducción***

El presente estudio tiene basamento en la propuesta de métricas que brinden al proyecto Informatización del Ministerio de Auditoría y Control (IMAC) un medio para evaluarlo. Acción que proporciona la toma de medidas preventivas y correctivas que permitan la mejora y por consiguiente la obtención de un producto de calidad.

El capítulo por lo antes expuesto será la síntesis del estado del arte de los diversos parámetros asociados al dominio del problema.

## ***1.2 Calidad del Software, Objetivo Importante.***

### **1.2.1 ¿Existen diferencias entre la calidad de un producto de software y otro de fabricación industrial?**

La Calidad de un producto sea software o no, siempre es compleja de evaluar. La razón es simple, la medida de la calidad puede abordarse desde muy diferentes perspectivas y tiene multitud de posibles soluciones. Por esta razón, para hablar de calidad de la forma más objetiva posible, debemos primero definir qué podemos entender por esa calidad, segundo, especificar cómo vamos a evaluarla, tercero, dejar claro qué nivel de calidad deseamos y si la podremos alcanzar.

Sobre lo que es calidad del producto existen varias definiciones dadas por especialistas de las ciencias, algunas de ellas son las siguientes:

(Deming, 1982) propuso la idea de la calidad como: *“conformidad con requisitos y confiabilidad en el funcionamiento”*.

(Juran, 1995) dice brevemente: *“Quality is fitness for use”*, o sea es la adecuación del producto al uso, suponiendo un producto libre de deficiencias, cuyas características permiten la satisfacción del usuario.

(Crosby, 1987) pone énfasis en la prevención y dice *“cero defectos”*.

Ahora bien, cuando se habla de calidad del producto de software no puede verse en su totalidad del mismo modo, se diferencia de la calidad de otros productos de fabricación industrial, ya que el software tiene ciertas características especiales. A las cuales hace referencia la Ing. Licet Gutiérrez Mompié en su artículo *“Ideas para la concepción de un modelo de evaluación de calidad de software educativo”*, citadas a continuación (Mompié, 2007):

- *“El software es un producto mental, no restringido por las leyes de la física o por los límites de los procesos de fabricación. Es algo abstracto, y su calidad también lo es.*
- *Se desarrolla, no se fabrica. El coste está fundamentalmente en el proceso de diseño, no en la producción. Y los errores se introducen también en el diseño, no en la producción.*
- *El software no se deteriora con el tiempo, no es susceptible a los efectos del entorno y su curva de fallos es muy diferente a la del hardware. Todos los problemas que surjan durante el mantenimiento estaban desde el principio y afectan a todas las copias del mismo; no se generan nuevos errores.*
- *Es artesanal en gran medida, se construye, en vez de ser ensamblando componentes existentes y ya probados, lo que dificulta aún más el control de su calidad.*
- *Aunque se ha escrito mucho sobre su reutilización, hasta ahora se han conseguido pocos éxitos tangibles.*

- *El mantenimiento del software es mucho más complejo que el mantenimiento del hardware. Cuando un componente de hardware se deteriora se sustituye por una pieza de repuesto, pero cada fallo en el software implica un error en el diseño o en el proceso mediante el cual se tradujo el diseño en código de máquina ejecutable.*
- *Es engañosamente fácil realizar cambios sobre un software, pero los efectos de estos cambios se pueden propagar de forma explosiva e incontrolada.*
- *Como disciplina, el desarrollo de software es aún muy joven, por lo que las técnicas de las que disponemos aún no son totalmente efectivas o no están totalmente calibradas.*
- *El software con errores no se rechaza. Se asume que es inevitable que el software presente errores”.*

Después del análisis de las características del software como producto, estaremos de acuerdo con Watts S. Humphrey cuando plantea que: *“El enfoque principal a cualquier definición sobre calidad de software al ser una definición profesional, debe ser sobre las bases de las necesidades de los usuarios”* (Humphrey, 1995). Siendo el usuario desde el punto de vista de su percepción de consumidor, el que asume o no con conformidad la capacidad del software para satisfacer sus necesidades.

Por lo cual el concepto más completo, desde el punto de vista del usuario sobre la calidad en el software, es el siguiente a nuestro modo de entender:

*“El conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas”* ISO 8402 (UNE 66-001-92)

Y desde el punto de vista ingenieril:

*“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” (S.Pressman, 2005)*

### **1.2.2 Gestión de la Calidad.**

La gestión de la calidad es definida por la (ISO 9000:2000, 2000) como: *“actividades coordinadas para dirigir y controlar una organización”*. Se implanta por medios tales como la planificación de la calidad, el control de la calidad, el aseguramiento (garantía) de la calidad y la mejora de la calidad, en el marco del sistema de calidad.

Es importante destacar que la gestión de la calidad se puede y es recomendable realizarse dentro de la gestión de proyectos de cada proyecto.

### **1.2.3 Sistemas de gestión de la calidad.**

Un sistema de gestión de la calidad es la forma en la que una empresa, organización, institución dirige y controla todas las actividades que están asociadas a la calidad.

La norma (ISO 9000:2000, 2000) define por sistema de gestión de la calidad a: *“la parte del sistema de gestión de la organización enfocada en el logro de resultados, en relación con los objetivos de la calidad, para satisfacer las necesidades, expectativas y requisitos de las partes interesadas, según corresponda.”*

Las partes que componen el sistema de gestión son:

- Estructura organizativa: departamento de calidad o responsable de la dirección de la empresa.
- Cómo se planifica la calidad.
- Los procesos de la organización.

- Recursos que la organización aplica a la calidad.
- Documentación que se utiliza.

Que una organización tenga implantado un sistema de gestión de la calidad, sólo quiere decir que esa empresa gestiona la calidad de sus productos y servicios de una forma ordenada, planificada y controlada.

#### **1.2.4 Control de la Calidad.**

La (ISO 9000:2000, 2000) define al control de la calidad como: *"parte de la gestión de la calidad orientada al cumplimiento de los requisitos de la calidad"*

El control de la calidad tiene como finalidad satisfacer las necesidades técnicas del producto. Se centra en dos objetivos fundamentales:

- Mantener bajo control los procesos.
- Eliminar las causas de los defectos en las diferentes fases del ciclo de vida.

En general son las actividades para evaluar la calidad de los productos desarrollado.

#### **1.2.5 Aseguramiento de la Calidad.**

Se define aseguramiento de la calidad como: *parte de la gestión de la calidad orientada a proporcionar confianza en que se cumplirán los requisitos de la calidad.* (ISO 9000:2000, 2000)

El aseguramiento de calidad del software se debe implementar antes de empezar su desarrollo y no después. Su objetivo principal es la búsqueda de la prevención de los errores, la reducción de costes y la conservación de la competencia. Está presente en:

- Estrategias de prueba multiescala.
- Métodos y herramientas de análisis, diseño, programación y prueba.
- Control de la documentación del software y de los cambios realizados
- Procedimientos para ajustarse a los estándares (y dejar claro cuando se está fuera de ellos)
- Mecanismos de medida (métricas).
- Registro de auditorías y realización de informes.
- Verificación y validación del software a lo largo del ciclo de vida.

Actividades para el aseguramiento de la calidad del software:

- Métricas de software para el control del proyecto.
- Verificación y validación del software a lo largo del ciclo de vida.
- Incluye las pruebas y los procesos de revisión e inspección.
- La gestión de la configuración del software.

### **1.2.6 Factores que determinan la calidad.**

Los factores que determinan la calidad del software se pueden categorizar en dos grandes grupos:

- Factores que se pueden medir directamente (por ejemplo: defectos por puntos de función).
- Factores que se pueden medir sólo indirectamente (por ejemplo: facilidad de uso o de mantenimiento).

Según McCall y sus colegas los factores que afectan la calidad del software se centralizan en tres aspectos importantes: sus características operativas, su capacidad de cambio y su adaptabilidad a nuevos entornos.

Refiriéndose a los factores de la figura 1, McCall proporciona las siguientes descripciones (McCall, 1977):

- *Operaciones del producto: características operativas*
  - *Corrección: grado en que un programa satisface su especificación y logra los objetivos marcados por el usuario. (¿Hace lo que se le pide?).*
  - *Fiabilidad: grado en que se puede esperar que un programa lleve a cabo las funciones esperadas con la precisión requerida. (¿Lo hace de forma fiable todo el tiempo?).*
  - *Eficiencia: cantidad de recursos de computadoras y de código requeridos por el programa para realizar sus funciones con los tiempos de respuesta adecuados. (¿Qué recursos hardware y software necesito?).*
  - *Integridad: grado en que puede controlarse el acceso al software o a los datos por usuarios no autorizados. (¿Puedo controlar su uso?).*
  - *Facilidad de uso: esfuerzo necesario para aprender, utilizar, preparar las entradas e interpretar las salidas de un programa. (¿Es fácil y cómodo de manejar?).*
  
- *Revisión del producto: capacidad para soportar cambios.*
  - *Facilidad de mantenimiento: esfuerzo requerido para localizar y arreglar un error en un programa. (¿Puedo localizar los fallos?).*
  - *Flexibilidad: esfuerzo requerido para modificar un programa. (¿Puedo añadir nuevas opciones?).*
  - *Facilidad de prueba: esfuerzo requerido para probar un programa de forma que se asegure que realiza la función requerida. (¿Puedo probar todas las opciones?).*
  
- *Transición del producto: adaptabilidad a nuevos entornos.*

- *Portabilidad: esfuerzo requerido para transferir un programa desde un entorno HW y/o SW a otro. (¿Podré usarlo en otra máquina?).*
- *Reusabilidad: grado en que un programa o componente SW se puede reutilizar en otras aplicaciones. (¿Podré utilizar alguna parte del software en otra aplicación?).*
- *Interoperabilidad: esfuerzo requerido para acoplar un sistema con otras aplicaciones o sistemas. (¿Podrá comunicarse con otras aplicaciones o sistemas informáticos?).*



Figura 1 : Factores que determinan la Calidad (Modelo de McCall)

### 1.2.7 Criterios que determinan calidad.

Se entiende por criterio de calidad a la: "condición que debe cumplir una determinada actividad, actuación o proceso para ser considerada de calidad." (Osorio, 2009). Es decir qué perseguimos, cuál es el objetivo, qué pretendemos teniendo en cuenta aquellas características que mejor representan (siempre que puedan medirse) lo que deseamos lograr.

Un buen criterio debe reunir los siguientes requisitos:

- Ser explícito, es decir debe dejar muy claro y sin lugar a dudas a qué se refiere, qué se pretende. Debe estar expresado con claridad y objetividad.
- Aceptado por los diferentes interesados (productores, clientes, etc.), siempre es deseable que todos los implicados acepten el criterio y que se comprometan a alcanzarlo.
- Elaborado en forma participativa, la mejor forma de lograr que sea aceptado es que en su elaboración participe el mayor número de personas posible.
- Comprensible, todos deben entender sin lugar a dudas lo mismo.
- Fácilmente cuantificable, de lo contrario ¿cómo vamos a saber si lo alcanzamos?
- Debe ser flexible, capaz de adaptarse a cambios difícilmente previsibles.
- Aceptable por el cliente, que al fin y al cabo es quien juzgará lo acertado de los criterios de calidad.

### **1.2.8 Métricas, Medida e Indicadores. La Medición como acción.**

Muchas veces nos encontramos que los términos: métrica, medida e indicador son usados como vocablos análogos generando confusión en su uso. Sin embargo aunque están estrechamente relacionados no son sinónimos.

El glosario de términos de estándares (IEEE,1990) define métrica como: *una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado.*

En el software métrica según (Pfleeger, 1997) es: *“una asignación de un valor a un atributo de una entidad de software, ya sea un producto o un proceso”*. Es válido aclarar, para no dar paso a futuras interpretaciones erróneas y en eso nos hace énfasis (Zuze, 1998) *“que toda métrica es una medida, pero*

*una medida no tiene porque ser una métrica. El término adecuado es “medida” a pesar de que ambos se usan indistintamente”.*

En cuanto a indicador podemos decir que es una: *“medida que proporciona un estimado o una evaluación de atributos especificados derivados de un modelo con respecto a necesidades de información definidas.”*(ISO/IEC 15939 2002), *“que puede usarse para estimar atributos de calidad del software o para estimar atributos del proceso de la producción. Ellos son medidas indirectas de los atributos”* (NC ISO/IEC 9126 -1:2005 2005 )

Entonces medida sería el *“Número o categoría asignada a un atributo de una entidad haciendo una medición”* (NC ISO/IEC 9126 -1:2005 2005 )

Para llegar entonces a la medición seguimos con (Fenton, 1997) nos dice que: *“la medición es un proceso por el cual, se debe asignar números o símbolos a atributos y entidades en el mundo real, de tal modo de describirlas de acuerdo a reglas definidas claramente.”*

Para concluir y dejar claro los términos haremos este pequeño análisis:

- La medida captura una característica individual.
- La medición permite capturar dicha característica.
- La métrica permite relacionar y comparar mediciones.
- Las métricas son el fundamento de los indicadores.
- Un indicador es una métrica o combinación de métricas que proporcionan una visión profunda del proceso del software, del proyecto de software o del producto en sí.
- Las medidas no sirven para comparar, necesitamos métricas.
- Una medida proporciona una indicación cuantitativa de la extensión, cantidad, dimensiones, capacidad o tamaño de algunos atributos de un proceso o producto.

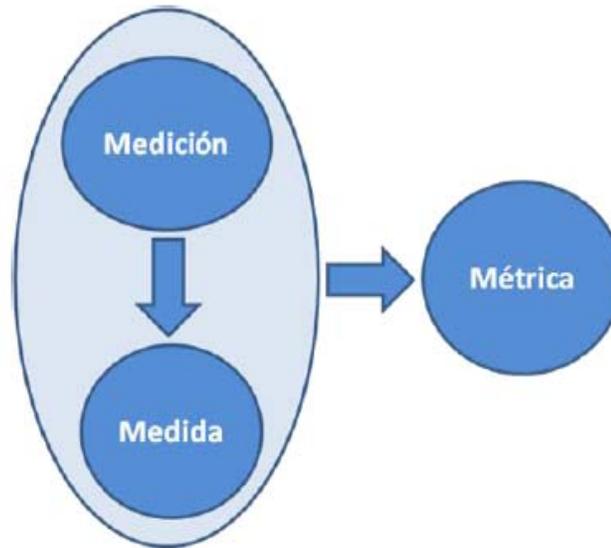


Figura 2: Relación entre medición, medida y métrica

## ***1.3 Métricas.***

### **1.3.1 Clasificación de las Métricas.**

Las métricas según (Lafuente, 2000) se clasifican en directa, indirecta, interna, externa, objetiva y subjetiva. Donde:

*Métrica Directa (Lafuente, 2000): es el resultado de una correspondencia o mapeo directo entre un atributo de un ente (del dominio empírico) y el valor (dominio numérico), es decir, va de un atributo a un número, y sirve como referencia para describir y evaluar aspectos y situaciones del mundo empírico.*

*Métrica Indirecta (Lafuente, 2000): es lo resultante de una correspondencia o mapeo entre relaciones de uno o más atributos, o de un atributo compuesto (del dominio empírico) y el nuevo valor (del dominio*

numérico), que sirve como referencia para describir y evaluar características y/o subcaracterísticas y/o atributos compuestos de un ente del mundo empírico.

Las métricas indirectas también se conocen por métricas derivadas.

Métrica Interna (Lafuente, 2000): es un valor numérico o nominal del atributo que siempre involucra al ente en sí, ya sea obtenido por una métrica directa o indirecta.

Métrica Externa (Lafuente, 2000): es el valor resultante del atributo al aplicar una métrica indirecta ya que siempre involucra al ente y su comportamiento con el entorno.

Métrica Objetiva (Lafuente, 2000): es el valor resultante del atributo de un ente, comprobable, independiente de juicio o subjetividad humana. Sin embargo existen grados de objetividad.

Métrica Subjetiva (Lafuente, 2000): es un valor numérico que siempre involucra el juicio humano por medio de heurísticas o criterios de preferencia directa.

### **1.3.2 ¿A qué aplicarle métricas durante el desarrollo del software?**

En el software se le aplican métricas al producto, proceso, recursos, y al proyecto como entidad rectora. Estas métricas están enfocadas a características específicas, que se desglosan a continuación:

*“Las métricas del producto se centran en las características del software y no en cómo fue producido. Un producto no es sólo el software o sistema funcionando sino también los artefactos, documentos, modelos, módulos, o componentes que lo conforman, por tanto, las métricas del producto deben hacerse sobre la base de medir cada uno de estos” (Nuñez, 2008)*

Según (Kan, 2000) *“las métricas sobre el producto describen características como el tamaño, complejidad, rasgos del diseño, rendimiento y nivel de calidad.”*

(Kruchten, 2003) en su libro del The Rational Unified Process (RUP) desglosa estas características a medir en el producto de la siguiente manera:

- *Tamaño: las métricas del tamaño del producto se refieren generalmente al volumen del producto desarrollado. Incluyen líneas de código (LOC), número de ficheros, páginas de la documentación.*
  
- *Calidad:*
  - *Defectos: indicadores de que un artefacto no funciona como ha sido especificado, o cualquier otra característica indeseable.*
  - *Complejidad de una estructura o un algoritmo: mientras mayor sea la complejidad y más difícil sea de comprender y modificar la estructura del sistema, mayor probabilidad habrá de que falle.*
  - *Acoplamiento: mediciones de cuántos elementos del sistema están interconectados y cuán extensivamente.*
  - *Cohesión: mediciones de cuán bien un elemento o un componente cumple con los requerimientos de tener un sólo y bien definido propósito.*
  - *Primitividad: el grado en el cual las operaciones o métodos de una clase pueden estar compuestos por otros de la misma clase.*
  - *Totalidad: medición de la magnitud en la cual un artefacto cumple con todos los requerimientos (plan / real).*
  - *Rastreabilidad: indicadores de que los requerimientos de determinado nivel se están satisfaciendo por determinados artefactos, o que todos los artefactos tengan razón de existir.*
  - *Volatilidad: el grado de cambio de un artefacto debido a defectos o a cambios en los requerimientos.*
  - *Esfuerzo: medición del trabajo (Unidad de tiempo del personal) que es requerido para producir un artefacto.*

Las métricas del proceso lo que intentan es proporcionar indicadores que lleven a mejoras de los procesos de software, donde los procesos de software pueden definirse como los pasos definidos para determinar quién, cuándo, cómo y dónde; debe hacer cada actividad dentro del proceso de desarrollo de software. Estas métricas brindan un mayor enfoque sobre la calidad lograda como consecuencia de un proceso repetible y ordenado. Es importante conocer que las métricas del proceso dependen esencialmente del entorno de desarrollo, haciéndose necesario medir atributos específicos de los propios procesos, como el tiempo empleado, su coste y el esfuerzo requerido en el desarrollo.

*“La relación entre las medidas de los resultados obtenidos en un proceso y los recursos usados en él permitirá medir la productividad, atributo clave para estimar costo y esfuerzo”* (Escorial, 2006) Algo a recordar y a tener muy en cuenta es la gran variedad de métricas que existen en la actualidad, y éstas deben usarse conforme se ajusten al proceso, no tomándose la ligera su selección.

Las métricas del proceso se caracterizan por:

- El control y ejecución del proyecto.
- Medición de tiempos del análisis, diseño, implementación, implantación y post-implantación.
- Medición de las pruebas (errores, cubrimiento, resultado en número de defectos y número de éxito).
- Medición de la transformación o evolución del producto.

Las métricas para los recursos medirán según (Días, 2008): *los elementos que incluyen a personas (experiencia, habilidades, coste, funcionamiento), los métodos y las herramientas (en términos de efecto sobre productividad y calidad, coste), tiempo, esfuerzo, y presupuesto (recursos consumidos, recursos restantes).*

Se aplican, fundamentalmente, a las horas de labor, el principal recurso del desarrollo del software. Aquí lo que concierne son las horas trabajadas, categorías de trabajo y realización de tareas. Cualquier interrupción tiene tres costos potenciales: la pérdida de tiempo, el tiempo adicional que toma reconstruir el

momento en que fue interrumpido y la probabilidad incremental de cometer errores. Un foco principal de cualquier esfuerzo para mejorar la productividad o el ciclo de tiempo debe identificar y reducir estas distracciones.

Las métricas para el proyecto son de vital importancia ya que ayudan al mejoramiento del mismo. A nivel de proyecto se minimiza la planificación de desarrollo haciendo los ajustes necesarios para evitar retrasos o riesgos potenciales, minimizar los defectos, y por tanto la cantidad de trabajo que ha de rehacerse, lo que ocasiona una reducción del coste global del proyecto, además puede evaluarse la calidad de los productos en el momento actual y cuando sea necesario.

Las principales métricas para el proyecto a tener en cuenta son (Kruchten, 2003):

- *Modularidad: Promedio de daños debido a cambios perfectivos o correctivos en la implementación.*
- *Adaptabilidad: promedio de esfuerzo debido a cambios perfectivos o correctivos en la implementación.*
- *Madurez: tiempo de prueba activo / número de cambios correctivos.*
- *Mantenimiento: mantenimiento productivo / desarrollo productivo.*
- *Progreso del proyecto: debe reportarse basándose en el plan del proyecto desde la perspectiva del valor devengado.*

Estas métricas del proyecto dan paso a las siguientes acciones (Kruchten, 2003):

- *Evaluar el estado del proyecto en curso.*
- *Seguir la pista de los riesgos potenciales.*
- *Detectar las áreas de problemas antes de que se conviertan en "críticas".*
- *Ajustar el flujo y las tareas del trabajo.*

- *Evaluar la habilidad del equipo del proyecto en controlar la calidad de los productos de trabajo del software.*

### **1.3.3 Reglas básicas para evitar el problema del factor humano.**

Cuando algo es medido automáticamente asume importancia. Las personas quieren “verse bien”, por tanto quieren que las métricas los hagan “verse bien”. La mejor forma de evitar el problema del factor humano en el trabajo con las métricas es seguir algunas reglas básicas tales como las que se enuncian a continuación (Wesfall, 2000):

- *No mida a los individuos: el estado del arte en las métricas de software no llega hasta ese punto. Hay que enfocarse en los procesos y en los productos, no en las personas.*
- *Nunca usar las métricas como un “garrote”: la primera vez que se usen las métricas en contra de los individuos o los equipos será la última vez que se obtendrán datos válidos.*
- *No ignorar los datos: una forma segura de matar un programa de métricas es ignorar los datos cuando se toman decisiones. Hay que respaldar a las personas cuando sus reportes estén retrocediendo debido a datos útiles a la organización.*
- *Nunca usar una sola métrica: el software es complejo y multifacético. Un programa de métricas debe reflejar esa complejidad. Debe mantenerse un balance entre los atributos del costo, la calidad y los cronogramas de forma que se satisfagan todas las necesidades de los usuarios. Enfocarse en una única métrica puede causar que el atributo que es medido mejore a costa de otros atributos.*
- *Proveer retroalimentación: proporcionando una retroalimentación regular al equipo sobre los datos que ellos ayudan a coleccionar tiene varios beneficios:*

- *Ayuda a mantener el foco en la necesidad de coleccionar los datos. Cuando el equipo vea que los datos están siendo realmente usados, será más probable que consideren la importancia de la colección de los datos.*
- *Si los miembros del equipo se mantienen informados sobre los detalles específicos de cómo los datos son usados, ellos tendrán menos posibilidades de empezar a sospechar sobre su uso.*
- *Involucrando a los miembros del equipo en el análisis de los datos y en los esfuerzos por mejorar los procesos, se obtienen beneficios de conocimiento y experiencia.*
- *La retroalimentación en problemas e integridad de la colección de datos, ayuda a educar a los miembros del equipo en la responsabilidad de coleccionar los datos. Los beneficios pueden ser datos más consistentes, exactos y oportunos.*
- *Lograr “pertenencia”: Para lograr un sentido de pertenencia tanto en las metas como en las métricas en un programa de medición, se tiene que lograr la participación en la definición de las métricas. Las personas que trabajan con un proceso en su quehacer diario tendrán un conocimiento íntimo de ese proceso, esto les da una perspectiva valiosa de cómo el proceso puede ser mejor medido para asegurar confiabilidad y validez, y cómo interpretar mejor las mediciones resultantes para maximizar la utilidad.*

#### **1.3.4 Características que debe poseer una métrica del software.**

(Doria, 2001) nos dice que hay 5 elementos claves a tener en cuenta a la hora de seleccionar las métricas, estos son:

- *Simple y fácil de calcular: debería ser relativamente fácil de aprender a obtener la métrica y su cálculo no obligará a un esfuerzo o a una cantidad de tiempo inusuales.*
- *Empírica e intuitivamente persuasiva: la métrica debería satisfacer las nociones intuitivas del ingeniero de software sobre el atributo del producto en cuestión (por ejemplo: una métrica que mide la cohesión de un módulo debería aumentar su valor a medida que crece el nivel de cohesión).*
- *Consistente en el empleo de unidades y tamaños: el cálculo matemático de la métrica debería utilizar medidas que no lleven a extrañas combinaciones de unidades. Por ejemplo, multiplicando el número de personas de un equipo por las variables del lenguaje de programación en el programa resulta una sospechosa mezcla de unidades que no son intuitivamente concluyentes.*
- *Independiente del lenguaje de programación: las métricas deberían apoyarse en el modelo de análisis, modelo de diseño o en la propia estructura del programa. No deberían depender de los caprichos de la sintaxis o semántica del lenguaje de programación.*
- *Un mecanismo eficaz para la realimentación de calidad: la métrica debería suministrar al desarrollador de software información que le lleve a un producto final de superior calidad.*

¿Por qué asegurarnos de que las métricas cumplen estas condiciones?

Las métricas deben permitir la mejora de los procesos dentro del equipo, el producto que se desarrolla y el proyecto como entidad rectora. De nada sirve aplicar métricas que en vez de ayudar en la toma de decisiones se conviertan en un problema, ya sea por no alcanzar los objetivos que se persiguen con ellas, porque su interpretación no sea la adecuada, o porque arrojen resultados imprecisos que no puedan ser interpretados por los ingenieros de software.

### **1.3.5 Análisis del comportamiento de las métricas para el software en el mundo.**

En la actualidad en la industria de desarrollo del software el uso de las métricas ha cobrado gran éxito. La causa es que las empresas productoras están reconociendo la importancia que tienen las mediciones para cuantificar y por consiguiente gestionar de forma más efectiva la calidad de los procesos y productos del software. En empresas que se dedican exclusivamente a su desarrollo, se tiene noción de la necesidad de formalizar los mecanismos de estimación, comprendiendo que los registros históricos de antiguos proyectos realizados pueden ayudar a estimar con mayor exactitud el esfuerzo, tiempo de desarrollo, costo, posibles errores, recursos y tamaño para los nuevos proyectos. Es válido aclarar que en ocasiones los resultados de los procesos de medición no son interpretados de la mejor manera, pues aún existen compañías que no tienen una cultura adecuada sobre la medición, desconociendo el alcance de madurez y calidad que pudiera alcanzar el producto final.

Varios estándares y modelos han incluido métricas de software, a continuación se muestran algunos de ellos:

- CMMI: Modelo de Capacidad de Madurez de Integración.
- PSP: Proceso de Software Personal.
- TSP: Proceso de Software en Equipos.
- Metriplica: (Sitio eMetrics). División especializada en medición web.
- SixSigma ( $6\sigma$ ): Metodología que ayuda en la prevención de errores en los procesos industriales. Proporciona la información adecuada a través de las mediciones y el uso de herramientas estadísticas para analizar desviaciones (errores).
- GQM: Objetivos, Cuestiones, Métricas.
- PSM: Mediciones Prácticas de Software y Sistemas.

Sin embargo aunque se reconoce que se usan las métricas a nivel mundial en la mayoría de los casos aún hay un compromiso con su uso, pues se dice que es un problema la gran variedad de métricas que existen y que generalmente no se tiene un conocimiento explícito de sus objetivos y la manera en que puedan aplicarse.

### **1.3.6 Las métricas del software en la UCI.**

Al realizar una entrevista en el departamento de Calidad Central de la UCI se logró constatar que en la actualidad en la universidad no existe un plan de métricas correctamente descrito y detallado para aplicar en los proyectos de producción. Los proyectos por el momento son los encargados de la selección de las métricas para aplicar en su marco de trabajo. Esta situación se prevé que para finales del presente curso ya esté resuelta, pues se está gestionando la obtención de la certificación del nivel 2 de CMMI, lo que genera un intenso trabajo en pos de lograr la calidad de los productos que son desarrollados dentro del centro de altos estudios; y una de las cosas que saldrán de este inmenso esfuerzo es la definición de métricas a aplicar en todos los proyectos que en ella se desarrollan.

Es válido aclarar que aunque se defina para toda la universidad un Plan de Métricas estas serían genéricas y se referirán a aspectos generales que la UCI como polo productivo necesita documentar. Esto quiere decir que cada proyecto puede, y me atrevería a decir que tiene el compromiso de gestionar métricas individuales para aplicar en su marco de trabajo.

## ***1.4 ¿Qué son las Normas o Estándares?***

Norma y estándar son vocablos análogos que según ISO se definen como: *“un conjunto de acuerdos documentados que contienen especificaciones técnicas u otros criterios precisos para ser usados constantemente, como reglas, lineamientos o definiciones de características. Todo esto con la finalidad de asegurar que los materiales, productos, procesos y servicios son óptimos para su propósito”* (Álvarez, 2004)

La estandarización o norma se basa en el trabajo conjunto de todas las partes involucradas: productores, profesionales, usuarios, administración pública, etc. Además recoge los deseos, las propuestas de todas las instituciones relevantes como son los fabricantes, las asociaciones de consumidores, los juristas, los centros de investigación, las entidades de certificación e inspección. Cubren todos los aspectos técnicos relacionados con la información, su producción y gestión. Son coherentes y consistentes. Han sido desarrolladas por comités técnicos bajo supervisión de un organismo especializado. Son aplicables a muchos tipos de organizaciones pequeñas, medianas, grandes, jerárquicas o planas, dinámicas o conservadoras, con recursos magros o amplios, local o internacional, etcétera; por esta razón, la implantación debe iniciar con un análisis de brecha entre lo que existe y lo que se requiere y además conocer las circunstancias internas y el entorno de la organización de tal manera que la norma o estándar se implante de manera ágil, que sea útil, de bajo costo y acorde a la estrategia a utilizar.

#### **1.4.1 ¿Qué es la IEEE?**

Institute of Electrical and Electronics Engineers (IEEE), su interpretación en español sería Instituto de Ingenieros Electrónicos y Eléctricos, la cual es una importante asociación de técnicos y profesionales, con sede en los Estados Unidos, que favorece la investigación en campos diversos, como la tecnología aeroespacial, la computación, las comunicaciones y la tecnología biomédica. Su misión es preservar, investigar y promover la información de las tecnologías eléctricas y electrónicas. Su finalidad es avanzar en la teoría, práctica y aplicación de las tecnologías de la información. Realiza conferencias, publicaciones, cursos de formación, y desarrolla estándares. Promueve la estandarización de normas.

#### **1.4.2 ¿Qué es la ISO?**

International Organization for Standardization (ISO), su interpretación en español sería Organización Internacional para la Estandarización. Es el organismo encargado de promover el desarrollo de normas internacionales de fabricación, comercio y comunicación para todas las ramas industriales a excepción de la eléctrica y la electrónica. Su función principal es buscar la estandarización de normas de productos y seguridad para las empresas u organizaciones a nivel internacional. Es una organización

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

### ***1.5 CMMI.***

El Capability Maturity Model Integration (CMMI) que tiene como interpretación en español Modelo de Integración de Capacidad de Madurez, es un marco de referencia que las organizaciones pueden emplear para mejorar sus procesos de desarrollo, adquisición, y mantenimiento de productos y servicios. Se hará referencia a él, pues la UCI está optando por la certificación del Nivel 2 de dicho modelo. Lo que implica que la propuesta de solución que se proponga como resultado de la investigación debe estar en correspondencia con la misma, para que futuros cambios se ajusten a lo realizado en el proyecto.

CMMI plantea que las organizaciones pueden ubicarse en alguno de cinco posibles niveles de madurez, dependiendo del grado de sofisticación de sus procesos. A su vez, cada nivel de madurez (con excepción del inicial) queda caracterizado por un conjunto de áreas de proceso que agrupan prácticas que, al ser ejecutadas colectivamente, permiten cumplir con algún objetivo que es considerado importante para el modelo. Además utiliza un amplio conjunto de métricas que determina la calidad de cada una de estas áreas claves, obteniéndose una visión precisa del rigor, la eficacia y la eficiencia de la metodología de desarrollo de una organización productora de software.

La investigación se va a centrar en el nivel 2 pues es el nivel por el cual esta optando la UCI.

#### **1.5.1 Áreas de Proceso de Nivel 2**

Según la empresa Axentia: *“en una organización que haya alcanzado este nivel de madurez encontraremos que hay una disciplina para la gestión de proyectos que se mantiene aún en periodos de estrés. Los recursos estarán capacitados para hacer su trabajo, y habrá políticas organizacionales formalmente establecidas, cuyo cumplimiento será monitoreado. Habrá visibilidad de las actividades realizadas, y los proyectos se ejecutarán siguiendo un plan y de acuerdo a un proceso formalmente establecido”* (Axentia , 2006)

Lo antes expuesto se ve directamente materializado en las siguientes áreas de procesos.

- Gestión de Requisitos.
- Planificación de proyectos.
- Monitorización y Control de proyectos.
- Medición y Análisis.
- Aseguramiento de la calidad.
- Gestión de la configuración.

Se comenzará a explicar cada una de las áreas de proceso con un poco más de detalle.

### **1.5.2 CMM-CMMI: Gestión de Requisitos o Requerimientos**

El objetivo de este proceso es gestionar los requisitos de los elementos del proyecto y sus componentes e identificar inconsistencias entre estos requisitos, el plan de proyectos y los elementos de trabajo.

En este proceso se deben gestionar todos los requisitos del proyecto, tanto los requisitos técnicos como los no técnicos. Estos han de ser revisados conjuntamente con su fuente así como con las personas que se encargarán del desarrollo posterior.

### **1.5.3 CMM-CMMI: Planificación de proyectos.**

El objetivo de la planificación de proyectos es establecer y mantener planes que definen las actividades del proyecto.

Las tareas que conlleva la planificación de proyectos son:

- Desarrollar un plan inicial del proyecto.
- Establecer una relación adecuada con todas las personas involucradas en el proyecto.
- Obtener compromiso con el plan.
- Mantener el plan durante el desarrollo del proyecto.

El plan incluye estimación de los elementos de trabajo y tareas, recursos necesarios, negociación de compromisos, establecimiento de un calendario, e identificación y análisis de los posibles riesgos que pueda tener el proyecto.

El plan de proyectos es un herramienta de trabajo viva que se debe actualizar con mucha frecuencia ya que los requisitos cambiarán, habrá que reestimar, habrá riesgos que desaparezcan y otros que surjan y habrán de tomarse acciones correctivas.

### **1.5.4 CMM-CMMI: Monitorización y Control de proyectos.**

El objetivo de la monitorización y control de proyectos es proporcionar una comprensión del estado del proyecto para que se puedan tomar acciones correctivas cuando la ejecución del proyecto se desvíe del plan.

El documento del plan de proyecto es la base para monitorizar las actividades, comunicar el estado y tomar acciones correctivas. El progreso se determina comparando los actuales elementos de trabajo:

tareas, horas realizadas, coste y calendario actual, con los estimados en el plan de proyecto. Una apropiada visibilidad permitirá tomar acciones correctivas antes que el trabajo real se desvíe mucho del plan.

Estas acciones crearán la necesidad de rehacer y ajustar el plan de proyectos.

### **5.5 CMM-CMMI: Medición y Análisis.**

El objetivo de la medición y el análisis es desarrollar y sostener una capacidad de medición que sea usada para 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 y análisis de los mismos de forma que las decisiones que se tomen puedan estar basadas en estos datos.

Este sistema tiene que permitir además:

- Planificación y estimación objetiva.
- Comparar el rendimiento actual contra el rendimiento esperado en el plan.
- Identificar y resolver problemas relacionados con los procesos.
- Proporcionar una base para añadir métricas en procesos futuros.

### **1.5.6 CMM-CMMI: Aseguramiento de la calidad**

El objetivo del aseguramiento de la calidad es proporcionar personas y gestión con el objetivo de que los procesos y los elementos de trabajo cumplan los procesos.

Esto se consigue mediante:

- Evaluar objetivamente la ejecución de los procesos, los elementos de trabajo y servicios contra las descripciones de procesos, estándares y procedimientos.
- Identificar y documentar los elementos no conformes.
- Proporcionar información a las personas que están usando los procesos y a los gestores, de los resultados de las actividades del aseguramiento de la calidad.
- Asegurar de que los elementos no conformes son arreglados.

Esta es un área clave del proceso, que a veces no se le da la suficiente importancia, pero que sin ella no será posible implantar un modelo de calidad.

### **1.5.7 CMM-CMMI: Gestión de la configuración**

El objetivo de la gestión de la configuración es establecer y mantener la integridad de los elementos de trabajo identificando, controlando y auditando dichos elementos. Mediante:

- La identificación de los elementos de trabajo que componen una línea base.
- Controlando los cambios de dichos elementos.
- Proporcionando formas de construir los elementos de trabajo a partir del sistema de control de la configuración.
- Mantener la integridad de las líneas base.
- Proporcionar información precisa de los datos de la configuración a desarrolladores y clientes.

## ***1.6 El proceso de la investigación científica.***

### **1.6.1 La Recolección de los datos.**

Cuando se realiza una investigación es importante que se abarque la mayor cantidad de información posible no limitándose sólo a la literatura que se pueda encontrar sobre la misma, esto le da validez y peso. Es este el motivo por el cual la opinión de los especialistas o simplemente personas asociadas al mundo, que es motivo de inspiración de la investigación, debe ser considerada como información valiosa y para su recopilación se utiliza la encuesta.

Sobre lo que es la encuesta (Baker ,1997) expresa: *“es un método de colección de datos en los cuales se definen específicamente grupos de individuos que dan respuesta a un número de preguntas específicas”* Otra de las definiciones es la dada por la Real Academia Española que define a la encuesta como: *“una averiguación o pesquisa, conjunto de preguntas tipificadas dirigidas a una muestra representativa, para averiguar estados de opinión o diversas cuestiones de hecho”* (Real Academia Española, 2001)

En resumen, las anteriores definiciones indican que la encuesta se utiliza para estudiar poblaciones mediante el análisis de muestras representativas a fin de explicar las variables de estudio y su frecuencia. Ahora bien, existen varios tipos de encuestas, dependiendo de numerosos factores, pero sólo serán manipuladas en la presente investigación las de tipo medio y dentro de estas las encuestas basadas en entrevistas y las de tipo cuestionario.

Una encuesta basada en entrevista es: *“una pieza de la interacción social en la cual una persona responde a otra una serie de preguntas sobre un tópico específico, en sí, representa una interacción cara a cara entre dos o más personas. La entrevista representa una excelente técnica de recolección de la información”* (Thompson, 2006)

La encuesta tipo cuestionario es un: *“instrumento utilizado para la recogida de información, diseñado para poder cuantificar y universalizar la información. Su finalidad es conseguir la comparabilidad de la información”* (Thompson, 2006)

Ya familiarizados con el término encuesta no es de extrañarse que el objetivo general que se persiga en esta investigación con su utilización es la de obtener información sobre nuestro objeto de estudio,

apropiándonos de todo el conocimiento posible de adquirir con las preguntas que se le realizarán a los trabajadores del departamento de calidad de la UCI y los administradores de calidad de los proyectos que serán muestra. Para eso necesitamos definir la población, la muestra y los pasos a seguir para la selección de esta última.

### 1.6.2 Algunos conceptos fundamentales para continuar...

Los conceptos fundamentales que se deben tener en cuenta para enfrentar la investigación son: población, unidad de estudio y muestra.

Se define población como: *“cualquier conjunto de elementos que tengan una o más propiedades en común definidas por el investigador y que puede ser desde toda la realidad, hasta un grupo muy reducido de fenómenos”* (González, 2002)

En primera instancia, la población según (Mora, 1999)

- *Corresponde al objeto del estudio (o de interés) y sobre la cual se pretende generalizar.*
- *Está conformada por todos los que concuerdan con las especificaciones.*
- *Se identifica por los criterios de inclusión.*
- *Es descrita en torno a sus características de contenido, lugar y tiempo.*
- *Resulta delimitada por los criterios de exclusión.*

Las principales características de una población (Mora, 1999) la resume en:

- *Las dimensiones cuantitativas y cualitativas de una población, son establecidas por el investigador de acuerdo con los objetivos propuestos y el nivel de explicación que se pretende con los resultados.*
- *Toda población puede incluirse en otra mayor o subdividirse en otras menores en función de los fines que se persiguen.*

Siempre se debe partir de una definición clara y precisa de la población para evitar que la generalización de los resultados afecte la calidad de su aplicación.

Unidad de estudio: *“son los elementos, fenómenos, sujetos o procesos que integran la población y pueden ser individuos, grupos de personas, hechos, procesos, talleres, turnos de trabajo, empresas, documentos, etc”* (González, 2002)

Las unidades que integran la población se determinan de acuerdo con la naturaleza de la investigación y el diseño teórico elaborado.

Muestra: *“es un grupo relativamente pequeño de unidades de estudio que representa las características de la población”* (González, 2002)

Otra definición es la dada el Dr. Edgardo José Avilés Garay donde dice que muestra es una: *“subcolección de elementos tomados de la población”* (Garay, 2006)

### **1.6.3 ¿Qué necesito para realizar la encuesta?**

Se conoce que se va a realizar una encuesta a una muestra de una población específica, por lo que ahora se necesita definir a cuántos individuos se le va a realizar la encuesta y cómo se a proceder para su selección.

Respecto al tamaño de la muestra a utilizar (Álvarez, 1997) recomienda que: *“cuando no se recurre a procedimiento matemático (como es el caso) una muestra se considera pequeña siempre que sea menor que 30 en poblaciones finitas y cuando el número de elementos de la población es suficientemente grande es recomendable que la muestra nunca sea menor del 10 por ciento de la población”*.

Para la seleccionar los individuos que van a pertenecer a la muestra se necesita realizar un muestreo, que se define por (González, 2002) como: *“técnicas y procedimientos que se utilizan para seleccionar la muestra que representa una población dada.”*

Existen varios tipos de muestreo el que se va a utilizar para la selección de la muestra es casual o incidental que según (Gallego, 2004) se trata de: "un *proceso en el que el investigador selecciona directa e intencionadamente los individuos de la población*". El caso más frecuente de este procedimiento el utilizar como muestra los individuos a los que se tienen fácil acceso.

Ya conociendo cómo seleccionar la muestra y el tipo de muestreo a utilizar se debe definir el método para la obtención de la información, en este caso será el Método de Expertos del Método Delphi.

#### **1.6.4 Método Delphi: Método de Expertos.**

El método Delphi es una metodología de investigación multidisciplinar para la realización de pronósticos y predicciones. "Tiene cuatro etapas sucesivas de envíos de cuestionarios, de los que se sintetizan las respuestas para elaborar la siguiente consulta, hasta llegar a un resultado de consenso.

*Tras un primer test de aproximación, en la segunda consulta se deben volver a responder viendo los resultados de la primera y justificar sus divergencias con el grupo. En la tercera consulta, se pide a cada testados comentar los argumentos que disienten de la mayoría y en el cuarto turno se reclama la opinión sobre el consenso final*" (Torrigo,2005)

Dentro del Método Delphi se va a utilizar el método de expertos que se basa en la consulta a personas que tienen grandes conocimientos sobre el entorno que es caso de estudio. Estas personas exponen sus ideas y finalmente se redacta un informe en el que se indican cuáles son, en su opinión, las posibles alternativas que se tendrán en el futuro.

Los métodos de expertos tienen las siguientes ventajas (Torrigo,2005) :

- "La información disponible está siempre más contrastada que aquella de la que dispone el participante mejor preparado, es decir, que la del experto más versado en el tema. Esta afirmación se basa en la idea de que varias cabezas son mejor que una.

- *El número de factores que es considerado por un grupo es mayor que el que podría ser tenido en cuenta por una sola persona. Cada experto podrá aportar a la discusión general la idea que tiene sobre el tema debatido desde su área de conocimiento”.*

Presenta según (Dias, 2008) tres características fundamentales:

- *“Anonimato: durante un Delphi, ningún experto conoce la identidad de los otros que componen el grupo de debate. Esto tiene una serie de aspectos positivos, como son:*
  - *Impide la posibilidad de que un miembro del grupo sea influenciado por la reputación de otro de los miembros o por el peso que supone oponerse a la mayoría. La única influencia posible es la de la congruencia de los argumentos.*
  - *Permite que un miembro pueda cambiar sus opiniones sin que eso suponga una pérdida de imagen.*
  - *El experto puede defender sus argumentos con la tranquilidad que da saber que en caso de que sean erróneos, su equivocación no va a ser conocida por los otros expertos.*
- *Iteración y realimentación controlada: la iteración se consigue al presentar varias veces el mismo cuestionario. como, además, se van presentando los resultados obtenidos con los cuestionarios anteriores, se consigue que los expertos vayan conociendo los distintos puntos de vista y puedan ir modificando su opinión si los argumentos presentados les parecen más apropiados que los suyos.*
- *Respuesta del grupo en forma estadística: la información que se presenta a los expertos no es sólo el punto de vista de la mayoría, sino que se presentan todas las opiniones indicando el grado de acuerdo que se ha obtenido.”*

Es muy importante para un correcto resultado en la aplicación de este método escoger bien a los expertos y definir bien el campo de investigación, con preguntas precisas, cuantificables e independientes.

### **1.6.5 Técnica de Pareto: (La regla del 80-20).**

Para escoger las métricas más significativas según el grado de frecuencia en que los expertos hayan seleccionado la métrica como importante al responder la encuesta, se va a utilizar la técnica de Pareto. Esta técnica separa: los “*pocos vitales*” de los “*muchos triviales*”, es decir le da preferencia al 20% que importa y que produce el 80% de los resultados (Sociedad Latinoamericana para la calidad, 2000) Su empleo es de gran importancia para la toma de decisiones, porque permite ahorrar tiempo, esfuerzo y trabajo a quien le corresponde tomar la decisión.

## ***1.7 Conclusiones.***

En la actualidad las Tecnologías de la Informaciones y la Comunicaciones (TICs) se han convertido en un elemento de articulación universal. Por lo que es normal encontrarnos con una creciente demanda de software, el cual debe ser capaz de satisfacer las necesidades imperativas de los usuarios. Esto implica que se necesite poner especial atención en la calidad del mismo, haciendo vital que se controle, monitoree y se le dé seguimiento sistemático al proceso, producto, recursos y proyecto en general en pos de lograr alcanzar la satisfacción del usuario, y ¿por qué no? Superarla; siendo las métricas las herramientas indiscutibles que nos permitirán medir la calidad del software y así aspirar a su alcance.

## *Capítulo 2: Métricas para el Software*

### *2.1 Introducción.*

En el capítulo anterior se alude a la importancia que tiene en una investigación científica agotar todas las fuentes posibles de conocimiento, más aún cuando el objeto de estudio abarca una extensa área y genera múltiples interpretaciones. Haciendo honor a esta apertura y no estando conformes sólo con la literatura encontrada, se decide hacer una encuesta tipo cuestionario con el fin de apropiarnos del conocimiento disperso en la UCI sobre las métricas. La misma tuvo en cuenta los resultados obtenidos en la encuesta de la tesis titulada “Propuesta de Métricas para el proyecto SIGAC” para su conformación. Esta se realizó utilizando la metodología de investigación multidisciplinar Delphi, específicamente el Método de Expertos. Inicialmente se proyectó la idea de darle seguimiento, aplicando su segunda iteración, pero se decidió que para lograr mejores resultados y ampliar el campo de conocimiento se llevará a cabo una selección de expertos más comprometidos con el objeto de estudio. Siendo así seleccionado como universo o población a encuestar, los administradores de calidad de los proyectos productivos de la Universidad. De dicha encuesta se retomó la idea de la pregunta número 6, agregándole nuevas métricas.

Al igual que en la encuesta de la tesis antes mencionada, se empleó en esta investigación el método de expertos en la aplicación de la encuesta tipo cuestionario.

El capítulo estará dedicado por consiguiente, al análisis de los datos recogidos durante de la primera iteración de la encuesta realizada al panel de expertos; obtenidos mediante el muestreo casual y a la fundamentación de la propuesta de sistemas de métricas a implantar como solución a la problemática existente en el proyecto IMAC.

## 2.2 La Encuesta: Análisis y Procesamiento de los Datos.

### 2.2.1 Selección de la Población y la Muestra.

Se definió como el universo o población los administradores de la calidad de los 127 proyectos existentes en la actualidad en la UCI (ver figura 3) los que se encuentran divididos entre Nacionales, de Exportación y propios de la Universidad. De los mismos era necesaria la selección de una muestra representativa para encuestar, la cual se obtuvo a través del muestreo casual o incidental. El tamaño se definió sin emplear ningún algoritmo matemático por lo cual se siguió lo aconsejado por (Álvarez, 1997) aludido en el capítulo anterior, el que hace referencia a un tamaño de muestra superior al 10% de la población, escogiendo el 21% que representa un 26.67 para un aproximado de 27 proyectos.

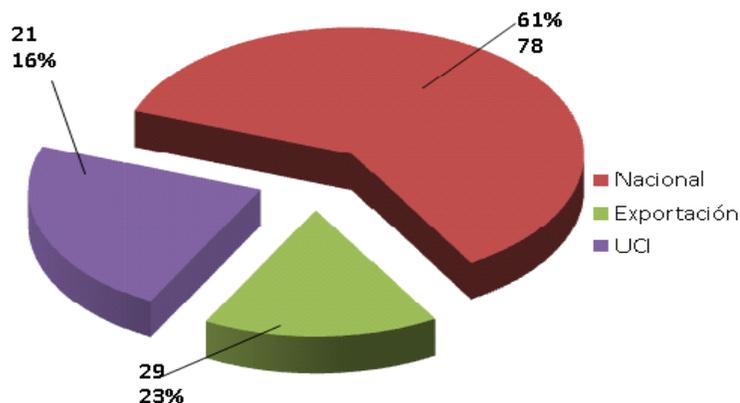


Figura 3: Total de Proyectos en la UCI

Al realizar una encuesta es muy poco probable que todos contesten, ya sea por apatía, por interés o por diversos factores. Por lo que pensando desde un inicio en el índice o coeficiente de respuesta, el cual se define según (EUSKAL ESTADISTIKA ERAKUNDEA(INSTITUTO VASCO DE ESTADISTICA, 2007) como: *“porcentaje de cuestionarios contestados y devueltos de todas las encuestas que se administren”*,

se decidió calcular el tamaño de distribución de la muestra suponiendo que nuestro índice o coeficiente de respuesta oscilaría entre un 60% y un 70%.

Tomando el caso peor para un 60%. La distribución sería:

$$TD = TN / IR$$

TD-Tamaño de Distribución = 45

TM- Tamaño necesario de la muestra (27)

IR-Índice de Respuesta (0.60)

Se distribuyeron después de realizado el cálculo, un total 45 cuestionarios en 45 proyectos para ser contestados por uno de los administradores de calidad.

### 2.2.2 ¿Por qué realizarle la encuesta al administrador de calidad?

El administrador de la calidad es la persona dentro de los proyectos que tiene entre sus responsabilidades asegurar la calidad del proceso de desarrollo de software y el asegurar que la aplicación producida se ajuste a las especificaciones y esté razonablemente libre de errores. Para lograr darle cumplimiento a estas responsabilidades debe estar familiarizado con las siguientes competencias según (Taurel, 2008):

- *Metodologías de desarrollo.*
- *Calidad de Software.*
- *Ingeniería de Software.*
- *Poseer al menos conocimientos básicos sobre el negocio.*

El administrador de la calidad es la persona encargada dentro del equipo del proyecto de diseñar y elaborar detalladamente el plan de medición, en el cual se recogen todas las métricas que posteriormente va a aplicar el proyecto así como la justificación de las mismas. Es esta la razón básica por la que el administrador de la calidad es nuestro centro de atención a la hora de lanzar una investigación en los proyectos sobre las métricas, ya que es el experto en ellas, convirtiéndose en el candidato perfecto a encuestar.

### 2.2.3 La encuesta

Cuando se realiza una encuesta se debe poner especial atención al diseño, ya que es este el punto preponderante del éxito de la obtención de la información necesaria. Esto se debe hacer de manera que se minimicen las posibles fuentes de error, encaminándose las preguntas directamente a cubrir los diferentes aspectos relacionados con los factores de estudio de manera concreta y en un lenguaje claro.

La encuesta utilizada en esta investigación posee un total de 12 preguntas orientadas a cubrir los factores de estudio propuestos, donde 4 de ellas son de contacto, 6 cerradas y 2 abiertas. Estas últimas a pesar de saber que no son eficaces eran necesarias para que el encuestado nos aportase la información solicitada.

En las siguientes tablas (ver Tablas 1, 2, 3) podemos observar los factores de estudio propuestos y temas específicos relacionados, además se especifica lo que se descubrirá con dichos temas y las preguntas correspondientes a la encuesta que se utilizaron para medir el resultado.

**Factor: Experiencia en el uso de métricas de software en los proyectos.**

<b>Tema</b>	<b>Objetivo</b>	<b>Número de pregunta en la encuesta</b>
-------------	-----------------	--

Experiencia del administrador	Identificar el tiempo que lleva trabajando el encuestado como administrador de calidad, teniendo en cuenta tiempo de vida hasta el momento del proyecto.	4,5
Empleo de las métricas en los proyectos.	Identificar si los administradores conocen la importancia de las métricas y las aplican como parte de sus actividades cotidianas.	6,7
Etapas en que se deben aplicar las métricas del software.	Conocer en que etapas o fases del desarrollo del software se considera importante aplicar métricas.	8

Tabla 1: Factor 1

**Factor: Técnicas y Herramientas para gestión de las métricas.**

<b>Tema</b>	<b>Objetivo</b>	<b>Número de pregunta en la encuesta</b>
Herramientas para la gestión de métricas	Conocer qué técnicas y herramientas se usa en los proyectos en la actualidad para gestionar métricas.	9,10

Tabla 2: Factor 2

**Factor: Grado de importancia conferido al conjunto de métricas propuesto.**

Tema	Objetivo	Número de pregunta en la encuesta
Propuesta de Métricas	Conocer el grado de importancia que los administradores de calidad de los proyectos le confieren a un conjunto de métricas seleccionadas.	11

Tabla 3 Factor 3

El último factor como se puede observar en la Tabla 3 se propone con el objetivo de que los expertos aporten su opinión a cerca de las métricas seleccionadas como posibles a aplicar según su experiencia en el proyecto IMAC. Métricas que fueron seleccionadas a partir de los siguientes criterios de calidad de interés del proyecto:

- Evaluar el estado del proyecto.
- Detectar el esfuerzo que se necesita en el desarrollo.
- Hacer un seguimiento de los riesgos.
- Detectar las áreas problemáticas.
- Ajustar el flujo y las tareas del trabajo para evitar retrasos.
- Evaluar la calidad del producto.
- Evaluar la habilidad del sistema de recuperarse ante fallos.

- Valorar de la usabilidad del sistema.
- Valorar la calidad de las pruebas del software.

## 2.2.4 Análisis de los Datos de la Encuesta.

De las 45 encuestas distribuidas fueron respondidas un total de 30, tres más que las necesitadas, para un índice de respuesta de 66.66 %. Procediéndose para su análisis mediante la separación por factores de las preguntas.

### 2.2.4.1 Experiencia en el uso de métricas de software en los proyectos.

El primer factor arrojó como resultado que la experiencia actual de los administradores de calidad UCI se comportaba según se muestra en la Tabla 4:

	2-4 meses	5-7 meses	7-9 meses	Más de 9 meses
Experiencia	16	4	7	3

Tabla 4: Experiencia de los Administradores de Calidad

Para un conocimiento actual o estado de familiarización con el término métricas de un 93.33% contra un estado de aplicación de un 73.33%. (Ver Figuras 4 y 5). Lo que demuestra que a pesar de que existe conocimiento de la importancia y ventajas que brindan, algunos proyectos no le confieren importancia a su uso.



Figura 4: Estado de Familiarización con las Métricas.

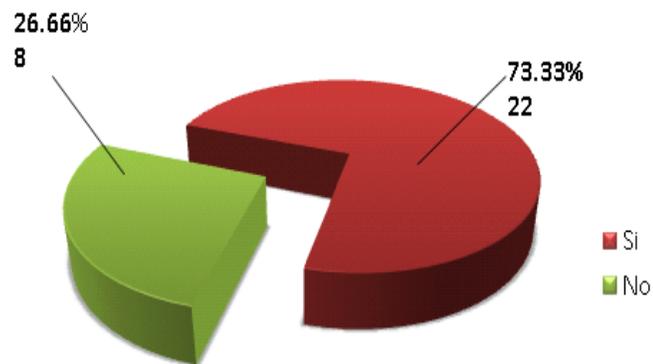


Figura 5: Aplicación de las Métricas en los proyectos.

De los aseguradores que respondieron afirmativamente a la importancia de la aplicación de las métricas 16 respondieron que las métricas se deben aplicar durante todo el ciclo de vida del desarrollo del software. Los restantes coincidieron que en su aplicación cuando empezara la implementación del mismo y en la fase de pruebas.

### 2.2.4.2 Técnicas y Herramientas para gestión de las métricas.

El segundo factor hace referencia en la encuesta a las técnicas y herramientas que utilizan los proyectos para la gestión de métricas. Siendo referenciado por las preguntas 9 y 10 (Ver anexo 1). La pregunta 9 fue respondida negativamente por un 73.33 % de proyectos, para un 13.33% de abstenciones y 13.33 de aceptación. Lo que incidió directamente en el fallo de la pregunta 10, no obteniéndose resultados en la misma. (Ver Figura 6).

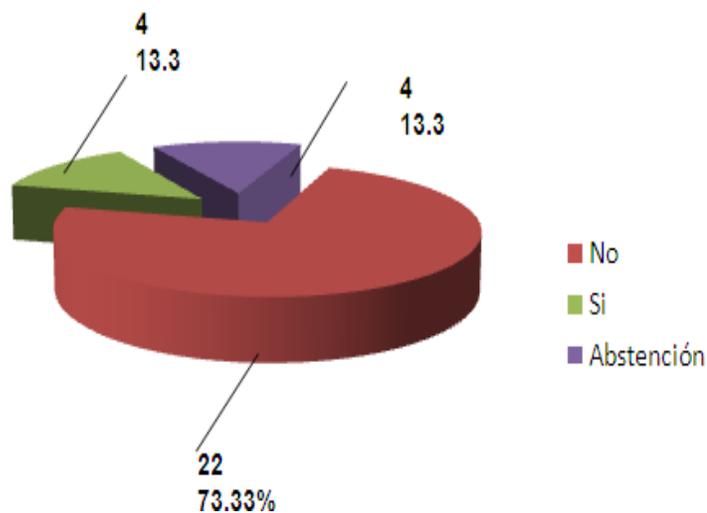


Figura 6: Estado de Empleo de Técnicas y Herramientas para Gestionar las Métricas en los Proyectos.

### 2.2.4.3 Grado de importancia conferido al conjunto de métricas propuesto.

El último factor y el más relevante para esta investigación es el del grado de importancia que le confieren los expertos a un conjunto de métricas seleccionadas teniendo en cuenta los criterios de calidad que se deben medir en el proyecto IMAC y los factores de calidad de McCall. A continuación se muestra la Tabla 2.5 que muestra los resultados obtenidos por cada métrica:

<b>Métricas</b>	<b>Clasificadas de Importantes</b>
Tiempo Real	78.6 %
Tiempo Estimado	39.3 %
Esfuerzo	46.4 %
Identificación de los Riesgos	53.6 %
Probabilidad de ocurrencia de un riesgo identificado.	42.5 %
Efectividad de la mitigación de riesgos.	46.4 %
Magnitud de Retraso.	57.1 %
Adecuación funcional	46.4 %
Cobertura de las pruebas.	53.6 %
Madurez de las pruebas	64.2 %
Restaurabilidad.	46.4 %
Efectividad de la restauración.	42.5 %

Comprensibilidad de entradas y salidas	25.8 %
Complejidad Estimada	53.6 %
Productividad	46.4 %
Eficacia en la Eliminación de Defectos	32.1 %
Defectos Removidos	46.4 %
Índice de Defectos	46.4 %
Porcentaje de defectos por tipo	50 %
Integridad de la Implementación Funcional	57.1 %
Integridad de la descripción del producto	42.8%

Tabla 5: Importancia que le confieren los expertos a Selección de Métricas Inicial y Aplicación de dichas Métricas en los Proyectos

Para la obtención de las métricas más significativas que serán las aplicadas en el proyecto IMAC empleamos la técnica de Pareto. (Ver tabla 6)

Métricas	Frecuencia	%	Acumulado %
Tiempo Real	22	7,77385159	7,77385159
Madurez de las pruebas	18	6,360424028	14,13427562
Magnitud de Retraso	16	5,653710247	19,78798587

Identificación de los Riesgos	15	5,300353357	25,08833922
Cobertura de las pruebas	15	5,300353357	30,38869258
Complejidad Estimada	15	5,300353357	35,68904594
Integridad de la Implementación Funcional	14	4,946996466	40,6360424
Porcentaje de defectos por tipo	14	4,946996466	45,58303887
Esfuerzo	13	4,593639576	50,17667845
Efectividad de la mitigación de riesgos.	13	4,593639576	54,77031802
Adecuación funcional	13	4,593639576	59,3639576
Restaurabilidad.	13	4,593639576	63,95759717
Productividad	13	4,593639576	68,55123675
Defectos Removidos	13	4,593639576	73,14487632
Índice de Defectos	13	4,593639576	77,7385159
Probabilidad de ocurrencia de un riesgo identificado.	12	4,240282686	81,97879859
Efectividad de la restauración.	12	4,240282686	86,21908127
Integridad de la descripción del producto	12	4,240282686	90,45936396
Tiempo Estimado	11	3,886925795	94,34628975
Eficacia en la Eliminación de Defectos	9	3,180212014	97,52650177
Comprensibilidad de entradas y salidas	7	2,473498233	100
<b>Total</b>	<b>283</b>	<b>100</b>	

Tabla 6: Cálculo de Pareto

A continuación para una mayor ilustración de los resultados se muestra la gráfica correspondiente a los cálculos anteriores (ver figura).

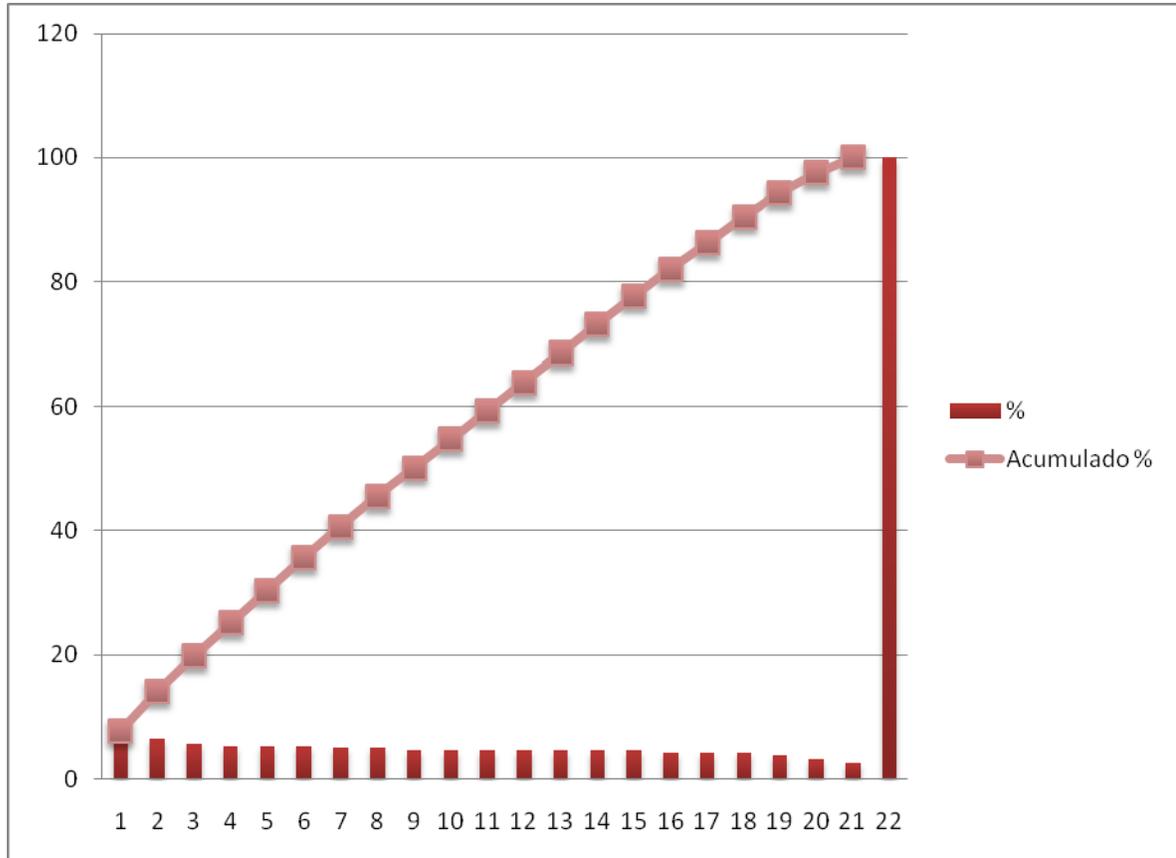


Figura 7: Gráfica de Pareto.

Como se puede observar las métricas más significativas según las respuestas de los expertos son las siguientes:

- 1- Tiempo Real
- 2- Madurez de las pruebas
- 3- Magnitud de Retraso
- 4- Identificación de los Riesgos

- 5- Cobertura de las pruebas
- 6- Complejidad Estimada
- 7- Integridad de la Implementación Funcional
- 8- Porcentaje de defectos por tipo
- 9- Esfuerzo
- 10- Efectividad de la mitigación de riesgos.
- 11- Adecuación funcional
- 12- Restaurabilidad.
- 13- Productividad
- 14- Defectos Removidos
- 15- Índice de Defectos

## ***2.3 Definición de la Propuesta de Métricas***

### **2.3.1 Tiempo Real**

**Descripción:** se define como tiempo real al dedicado por la persona en cada una de las tareas ejecutadas como parte del proyecto. Su registro es necesario para saber como distribuimos el tiempo en la fase de desarrollo. El mismo nos sirve para armar nuestra base datos de tiempos que utilizaremos para mejorar futuras planificaciones.

**Meta:** identificar el tiempo real del trabajo del equipo de desarrollo.

Identificar el tiempo real dedicado por los miembros del equipo en la realización de las tareas asignadas.

**Procedimiento para su análisis:** el tiempo real realizado por el equipo de desarrollo tiene que estar acorde con el planificado por el proyecto.

**Procedimiento para el cálculo:**  $x = \text{jornada laboral} * \text{nro. de miembros del equipo} * \text{días de la semana.}$  o por el tiempo de realización de las tareas asignadas.

**Procedimiento para la recolección:** la recolección del tiempo real se llevará a cabo durante todas las fases de desarrollo del software, será de forma manual, y se almacenará en el repositorio del proyecto con el objetivo de llevar un registro histórico de manera que se mejoren las futuras planificaciones del proyecto.

### 2.3.2 Madurez de las pruebas

**Descripción:** la métrica madurez de las pruebas se orienta en función de qué tan bien se está desarrollando el proceso de pruebas, no solo se preocupa de la completitud de los casos de prueba según los definidos para cumplir los requisitos, sino que también se interesa por cuáles han obtenido resultados satisfactorios, para ello es necesario llevar un control de los casos de prueba que arrojaron resultados satisfactorios y el total de los casos de prueba definidos para el cumplimiento de los requisitos.

**Meta:** evaluar la completitud de los casos de prueba definidos para cumplir los requisitos y verificar cuáles han obtenido resultados satisfactorios.

**Procedimiento para su análisis:** para su análisis el resultado será interpretado por el rango de evaluación de:  $0 \leq X \leq 1$ . Mientras más cercano al 1, mayor madurez de las pruebas.

**Procedimiento para el cálculo:**

$$X = A / B$$

Donde:

A - Número de casos de pruebas que han obtenido un resultado satisfactorio al ser ejecutados o durante su operación.

B - Número de casos de pruebas a ejecutar para cubrir los requisitos.

**Procedimiento para la recolección:** la métrica madurez de las pruebas se llevará a cabo durante la fase de construcción del software en el flujo de trabajo de prueba, será de forma manual, y se almacenarán los resultados obtenidos en el repositorio del proyecto.

### 2.3.3 Magnitud del Retraso

**Descripción:** La magnitud de retraso es una métrica de calidad que indica el grado de retraso existente con respecto a la planificación.

**Meta:** Conocer el tiempo de retraso que existe en comparación con la planificación establecida.

**Procedimiento para su análisis:** para su análisis se verificara el grado de lejanía existente a cero ( $x \geq 0$ ), es decir entre más crezca su valor mas aumenta el retraso en el cronograma.

**Procedimiento para el cálculo:**  $x = \text{tiempo estimado} - \text{tiempo real}$ .

**Procedimiento para la recolección:** la métrica magnitud del retraso se va a recoger durante todo el ciclo de desarrollo del software, será de forma manual, y se almacenarán los resultados obtenidos en el

repositorio del proyecto.

### 2.3.4 Identificación de Riesgos

**Descripción:** la identificación de riesgos es una métrica que se utiliza como medida para guardar los riesgos más comunes en cada una de las etapas del desarrollo del software así como las consecuencias que traen consigo cada uno de ellos (el incremento de los costos, la cancelación del proyecto, la insatisfacción del cliente, entre otras), de manera tal que al cabo de cierto tiempo guardando estos registros históricos al comenzar un nuevo proyecto se tengan identificados los posibles riesgos y prevenirlos, valorando además su repercusión en cuanto al alcance (cuánto se afecta) y la duración (por cuánto tiempo se manifiesta).

**Meta:** identificar los riesgos con el objetivo de planear soluciones que puedan mitigar su efecto.

**Procedimiento para su análisis:** para el análisis de los riesgos se categorizan los mismos en el grado de daño que puedan provocar dentro del proyecto.

Daño 1 – Catastrófico.

Daño 2 – Serio.

Daño 3 – Tolerable.

y establece un rango de impacto que sería:

1-muy bajo

2-bajo

3-medio

4-alto

5-muy alto

**Procedimiento para el cálculo:**

x= Riesgo.

**Procedimiento para la recolección:** la recolección de la identificación de los riesgos se llevará a cabo durante todas las fases de desarrollo del software, será de forma manual, y los resultados de su aplicación se almacenarán en el repositorio del proyecto, con el objetivo de llevar un registro histórico.

### 2.3.5 Cobertura de las pruebas

**Descripción:** la cobertura de las pruebas es una métrica que indica cómo se van cumpliendo los casos de prueba especificados, por lo tanto mientras mayor sea la cobertura, mayor número de casos de prueba se estarán cumpliendo, de esta manera se llevará un control del cumplimiento de los casos de prueba requeridos para cubrir los requisitos lo que por supuesto da una medida de cuán correctamente se está desarrollando el proceso de prueba. Permitiendo hacer un análisis de cuántos casos de pruebas requeridos han sido ejecutados detectados durante las pruebas.

**Meta:** llevar un control del cumplimiento de los casos de prueba requeridos para cubrir los requisitos.

**Procedimiento para su análisis:** la interpretación esta dada por el rango de evaluación de:

$0 \leq X \leq 1$ . Mientras más cercano al 1, mejor cobertura.

**Procedimiento para el cálculo:**

$X = A / B$  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.

**Procedimiento para la recolección:** la métrica cobertura de las pruebas se llevará a cabo durante la fase de construcción del software en el flujo de trabajo de prueba, será de forma manual, y se almacenarán los resultados obtenidos en el repositorio del proyecto.

### 2.3.6 Complejidad Estimada

**Descripción:** la complejidad estimada es relativa a la propiedad externa que involucra el proceso de programación (que incluye al programador) en el marco de su complejidad psicológica.

**Meta:** estimar tempranamente la complejidad estructural del software y determinar su factibilidad, además de identificar tempranamente los posibles riesgos de error que puedan presentarse y el grado de responsabilidad que debe adquirir el proceso de pruebas.

**Procedimiento para su análisis:** para su análisis se establecen los siguientes rangos, en donde si la complejidad estimada se encuentra entre:

1-10	Programa Simple, sin mucho riesgo
------	-----------------------------------

11-20	Más complejo, riesgo moderado
21-50	Complejo, Programa de alto riesgo
50	Programa no testeable, Muy alto riesgo

Tabla 7: Rango de Análisis de la Complejidad Estimada de un Sistema.

**Procedimiento para el cálculo:** el cálculo de la complejidad se va realizar por el método Albrech conocido como puntos de función; el cual esta basado en el empleo de factores normalizados para juzgar la importancia relativa de varios requisitos funcionales. Parte de cinco funciones básicas que suelen aparecer en muchos sistemas:

- Entradas: pantallas o formatos empleados para introducir datos a un programa.
- Salidas: pantallas o informes empleados para utilizarlos con otros programas o para lectura directa.
- Consultas: mecanismos para pedir ayuda o dar órdenes de ejecución.
- Ficheros de datos: conjuntos lógicos de información empleados por una aplicación (ya sean tablas en memoria como ficheros de disco) junto con los procedimientos de acceso a los mismos.
- Interfaces: Ficheros compartidos con otras aplicaciones.

La idea básica del método consiste en definir unas estimaciones de complejidad para cada una de estas funciones (en forma de pesos relativos) y estimar, dadas las especificaciones del sistema, cuántos elementos de cada tipo van a ser necesarios.

**Procedimiento para la recolección:** la recolección de la complejidad dentro del proyecto IMAC va a ser manual, siendo almacenados sus resultados en el repositorio del mismo, con el objetivo de calibrar los

niveles de medición obtenidos. La fase que se propone para su aplicación es la de elaboración en el flujo de trabajo de implementación.

### 2.3.7 Integridad de la Implementación Funcional

**Descripción:** es una medida de cuán completa ha sido la implementación según la especificación de requisitos, se detectan el número de funciones perdidas, aquellas que fueron descritas en la especificación de requisitos y no fueron implementadas. Con esta métrica se evalúa la completitud de la implementación, si se tienen muchas funcionalidades perdidas, no si se desarrolló una buena implementación, lo cual implicará la toma de acciones correctivas para controlar este proceso de manera tal que no se vea afectada la calidad del producto final.

**Meta:** evaluar la completitud de la implementación.

**Procedimiento para su análisis:** su interpretación esta dada por el rango de evaluación de:

$0 \leq X \leq 1$  .Donde entre más cercano a 1 este el resultado más completo habrá sido la integridad de implementación.

**Procedimiento para el cálculo:**

$$X = 1 - A/B$$

Donde:

A - Número de funciones perdidas detectadas en la evaluación.

B - Número de funciones descritas en la especificación de requisitos.

**Procedimiento para la recolección:** la métrica integridad de la implementación funcional se aplicará durante la fase de elaboración del software en el flujo de trabajo de implementación, será de forma manual, y se almacenarán los resultados obtenidos en el repositorio del proyecto.

### 2.3.8 Porcentaje de Defectos por tipo

**Descripción:** el porcentaje de defectos por tipo es una métrica de calidad orientada al producto. Permite el establecimiento de un registro histórico que ayuda a la toma de acciones correctivas por parte del equipo de desarrollo. Esta métrica puede ser llevada de forma individual para cada desarrollador o por equipo de trabajo. Es de gran importancia para el establecimiento de futuros proyectos.

**Meta:** identificar los tipos de defectos más comunes que puedan presentarse en cualquiera de las etapas del proceso de desarrollo del software.

**Procedimiento para su análisis:** la interpretación esta dada por el rango de evaluación de:

A-Sintácticos  $0 \leq A \leq 100$

B-Ejecución  $0 \leq B \leq 100$

C-Seguridad  $0 \leq C \leq 100$

D-Interfaz  $0 \leq D \leq 100$

E-Descripción  $0 \leq E \leq 100$

F-Precondición  $0 \leq F \leq 100$

G-Flujo Básico  $0 \leq G \leq 100$

H-Semánticos  $0 \leq H \leq 100$

I-Lógicos  $0 \leq H \leq 100$

Donde que el valor resultante este más cercano a cero es mejor indica la baja ocurrencia del defecto.

**Procedimiento para el cálculo:** tipo de defecto\* 100 / total de defectos encontrados en el software.

**Procedimiento para la recolección:** la recolección del porcentaje de defectos por tipo se llevará a cabo durante todas las fases de desarrollo del software, será de forma manual, y se almacenará en el repositorio del proyecto, con el objetivo de darle seguimiento y tratar de reducirlos lo más posible.

### 2.3.9 Esfuerzo de desarrollo del software (personas-mes)

**Descripción:** esfuerzo de desarrollo del software es una métrica que permite conocer la cantidad personas (x) que se necesitan por mes (x) en un proyecto para desarrollar un producto de software. Su valor es uno de los más difíciles de obtener y propensos a error de la ingeniería del software, ya que es difícil una estimación exacta debido al gran número de factores conocidos o esperados que determinan la complejidad y desconocidos o no esperados que van a producirse en cualquier momento, determinando la incertidumbre.

**Meta:** estimar el esfuerzo humano requerido y tiempo aproximado para desarrollar el producto de software.

**Procedimiento para el cálculo:** para el cálculo del esfuerzo de desarrollo del software se va a emplear la técnica conocida como COCOMO 2 (Modelo Constructivo de Costo). Él cual consiste básicamente en la aplicación de ecuaciones matemáticas sobre los puntos de función sin ajustar o la cantidad de líneas de código (SLOC, SourceLines Of Code) estimados para un proyecto. Estas ecuaciones se encuentran ponderadas por ciertos factores de costo (cost drivers) que influyen en el esfuerzo requerido para el desarrollo del software. Se escoge por que permite estimar de forma más detallada y exacta, además de que los resultados pueden refinarse con su uso a la vez que progresa el desarrollo del proyecto.

**Procedimiento para su análisis:** su análisis es literal.

**Procedimiento para la recolección:** la recolección del esfuerzo de desarrollo se llevará a cabo durante la fase de inicio de desarrollo del software, se refinara durante las fases siguientes, será de forma manual, y

se almacenará en el repositorio del proyecto, con el objetivo de darle seguimiento y tratar de reducirlos lo más posible.

### 2.3.10 Efectividad en la Mitigación de Riesgos

**Descripción:** la efectividad en la mitigación de riesgos es una métrica que determina la relación existente entre los riesgos mitigados y los identificados.

**Meta:** identificar cuán efectivos han sido los planes de mitigación de riesgo.

**Procedimiento para su análisis:** la escala de valoración de efectividad de la mitigación se ajustará al siguiente rango:

Control	Efectividad
Ninguno	1
Bajo	2
Medio	3
Alto	4
Destacado	5

Tabla 8: Rango de Control de Riesgos.

**Procedimiento para el cálculo:** se establece de forma lineal.

**Procedimiento para la recolección:** la recolección de efectividad de la mitigación de riesgos se llevará a cabo durante todas las fases de desarrollo del software, será de forma manual, y se almacenará en el repositorio del proyecto con el objetivo de llevar un registro histórico.

### 2.3.11 Adecuación funcional

**Descripción:** la métrica adecuación funcional permite hacer un análisis de cuán adecuada es la función evaluada.

**Meta:** analizar cuan adecuada es la función evaluada.

**Procedimiento para su análisis:** la interpretación esta dada por el rango de evaluación de:  
 $0 \leq X \leq 1$ . A mayor cercanía al 1 resultará más adecuada

**Procedimiento para el cálculo:**

$$X = 1 - A/B$$

Donde:

A - Número de funciones en las cuales se detectaron problemas en la evaluación.

B - Número de funciones evaluadas.

**Procedimiento para la recolección:** la métrica adecuación funcional se aplicará durante la fase de elaboración del software en el flujo de trabajo de implementación, su recolección será de forma manual, y se almacenarán los resultados obtenidos en el repositorio del proyecto.

### 2.3.12 Restaurabilidad

**Descripción:** la métrica restaurabilidad permite hacer un análisis de cuán capaz es el producto de auto restaurarse luego de un evento anormal o una solicitud.

**Meta:** capacidad de autorestauración del producto luego de algún fallo.

**Procedimiento para su análisis:** la interpretación esta dada por el rango de evaluación de:

$0 \leq X \leq 1$ . A mayor cercanía al 1 mejor y el producto es más capaz de restaurarse en casos definidos.

**Procedimiento para el cálculo:**

$$X = A / B$$

Donde:

A - Número de casos de restauración exitosos.

B - Número de casos de restauración probados por los requisitos.

El valor de A proviene de las revisiones y el de B del documento de los requerimientos o el del diseño.

**Procedimiento para la recolección:** la métrica restaurabilidad se aplicará durante las fases de construcción y transición del desarrollo del software, su recolección será de forma manual, y se almacenarán los resultados obtenidos en el repositorio del proyecto.

### 2.3.13 Productividad

**Descripción:** la productividad es una métrica que calcula la generación de producto por unidad de tiempo.

**Meta:** evaluar el rendimiento del trabajo del proyecto.

**Procedimiento para su análisis:** la interpretación esta dada por el rango de evaluación de: Cuando

$X \leq 60\%$  la productividad es baja.

Cuando  $60\% < X \leq 70\%$  la productividad es media.

Cuando  $70\% < X \leq 100\%$  la productividad es alta.

Cuando  $X > 100$  la productividad es muy alta.

**Procedimiento para el cálculo:**

$$TE / TR * TA / TC = X / 100$$

Donde:

TR – Tiempo Real de Cumplimiento de las Tareas

TE- Tiempo Estimado de Cumplimiento de las Tareas

TA- Tareas Asignadas

TC- Tareas Completadas

X- Productividad

**Procedimiento para la recolección:** la recolección de la productividad se llevará a cabo durante todas las fases de desarrollo del software, durante el cierre del mes, será de forma manual, y los resultados de

su aplicación se almacenarán en el repositorio del proyecto. Esta tarea se realizará con el objetivo de llevar un registro de la productividad mensual, que permita la obtención de diferentes medidas, con lo que se podrá evaluar los sus resultados históricos de forma tal que se incremente su valor en los próximos meses.

### 2.3.14 Defectos Removidos

**Descripción:** defectos removidos es una métrica de calidad orientada al producto, que tiene como finalidad la contabilización del total de defectos al que se le ha dado solución.

**Meta:** identificar el total de defectos que han tenido solución de los identificados.

**Procedimiento para su análisis:** la interpretación esta dada por el rango de evaluación de:

- A-Sintácticos  $0 \leq A \leq$  total de los defectos encontrados.
- B-Ejecución  $0 \leq B \leq$  total de los defectos encontrados.
- C-Seguridad  $0 \leq C \leq$  total de los defectos encontrados.
- D-Interfaz  $0 \leq D \leq$  total de los defectos encontrados.
- E-Descripción  $0 \leq E \leq$  total de los defectos encontrados.
- F-Precondición  $0 \leq F \leq$  total de los defectos encontrados.
- G-Flujo Básico  $0 \leq G \leq$  total de los defectos encontrados.
- H-Semánticos  $0 \leq H \leq$  total de los defectos encontrados.
- I-Lógicos  $0 \leq J \leq$  total de los defectos encontrados.

Donde si el valor resultante es más cercano al total de los defectos encontrados significa que se está llevando a cabo una buena remoción de defectos por parte del equipo de desarrollo.

**Procedimiento para el cálculo:**  $x \rightarrow$  número de defectos removidos por tipo.

**Procedimiento para la recolección:** la recolección de los defectos removidos se llevará a cabo durante todas las fases de desarrollo del software, será de forma manual, y se almacenará en el repositorio del proyecto con el objetivo de llevar un registro histórico de los defectos que han ido eliminándose.

### 2.3.15 Índice de Defectos

**Descripción:** el índice de defectos es una métrica de calidad orientada al producto. La aplicación de esta métrica permite contabilizar los errores introducidos en el desarrollo del software, proporcionando experiencia para el equipo de desarrollo en futuros proyectos.

**Meta:** identificar, describir y contabilizar el número total de defectos que afectan al producto de software. Se aplica con el objetivo de reducir lo más posible los defectos con para minimizarlos y reducirlos a un valor lo más cercano posible a cero.

**Procedimiento para su análisis:** los defectos se podrán clasificar por el tipo:

A-Sintácticos: los defectos de sintaxis son aquellos que aparecen mientras escribe el código o documentación relativa al producto de software.

B-Ejecución: error de ejecución se produce cuando el ordenador no puede ejecutar alguna instrucción de forma correcta.

C-Seguridad: los defectos de seguridad son los baches o los llamados huecos que nos indican que el sistema informático que se está desarrollando no está libre de todo peligro, daño o riesgo.

D-Interfaz: los defectos de interfaz son los fallos o errores encontrados en la Conexión, física o lógica,

entre un computador y el usuario, un dispositivo periférico o un enlace de comunicaciones.

E-Descripción: un defecto o error de descripción es aquel que puede ocasionar una mala interpretación de la lógica del negocio.

F-Precondición: los defectos de precondición se le llama al no cumplimiento de las condiciones que ha de

G-Flujo Básico: los defectos de flujo básico se le llaman a los errores introducidos en el flujo normal de eventos.

H-Semánticos: los defectos semánticos son aquellos que se presentan en el conjunto de reglas que proporcionan el significado de una sentencia o instrucción de cualquier lenguaje de programación.

I-Lógicos: defectos lógicos son aquellos que aparecen cuando la aplicación está en funcionamiento. Son a menudo resultados no deseados o inesperados en respuesta a acciones del usuario. Por ejemplo, una clave mal escrita u otra influencia externa podrían hacer que la aplicación dejase de funcionar aún siendo correctos los parámetros, o que simplemente no funcionase.

y por el grado de severidad en:

Severidad 1 – Critico. El defecto hace que el software inutilizable.

Severidad 2 – Defecto Importante. Puede trabajar pero no es deseable.

Severidad 3 – Defecto medio. Puede continuar el trabajo.

Severidad 4 – Defecto leve. Defecto superficial.

**Procedimiento para el cálculo:** la interpretación esta dada por el rango de evaluación de:

- A- total de defectos de sintácticos encontrados en el software y grado de severidad que poseen.
- B- total de defectos de ejecución encontrados en el software y grado de severidad que poseen.
- C- total de defectos de seguridad encontrados en el software y grado de severidad que poseen.
- D- total de defectos de interfaz encontrados en el software y grado de severidad que poseen
- E- total de defectos de descripción encontrados en el software y grado de severidad que poseen
- F- total de defectos de preconditionación encontrados en el software y grado de severidad que poseen.
- G- total de defectos de flujo básico encontrados en el software y grado de severidad que poseen.
- H- total de defectos semánticos encontrados en el software y grado de severidad que poseen.
- I- total de defectos lógicos encontrados en el software y grado de severidad que poseen.

**Procedimiento para la recolección:** la recolección de los defectos se llevará a cabo durante todas las fases de desarrollo del software, involucrará a todos los artefactos generados en el proceso, será de forma manual y se almacenará en el repositorio del proyecto, con el objetivo de darle seguimiento y tratar de reducirlos lo más posible.

## ***2.4 Conclusiones***

Este capítulo es una síntesis del estudio realizado para la obtención de la propuesta de métricas a aplicar en el proyecto IMAC. Durante el desarrollo del mismo se expusieron las características de la cada una de las métricas seleccionadas, objetivos que se persiguen con su aplicación, como se va a proceder para su análisis, el proceder para su recolección, y posterior evaluación. Lo que propició que se obtuviese una guía detallada para la aplicación de las mismas dentro del proyecto.

El desarrollo de la propuesta realizada fue guiado por lo establecido en CMMI en el Área de Procesos de Medición y Análisis, su definición se basó en los criterios de calidad formulados en el proyecto IMAC y su

filtrado se obtuvo a través del criterio de los expertos recogido a través de la encuesta tipo cuestionario llevada a cabo como parte del proceso investigativo del presente trabajo.

## ***Capítulo 3: Estudio del Caso Real (IMAC):***

### ***3.1 Introducción***

El presente capítulo está dedicado a la evaluación de la propuesta de métricas en el proyecto IMAC. La misma será evaluada en el período comprendido entre los meses de febrero a mayo.

El proyecto IMAC se encarga del desarrollo de tres módulos para automatizar procesos del MAC (Planificación, Acciones de Control e Integración). Actualmente está transitando por la fase de Transición. Su conceptualización se basa en la metodología RUP (Proceso Unificado de Desarrollo) y utiliza IDEF (Integrated Definition Methods), que es un subconjunto de la metodología del SADT (Structured Analysis and Design Technique) para el Análisis Estructurado y Técnicas de Diseño que se caracteriza por ser un método formalizado de descripción de procesos.

El módulo de Planificación se dedica a automatizar el proceso de gestión de la información en la dirección de planificación, análisis y control del MAC. Para la implementación del mismo se utilizó como lenguaje de programación PHP y como gestor de base de datos PostgreSQL.

El módulo de Acciones de Control ha sido modelado y desarrollado mediante el EMS (Enterprise Manager System), Alfresco en su versión 3.0, solución OpenSource para la gestión empresarial de contenido documental. En esencia recrea los procesos de la Dirección de Control Gubernamental (DCG), Dirección de Auditorías Gubernamentales (DAC) y Dirección de Auditoría y Comprobaciones Especiales (DACE). Dichos procesos están relacionados con las actividades que se realizan en estas direcciones a la hora de llevar a cabo acciones de control hacia cualquier entidad o ministerio. Así como de gestionar la correcta búsqueda y almacenamiento de esta información para futuras consultas o reportes de cumplimiento.

El módulo de Integración es el que se encarga de sincronizar la información proveniente de los dos

módulos antes mencionados de forma que se recree una única interfaz de comunicación. Su implementación es desarrollada con php y se utiliza el api de Alfresco para la integración. Hasta el momento sólo ha implementado una de sus funcionalidades.

Aparte del equipo de los módulos antes señalados el proyecto cuenta con los equipos de línea horizontal estos serían los de Despliegue, Análisis, Calidad, y Configuración los que a la hora de analizar sus actividades no serán analizados de manera individual sino en los aspectos generales del proyecto.

### 3.2 Proceso de Evaluación.

#### 3.2.1 Resultados de la Aplicación de la Métrica Complejidad Estimada.

La complejidad estimada es una de las métricas de software más ampliamente aceptada, ya que ha sido concebida para ser independiente del lenguaje de desarrollo. En su aplicación en el proyecto IMAC como se puede observar (Ver tabla 8) se obtuvieron para los tres módulos los resultados siguientes:

	Módulo de Planificación	Módulo Acciones de Control	Módulo de Integración
Complejidad Estimada	21	17	27

Tabla 9: Cálculo de la Complejidad Estimada

Indicador de que el sistema que se está desarrollando por parte del proyecto IMAC es un programa de alto riesgo, lo que implica un grado elevado de responsabilidad en su desarrollo. Por lo que el proceso de

pruebas a realizar como parte de las actividades para asegurar la calidad del software desarrollo debe ser un trabajo comprometido y riguroso, que cubra todas las funcionalidades del sistema.

### 3.2.2 Resultados de la Aplicación de la Métrica Esfuerzo de desarrollo de (personas-mes).

El esfuerzo de desarrollo (personas-mes) es una métrica que permite el apoyo a la planificación, lo que es vital para la gerencia en los proyectos. Aunque los resultados de su aplicación generalmente no son del todo correctos debido a que no se toman en cuenta un conjunto de factores que introducen incertidumbre, nos dan un aproximado del esfuerzo requerido. En la aplicación de la misma dentro del IMAC se obtuvieron en los tres módulos los resultados mostrados en la siguiente tabla:

	Módulo de Planificación	Módulo Acciones de Control	Módulo de Integración
Esfuerzo	7 personas / 6 meses	5 personas / 10 meses	2 personas / 1 mes

Tabla 10: Cálculo del Esfuerzo de desarrollo de (personas-mes).

De estos resultados se ha determinado que:

- Se estima un tiempo de duración aproximado para el desarrollo del proyecto IMAC de 11 meses.

### 3.2.3 Resultados de la Aplicación de las Métricas Índice de Defectos, Porcentaje de Defectos por tipo y Defectos Removidos.

Un defecto es el resultado de un fallo o deficiencia durante el proceso de creación de programas para

ordenadores. El mismo puede presentarse en cualquiera de las etapas del ciclo de vida del software y en cualquier artefacto que se genere como parte de su proceso creativo.

A continuación se muestra una tabla (Ver tabla 10) con el comportamiento de los defectos más comunes encontrados en el proyecto IMAC y el seguimiento que se les ha dado a los mismos.

Clasificación de los Defectos	Índice Defectos	Porcentaje de Defectos por tipo	Removidos Defectos
Semánticos	5	3.7%	5
Flujo Básico	5	3.7%	5
Precondición	7	5.1%	7
Descripción	10	7.6%	10
Seguridad	6	4.4%	1
Interfaz	4	2.9%	3
Ejecución	5	3.7%	4
Sintácticos	93	68%	80
Lógicos	8	6%	8

Tabla 11: Defectos Encontrados en el IMAC

De estos resultados anteriores se concluye que:

- Existe en el proyecto un índice de defectos de 143 para un total de 123 defectos removidos.
- Los defectos más comunes encontrados son los sintácticos.
- Los que más dañinos en estos momentos son los de seguridad, de los cuales sólo ha sido removido uno. Lo que indica que el sistema es poco robusto y se encuentra vulnerable a ataques

### 3.2.4 Resultados de la Aplicación de las Identificación de los Riesgos, Efectividad de la mitigación de riesgos.

Un riesgo es una contingencia o daño que en cualquier momento puede materializarse o no hacerlo nunca. Gráficamente se tiene:

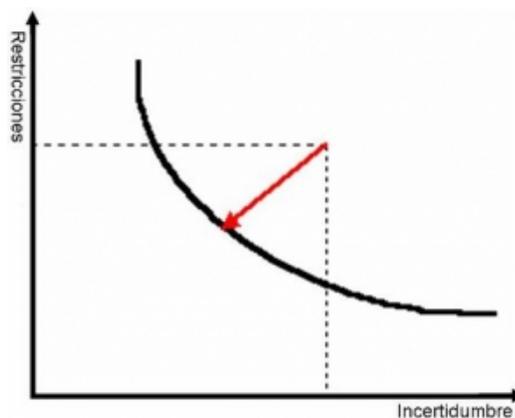


Figura 8: Curva Riesgo Aceptable

La curva del gráfico muestra el “Nivel Aceptable de Riesgo”. El riesgo se puede reducir hasta el Nivel Aceptable por medio de la disminución de las restricciones (técnicas, económicas, disponibilidad, etc.) y de la incertidumbre (probabilidad que ocurra el riesgo). En la práctica, las restricciones son muy difíciles de eliminar o flexibilizar, de manera que el foco se coloca en la disminución de la incertidumbre. Del gráfico se desprende que eliminar el riesgo totalmente no es posible o no es económicamente posible.

Ahora bien, la anticipación del riesgo permite hacer acciones de mitigación, que son mucho más baratas de hacer que las acciones correctiva cuando se presenta el riesgo.

A continuación se muestra una tabla (Ver Tabla 8) con el registro de los riesgos identificados por el proyecto IMAC y el estado de incidencia de los mismos durante los meses de estudio (Febrero (IF), Marzo (IM), Abril (IA), Mayo (IM)).

Nro.	Riesgo	Descripción	IF	IM	IA	IM	Daño	Impacto
R1	Planificación demasiado optimista	Planificaciones demasiado optimistas a partir de la inexperiencia y falta de datos históricos.	15	10	19	17	2	4
R2	Incumplimiento del cronograma del IMAC.	Retraso de las actividades previstas en el cronograma en el nivel central incide en los cronogramas desconcentrados	15	10	19	17	2	4
R3	Falta de compromiso y responsabilidad del equipo de trabajo del IMAC.	No apoyan activamente el proceso algunos miembros del equipo de trabajo	28	31	30	31	1	5
R4	Desmotivación del equipo de IMAC.	Desmotivación por razones internas y externas.	28	31	30	31	2	4
R5	Problemas con la electricidad.	Faltas de fluido eléctrico.	0	0	2	2	2	4
R6	Falta de coordinación de las tareas conjuntas entre clientes y el equipo de desarrollo.	Problemas de coordinación que puedan afectar alguna tarea que implique ambas partes.	2	0	7	2	2	4

R7	La falta de la especialización en las áreas necesarias para desarrollar el SIGAC.	No contar con personal especializado para el desempeño de determinados roles.	0	0	0	0	0	0
R8	Escasez de máquinas para trabajar.	Más de tres desarrolladores en un mismo puesto de trabajo.	0	0	0	0	0	0
R9	Los miembros del equipo no trabajan unidos.	Falta de entendimiento entre los miembros del equipo a la hora de tomar decisiones, etc.	28	31	30	31	1	5
R10	No poder lograr la integración de los módulos	Que no se pueda lograr la funcionalidad de los métodos	28	31	30	31	1	5
R11	Cambio de línea de trabajo.	Cambio de la arquitectura.	0	0	0	0	0	0
R12	Cambio del equipo de trabajo.	Reestructuración de los miembros del equipo.	0	0	0	0	0	0
R13	No poder obtener las actualizaciones del SVN del ERP-CUBA.	No conexión con el Repositorio del proyecto ERP-CUBA	28	31	30	31	2	4

R14	No poder integrar el reportador.	No se conoce como realizar la integración con reportador en la versión ERP utilizada en el módulo de planificación	0	0	30	31	2	4
-----	----------------------------------	--	---	---	----	----	---	---

Tabla 12: Listado de Riesgos.

Como se puede observar en la tabla los riesgos R3,R9,R10 son de daño 1 y los R1, R2, R4, R5,R6,R13,R14 son de daño 2 y se encuentran en grado de impacto alto, los riesgos R1,R2,R4,R5,R6,R13,R14 y muy alto R3, R9 R10, no ocurrieron en el cuatrimestre analizado.

Para darle solución a los riesgos identificados el proyecto IMAC propuso un conjunto de acciones de mitigación. Las mismas son mostradas en la tabla 9 que a continuación se observa en conjunto con el grado de efectividad logrado.

Nro.	Solución
R1	Se implementó una planilla de extensión .xls para que cada miembro del proyecto planificara sus actividades de la semana, en el mismo en conjunto con el tiempo estimado que se esperaba demorar. Esta actividad es supervisada de forma constante. Siendo responsable de la calidad de dicha actividad el planificador y el asesor de calidad del IMAC.
R2	Cada miércoles se realizan reuniones con los líderes de los módulos para rendir cuentas de los artefactos que se debieron generar en la semana pasada.

<i>R3</i>	Se evalúa el desempeño de los estudiantes vinculados al proyecto y con este se obtiene la nota de la asignatura de Práctica Profesional.
<i>R4</i>	Boletín semanal del proyecto.
<i>R5</i>	Caso 1: Falta de fluido eléctrico en la escuela. Existencia de una planta energética que abastece al laboratorio. Configurar en el setup del servidor el encendido automático. Caso2: Falta de fluido eléctrico en las pcs. Localizar de inmediato al técnico e informar la ausencia de electricidad en las pcs.
<i>R6</i>	Se media la comunicación entre el proyecto y las entidades pertenecientes al ministerio mediante un asesor del Ministerio de Auditoría y Control que se encarga de tramitar la comunicación.
<i>R7</i>	Capacitación del equipo de trabajo.
<i>R8</i>	Establecimiento de un horario laboral en dependencia de las necesidades inmediatas del proyecto. Establecimiento de contraseña y usuario únicos para el proyecto.
<i>R10</i>	Búsqueda de Asesoramiento.
<i>R11</i>	Capacitación y Estudio Individual de las herramientas propuestas.
<i>R12</i>	Curso de capacitación de los miembros.

<i>R13</i>	No se ha definido.
<i>R14</i>	Búsqueda de Asesoramiento.

Tabla 13: Soluciones Propuestas para la Mitigación de los Riesgos.

La efectividad de las soluciones propuestas se muestra a continuación (Ver tabla 14):

Riesgos	Efectividad de la mitigación de Riesgos
<i>R1</i>	1
<i>R2</i>	1
<i>R3</i>	2
<i>R4</i>	1
<i>R5</i>	2

<i>R6</i>	3
<i>R7</i>	4
<i>R8</i>	-
<i>R9</i>	2
<i>R10</i>	2
<i>R11</i>	-
<i>R12</i>	-
<i>R13</i>	3
<i>R14</i>	1

Tabla 14: Efectividad de las Soluciones Propuestas para los Riesgos.

Como se puede observar en la tabla la solución más efectiva fue la capacitación del equipo de desarrollo para solucionar el problema de la falta de la especialización en las áreas necesarias para desarrollar el SIGAC.

### 3.2.5 Resultados de Aplicación de la Métrica de Productividad y Tiempo Real.

La productividad es la relación entre los resultados y el tiempo utilizado para obtenerlos, cuanto menor sea el tiempo que lleve obtener el resultado deseado, más productivo es el proyecto. Se calculó en el IMAC a través del análisis de las tareas asignadas y completadas contra tiempo estimado y real, según los cronogramas existentes de los meses de febrero a mayo. A continuación se muestran las tablas (11, 12, 13,14) correspondientes dichos meses.

<b>Mes de Febrero</b>					
<b>Módulo de Planificación</b>					
TA	T C	TE	TR	R	P
19	19	25 días	28 días	3 días	89%
<b>Módulo de Acciones de Control</b>					
TA	T C	TE	TR	R	P
7	7	20 días	24 días	4 días	83%
<b>Módulo de Integración</b>					
TA	T C	TE	TR	R	P
4	4	10 días	18 días	8 días	55%

Tabla 15: Comportamiento de la Productividad en el Mes de Febrero.

<b>Mes de Marzo</b>					
<b>Módulo de Planificación</b>					
TA	T C	TE	TR	R	P
20	20	26 días	30 días	4 días	87%
<b>Módulo de Acciones de Control</b>					

TA	T C	TE	TR	R	P
15	15	20 días	22 días	2 días	91%
<b>Módulo de Integración</b>					
TA	T C	TE	TR	R	P
12	12	16 días	20 días	4 días	80%

Tabla 16: Comportamiento de la Productividad en el Mes de Marzo.

<b>Mes de Abril</b>					
<b>Módulo de Planificación</b>					
TA	T C	TE	TR	R	P
7	7	9	25 días	8 días	36%
<b>Módulo de Acciones de Control</b>					
TA	T C	TE	TR	R	P
13	13	21 días	30 días	9 días	70 %
<b>Módulo de Integración</b>					
TA	T C	TE	TR	R	P
7	7	12 días	20 días	2 días	86%

Tabla 17: Comportamiento de la Productividad en el Mes de Abril

<b>Mes de Mayo</b>					
<b>Módulo de Planificación</b>					
TA	T C	TE	TR	R	P
12	12	16 días	25 días	9 días	64%
<b>Módulo de Acciones de Control</b>					

TA	T C	TE	TR	R	P
14	14	20 días	30 días	10 días	67%
Módulo de Integración					
TA	T C	TE	TR	R	P
6	6	9 días	20 días	11 días	45%

Tabla 18: Comportamiento de la Productividad en el Mes de Mayo.

Donde:

TA - Cantidad de Tareas Asignadas

TC - Cantidad de Tareas Completadas

TE - Tiempo Estimado en Días

TR - Tiempo Real de Cumplimiento

R -Tiempo de Retraso

P – Productividad

Resumen general del proyecto(Febrero)				
Total Asignadas	Total Completadas	Tiempo Total E	Tiempo Total R	Productividad
30	30	25 días	28 días	89,2%

Tabla 19: Resumen General de las Tareas del Proyecto en el Mes de Febrero.

Resumen general del proyecto(Marzo)				
Total Asignadas	Total Completadas	Tiempo Total E	Tiempo Total R	Productividad
47	47	26 días	30 días	86,7%

Tabla 20: Resumen General de las Tareas del Proyecto en el Mes de Marzo.

Resumen general del proyecto(Abril)				
Total Asignadas	Total Completadas	Tiempo Total E	Tiempo Total R	Productividad
85	85	21 días	30 días	42%

Tabla 21: Resumen General de las Tareas del Proyecto en el Mes de Abril.

Resumen general del proyecto(Mayo)				
Total Asignadas	Total Completadas	Tiempo Total E	Tiempo Total R	Productividad
68	68	20 días	30 días	61.4%

Tabla 22: Resumen General de las Tareas del Proyecto en el Mes de Mayo.

Para ver más claramente la productividad alcanzada durante el cuatrimestre que se estudia, se muestra a continuación la siguiente imagen que ilustra gráficamente comportamiento general del proyecto.

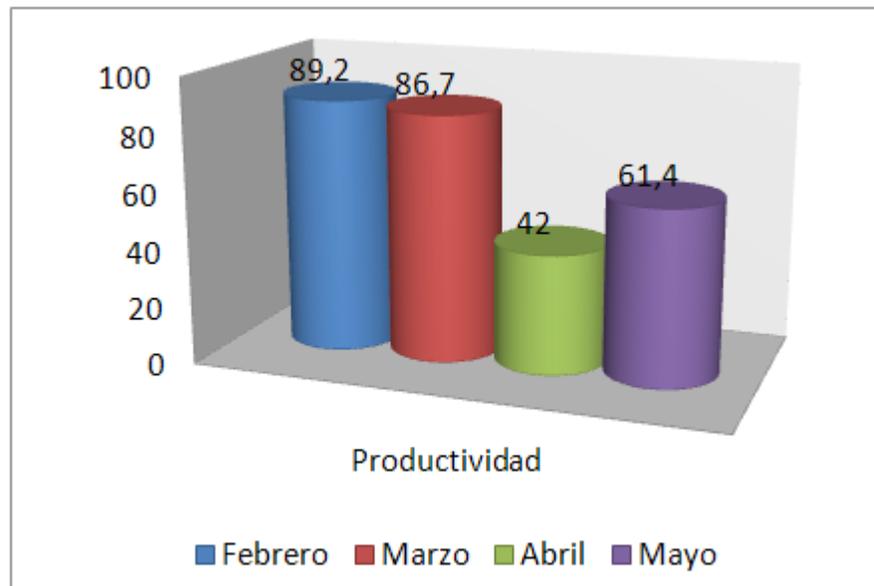


Figura 9: Gráfica de Productividad del cuatrimestre (febrero-mayo).

De estos resultados se ha determinado que:

- La productividad ha bajado con respecto a los meses anteriores.

Al hacer un análisis de las causas que han influido en la disminución de la productividad se hace referencia a la incidencia de los riesgos acontecidos durante los meses de estudio, que han incidido directamente en su valor, estos son:

- Falta de compromiso y responsabilidad del equipo de trabajo del IMAC.
- Desmotivación del equipo de IMAC.

Lo que ha provocado el incumplimiento del cronograma del IMAC en cuanto a la planificación del tiempo y las tareas, mostrándose un desbalance elevado entre el tiempo estimado o planificado y el real (Ver figura).

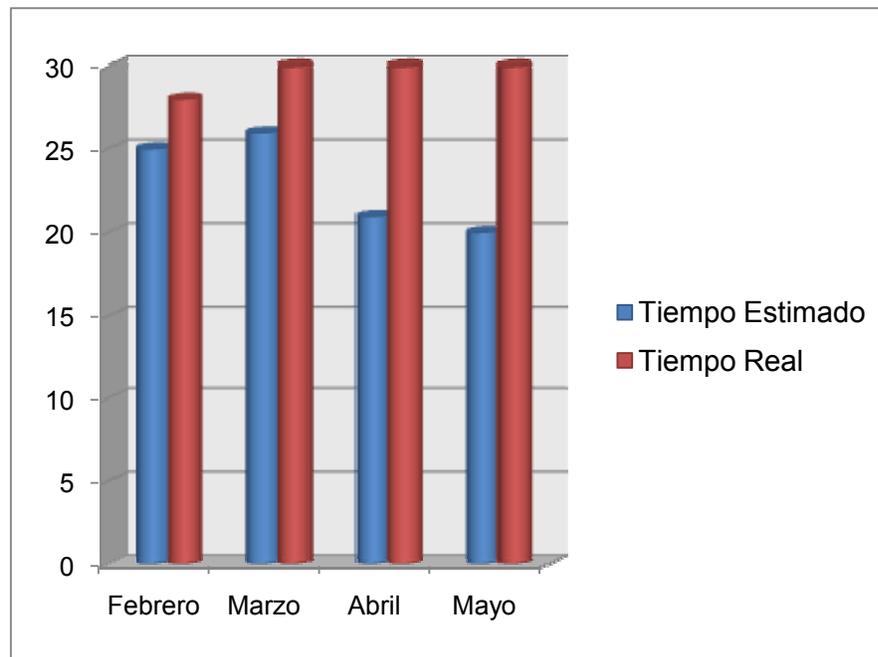


Figura 10: Tiempo Estimado y Real.

En la métrica del tiempo real se obtuvo un total de 16576 horas trabajadas en el período evaluado contra un tiempo estimado de 12244 horas de trabajo lo que indica un atraso de 32 días.

### 3.2.6 Resultados de Aplicación de las Métricas de Cobertura de las Pruebas, Madurez de las Pruebas y Restaurabilidad.

Las métricas cobertura de las pruebas, madurez de las pruebas y restaurabilidad están orientadas en función al grado de confianza que le pueda brindar el software desarrollado al usuario. En su aplicación se obtuvieron en el módulo de planificación y módulo de acciones de control los resultados mostrados en la siguiente tabla (Ver tabla 23):

	Módulo de Planificación	Módulo de Acciones de control
<b>Cobertura de las Pruebas</b>	A =204 B=204 $X = A / B$ $X = 204/204$ $X = 1$	A =15 B =23 $X = A / B$ $X = 15/23$ $X = 0.65$
<b>Madurez de las Pruebas</b>	A =203 B =204 $X = A / B$ $X = 203/204$ $X = 0.99$	A = 6 B = 23 $X = A / B$ $X = 6/23$ $X = 0.26$

<b>Restaurabilidad</b>	A =99	A =4
	B =149	B =10
	$X = A / B$	$X = A / B$
	X = 99/149	X = 4/10
	X=0.66	X=0.4

Tabla 23: Contenedora de los Resultados de las Métricas de Cobertura de las Pruebas, Madurez de las Pruebas y Restaurabilidad.

Resultados que indican que:

- Se han cubierto los casos de pruebas especificados para los dos módulos.
- Para el módulo de planificación se realizó un proceso de pruebas completo de manera satisfactoria.
- Para el módulo de acciones de control se realizó un proceso de pruebas donde no se obtuvo la madurez requerida, indicando que queda todavía por hacer en función de su completitud.
- Los módulos de planificación y acciones de control no son capaces de auto restaurarse ante algún evento anormal.

### 3.2.7 Resultados de Aplicación de las Métricas de Adecuación Funcional e Integridad de la Implementación Funcional.

las métricas de adecuación funcional e integridad de la implementación funcional evalúan la capacidad funcional del software. Al aplicarlas en el proyecto IMAC se los resultados mostrados en la siguiente tabla (Ver tabla 24):

	<b>Modulo de Planificación</b>	<b>Modulo de Acciones de control</b>
<b>Adecuación Funcional</b>	A =4 B =10 $X = 1 - A / B$ $X=1 - 4/10$ $X =0.6$	A =1 B =149 $X = 1 - A / B$ $X=1 - 1/149$ $X =0.99$
<b>Integridad de la Implementación Funcional.</b>	A=0 B=10 $X=A/B$ $X=0/10$ $X=1$	A=0 B=149 $X=A/B$ $X=0/149$ $X=1$

Tabla 24: Contenedora de los Resultados de las Métricas de Adecuación Funcional y Integridad de la Implementación Funcional.

Resultados que indican que:

- Para módulo de planificación se obtuvo que no son adecuadas las funcionalidades evaluadas.
- Para módulo de acciones de control se obtuvo funcionalidades evaluadas han sido completadas.
- Para los módulos de planificación y acciones de control se obtuvo que se ha realizado una implementación completa según las especificaciones de los requerimientos.

### **3.2.8 Veredicto conclusivo**

El proyecto IMAC en estos momentos según lo analizado se encuentra en estado crítico. Las causas se listan a continuación:

El equipo de desarrollo presenta:

- Desmotivación.
- Falta de compromiso y responsabilidad.
- No logra la coordinación de las tareas conjuntas con los clientes.
- No tiene especialización en las algunas de las áreas necesarias para desarrollar el SIGAC.

No se logra en el proyecto:

- La integración de los módulos.
- No se logra la obtención de las actualizaciones del repositorio del proyecto ERP-CUBA. Siendo la versión del 1.5 no compatible con la actual que utiliza ERP-CUBA. Lo que implica que no se puede integrar el reportador desarrollado por este proyecto.

Existe en el proyecto:

- Atraso según cronograma.
- Baja productividad.
- Falta de objetividad en las tareas a realizar.
- Riesgo de no completitud del software.

### ***3.3 Conclusiones***

Durante el desarrollo del presente capítulo se ha tratado la evaluación de la propuesta de métricas realizada en el capítulo 2, la cual arrojó resultados alarmantes del estado del proyecto IMAC. Indicador de la importancia que tiene el uso de las métricas dentro de un proyecto productivo para su evaluación, caracterización, control y mejora.

## *Conclusiones Generales*

Después de culminar con el proceso investigativo de esta tesis hacemos un análisis de lo que se ha realizado en ella, y encontramos que se ha logrado realizar:

- El análisis de métricas estandarizadas internacionalmente según la ISO/IEC TR 9126-3 que se encuentran en correspondencia con CMMI y los criterios de calidad definidos en el IMAC.
- La obtención de una guía con los pasos imprescindibles para la aplicación de las métricas propuestas como solución a la problemática surgida en el marco del IMAC.
- La realización de la validación de la propuesta de métricas, lo que demostró que la aplicación continua de dichas métricas ayuda de manera gradual obtención de la calidad en el proyecto objeto de investigación. Lo que implica desde el punto de vista teórico el cumplimiento de la idea a defender trazada.

La investigación llega finalmente a:

La obtención por parte del proyecto IMAC de una propuesta de métricas que permite la evaluación de la calidad en el mismo. Indicador de que se lograron los objetivos propuestos en la investigación.

## *Recomendaciones*

Con el objetivo de promover la continúa mejora del desarrollo del proceso creativo del software SIGAC se recomienda al IMAC:

- La continúa aplicación de las métricas propuestas.
- La actualización del registro histórico de manera sistemática.
- La generación de reportes semanales a partir de los resultados obtenidos con la aplicación de la propuesta.
- Realizar las acciones correctivas pertinentes en base a los datos aportados en el proceso de medición.

## *Referencias Bibliográficas*

1. Álvarez, Ibis M. Valdivia. 1997. Proceso general de la investigación. 1997.
2. Axentia . 2006. White Paper 2006, Una Introducción a CMMI. 2006.
3. Crosby, Philip B. 1987. La Calidad no cuesta. México : CECSA, 1987. ISBN 968-26-1220-9.
4. Dias, Aymara Marin. 2008. Propuesta de Métricas para mejorar los indicadores en el proyecto "SIGAC". Ciudad Habana : s.n., 2008.
5. Doria, H.G. 2001. Las Métricas de Software y su Uso en la Región. México. : Universidad de las Américas Puebla. UDLA: Cholula, Puebla, México., 2001.
6. Escorial. 2006. Calidad del Software. Métricas del proceso. 2006.
7. EUSKAL ESTATISTIKA ERAKUNDEA(INSTITUTO VASCO DE ESTADISTICA). 2007. ESTANDARIZACIÓN Y SISTEMATIZACIÓN DEL CÁLCULO DE LAS TASAS DE RESPUESTA. 2007.
8. Garay, Edgardo José Avilés. 2006. Estadísticas. 2006. EDUC 800.
9. González, Rolando Alfredo Hernández León y Sayda Coello. 2002. EL PARADIGMA CUANTITATIVO DE LA INVESTIGACIÓN CIENTIFICA. Ciudad de la Habana : Universidad de las Ciencias Informáticas, 2002.
10. ISO 9000:2000. 2000. NORMA INTERNACIONAL. Suiza : s.n., 2000.
11. Kan, Stephen H. 2000. Metrics and Models in Software Quality Engineering. 2000.
12. Kruchten, Philippe. 2003. The Rational Unified Process. España : Addison-Wesley Iberoamericana España, S.A., 2003. ISBN: 0321197704.
13. Lafuente, Guillermo Javier. 2000. Automatizando Métricas en la Web. Buenos Aires,Argentina : UNIVERSIDAD NACIONAL DE LUJÁN, 2000.
14. M.Álvarez, J. L. 2004. Aplicación de un Sistema Experto para el desarrollo de Sistema Evaluador del modelo Capability Maturity Model (CMM) niveles dos y tres. México : Universidad de las Américas, 2004.
15. Marco, Tom De. 1995. Why does software cost so much? New York : Dorset house, 1995.

16. Mompié, Licet Gutiérrez. 2007. Ideas para la concepción de un modelo de evaluación de calidad de. Ciudad Habana: s.n., 2007.
17. Mora, Desirée Sáenz Campos y Zahira Tinoco. 1999. INTRODUCCIÓN A LA INVESTIGACIÓN CIENTÍFICA. 1999.
18. Nuñez, Ludisley la Torre Hernández y Mariela Cepero. 2008. Propuesta de métrica de perfeccionamiento de gestión de la calidad en el proceso de desarrollo de Software. . Ciudad Habana : UCI, 2008.
19. Osorio, Carol Baragán. 2009. Programación de agenda de actividades en la unidad administrativa. 2009.
20. Pfleeger, Shari. 2002. Ingeniería del Software. Teoría y Práctica. s.l.: Prentice Hall, 2002.
21. Real Academia Española. 2001. DICCIONARIO DE LA LENGUA ESPAÑOLA - Vigésima segunda edición. 2001.
22. S.Pfleeger, N.Fenton y. 1997. Software Metrics:A Rigorous Aproach. s.l. : Chapman & Hall, 1997.
23. S.Pressman, Roger. 2005. Ingenieria del Software: Un enfoque práctico. Sexta Edición. España: McGraw-Hill, 2005. ISBN: 9701054733.
24. Sociedad Latinoamericana para la calidad. 2000. Gráfica de Pareto. 2000.
25. Thompson, Ben. 2006. Técnicas de Investigación. 2006.
26. Wesfall, Linda. 2000. 12 Steps to Useful Software Metrics. 2000.

## *Bibliografía*

1. ISO, 200, IDT. INGENIERÍA DE SOFTWARE—CALIDAD DEL PRODUCTO—PARTE 1: MODELO DE LA CALIDAD, NORMA CUBANA ISO/IEC 9126-1: 2005.
2. Álvarez, J. C. (2007). Controles y Métricas Técnicas del Software. Controles y Métricas Técnicas del Software. Ciudad Habana: Grupo GILSIIS, Universidad de Granada, España.
3. Asociación Española de Métricas del Software (AEMES). (2007). Volumen 4 Número 1. Revista de Procesos y Métricas de las tecnologías de la información.
4. C., M. P. (1995). Modelo de Gestión de Proyectos Software: Estimación del Esfuerzo de Desarrollo. Modelo de Gestión de Proyectos Software: Estimación del Esfuerzo de Desarrollo. Depto. Ing. Informática y Cs. de la Computación. Universidad de Concepción.
5. Díaz, Y. P. (2007). SIMETSE – Sistema de Métricas para evaluar el software educativo. Ciudad de la Habana: UCI.
6. Dirección de Calidad de Software.UCI. (2009). Proceso de Medición de Software. Métricas. (pág. Proceso de Medición de Software. Métricas.). Ciudad de la Habana: UCI.
7. Dra. Coral Calero, D. M.-M. (2005). Programa de doctorado: Arquitectura y Gestión de la Información. CALIDAD DE SISTEMAS DE INFORMACIÓN.
8. Estrada, M. L. Medir el proceso de control de configuración, ¿una utopía para la Industria Nacional de Software? Ciudad Habana, Cuba: Instituto Superior Politécnico “José Antonio Echeverría”.
9. Fuente, A. A. Necesidad de Sistemas Formales de Métricas para proyectos de software OO. Universidad de OVIEDO.
10. Hinojosa, L. C. (2008). MÉTRICAS ESTANDARIZADAS INTERNACIONALMENTE, PROPUESTAS PARA EVALUAR LA CALIDAD DE LOS PRODUCTOS DE SOFTWARE. Ciudad de la Habana.
11. Humphrey. (1989). Managing the Software Process. SEI Series in Software Engineering.
12. Ing. Esteban Vargas, A. e.-S. PLAN DE MÉTRICAS EN OCHO PASOS.
13. Querétaro. (2006). ISO 9126-3: Métricas Internas de la Calidad del Producto de Software.
14. Ramírez, A. A. Propuestas de Métricas para evaluar el flujo de trabajo de Análisis y Diseño. Ciudad de La Habana: 2008.

15. Rodríguez, R. R. (2008). Evaluación de posibles elementos a considerar en la integración de los métodos de Boehm y Humphrey para la estimación de la duración de un proyecto de software para aplicaciones Multimedia. Ciudad Habana: UCI.
16. Rossi, L. B. (2002). METRICAS DE SOFTWARE.
17. Saffirio, M. (2009). Tecnologías de Información y Gestión de Procesos de Negocios (BPM) .
18. TAPIA, R. G. (2003). MAESTRIA EN SISTEMAS DE INFORMACION. Universidad Autonoma de TAMAULIPAS.
19. Albuquerque, J., Coelho, C., & Meira, S. (1998) Practical limits of the PSP model. SoST'98
20. Symposium on Software Technology.Process Improvement: Putting Software Engineering to Work. Soc.Argentina de Inf. & Invetigacion Oper, Buenos Aires, Argentina; 1998; 230 pp. p.95100.

## *Glosario de Términos*

### ----A----

**Aseguramiento de Calidad:** (*quality assurance*, ISO 8402, 1994) Todas las actividades planificadas y sistemáticas necesarias para aportar la confianza suficiente en que un producto o servicio cumplirá con unos requisitos dados de calidad.

### ----C----

**Calidad de Software:** “Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente” (R.S.Pressman, 1993)

**Capacidad de Recuperación:** (*recoverability*, ISO 9126) Subcaracterística de fiabilidad, que indica la capacidad del sistema para restablecer su nivel de respuesta después de un fallo crítico o error hardware.

**CMMI: Capability Maturity Model Integration.** Modelo para la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software. Fue desarrollado por el Instituto de Ingeniería del Software de la Universidad Carnegie Mellon (SEI), y publicado en su primera versión en enero de 2002.

**Cocomo II:** Método matemático que consiste básicamente en la aplicación de ecuaciones matemáticas sobre los Puntos de Función sin ajustar o la cantidad de líneas de código (SLOC, Source Lines Of Code) estimados para un proyecto.

### ----D----

**Defecto:** Fallo o deficiencia durante el proceso de creación de programas para ordenadores. El mismo puede presentarse en cualquiera de las etapas del ciclo de vida de desarrollo del software. Reduce la capacidad del software para cumplir de manera completa y eficiente la necesidad de los usuarios

---- *E* ----

**Eficiencia:** (*Efficiency*, ISO 9126) Conjunto de características que determinan la relación entre el nivel de rendimiento del software y el número de recursos usados, bajo ciertas condiciones dadas. Se divide en las subcaracterísticas comportamiento temporal, utilización de recursos.

**Error:** (*error, mistake*, BCS SIGIST) Una acción humana que puede producir resultados incorrectos.

**Experto:** Persona que posee extensos conocimientos o experiencia específica con respecto a una organización, proceso, actividad o materia.

---- *F* ----

**Fallo:** (*Failure*, BCS SIGIST) Una desviación del funcionamiento esperado.

**Fiabilidad:** (*reliability*, ISO 9126) Grado en que el sistema responde bajo las condiciones definidas durante un intervalo de tiempo dado. Se divide en las subcaracterísticas madurez, tolerancia a fallos, capacidad de recuperación.

**Funcionalidad:** (*functionality*, ISO 9126) Grado en que las necesidades asumidas o descritas se satisfacen. Se divide en las subcaracterísticas idoneidad, precisión, interoperabilidad, seguridad.

---- *G* ----

**Gestión:** Actividades coordinadas para dirigir y controlar una organización.

---- *M* ----

**Madurez:** (maturity, ISO 9126) Subcaracterística de fiabilidad, que indica la frecuencia con que ocurren los fallos.

---- *N* ----

**Normas:** Modelo, patrón o regla de obligado cumplimiento.

---- *O* ----

**Operabilidad:** (*operability*, ISO 9126) Subcaracterística de facilidad de uso, que indica las características del software que influyen en el esfuerzo del usuario para operar y control operacional.

---- *P* ----

**Proyecto:** Proceso único consistente en un conjunto de actividades coordinadas y controladas con fechas de inicio y de finalización, llevadas a cabo para lograr un objetivo conforme con requisitos específicos.

---- *R* ----

**Riesgo:** Un riesgo es una contingencia o daño que en cualquier momento puede materializarse o no hacerlo nunca.

---- *S* ----

**Seguridad:** (*security*, ISO 9126) Subcaracterística de funcionalidad, que indica el grado en que un acceso no autorizado (accidental o deliberado) se prevenga y se permita un acceso autorizado.

---- *V* ----

**Validación:** Comprobación de que se está construyendo el producto correcto.

**Verificación:** Comprobación de que se está construyendo el producto correctamente.

## Anexos

### **Anexo Nro. 1: Encuesta Realizada a los Administradores de Calidad de los Proyectos en la UCI.**

#### **A quién pueda interesar:**

La presente encuesta esta dirigida a los administradores de calidad de los proyectos de producción de software que se desarrollan en la UCI. Tiene el propósito de contribuir con la investigación y desarrollo de un trabajo de diploma.

**Nota:** Posee carácter confidencial.

#### **Compañero/a:**

Ud. es el administrador de calidad de su proyecto al cual le solicitamos que responda este cuestionario, que tiene como propósito ayudar en el desarrollo de un trabajo de diploma. Esperamos que colabore y para ello le pedimos que al responder las preguntas lo haga con la mayor seriedad posible, ya que de ello dependerá en gran medida el resultado de nuestro trabajo, muchas gracias de antemano por su ayuda.

#### **1- Nombre de Proyecto:**

---

#### **2- Facultad a la que pertenece:**

---

#### **3- Tiempo de vida del proyecto:**

---

#### **4- Tiempo que lleva como administrador de calidad del proyecto:**

---

#### **5- ¿Estás familiarizado con el término: métricas del software? Marque con una X la respuesta.**

Si ---- No-----

#### **6 - ¿Consideras importante su aplicación? Marque con una X la respuesta.**

Si ---- No-----

**7- En que etapa/s del desarrollo del software se deben aplicar las métricas. Marque con una X la respuesta.**

- Todo el ciclo de vida
- Cuando se este implementando.
- Cuando se este diseñando.
- Cuando se estén llevando a cabo las pruebas.
- En la fase de Análisis

**8 - ¿Se aplican métricas en tu proyecto? Marque con una X la respuesta.**

Si ---- No-----

**9 - Conoces herramientas para la gestión de métricas.Marque con una X la respuesta.**

Si ---- No-----

**10 - En caso de ser positiva la pregunta anterior menciónalas y diga que métrica conoce que gestiona:**

-----

-----

-----

**11- La siguiente tabla tiene dos columnas, la primera (Métricas) contiene un conjunto de métricas, la segunda columna (Importancia), la cual se les pide completar, definiendo el grado de importancia que Ud. le confiere a las métricas que se muestran partiendo de las que considere más importante aplicar en un proyecto según su criterio y experiencia; (Clasifíquelas según la siguiente escala: 1\_Importante; 2\_No importante). Se les pide además que según su criterio si considera que faltan algunas métricas las incluya al final, pues nos sería de gran ayuda.**

Métricas	Importancia
Tiempo: Tiempo real dedicado por la persona en cada una de las tareas ejecutadas como parte del proyecto.	
Tiempo estimado: Tiempo estimado por la persona en cada una de las tareas que debe ejecutar como parte del proyecto. Se expresa en minutos.	
Complejidad estimada: Estimado de complejidad de las tareas.	

Productividad: la productividad es una métrica que calcula la generación de producto por unidad de tiempo.	
Defectos removidos: Cantidad de defectos eliminados.	
Eficacia en la Eliminación de Defectos. Revela la relación entre el índice de defectos y los defectos removidos.	
Índice de defectos: Revela la cantidad real de defectos que existen en el producto.	
Porcentaje de defectos por tipo: Revela el porcentaje de defectos por tipo que se encuentran en el software.	
Integridad de la implementación funcional: es una medida de cuán completa ha sido la implementación según la especificación de requisitos, se detectan el número de funciones pérdidas, aquellas que fueron descritas en la especificación de requisitos y no fueron implementadas.	
Comprensibilidad de entradas y salidas: La métrica Comprensibilidad de las entradas y salidas permite hacer un análisis de cuánto pueden los usuarios comprender lo que se requiere como entrada y lo que suministra el sistema de software como salida	
Adecuación funcional: La métrica Adecuación Funcional permite hacer un análisis de cuán adecuada es la función evaluada.	
Cobertura de las pruebas: Esta métrica indica cómo se van cumpliendo los casos de prueba especificados, por lo tanto mientras mayor sea la cobertura, mayor número de casos de prueba se estarán cumpliendo.	
Madurez de las pruebas: La métrica madurez de las pruebas se orienta en función de qué tan bien se está desarrollando el proceso de pruebas	
Restaurabilidad: Se la métrica restaurabilidad permite hacer un análisis de cuán capaz es el producto de auto restaurarse luego de un evento anormal o una solicitud.	

Efectividad de la restauración: Con el objetivo de hacer un análisis de cuán efectiva es la capacidad de restauración se define la métrica eficacia de restauración.	
Identificación de los Riesgos: Revela la cantidad de riesgos identificados.	
Probabilidad de ocurrencia de un riesgo identificado: Revela la probabilidad de ocurrencia de un evento de riesgo identificado.	
Efectividad de la mitigación de riesgos: Con el objetivo de determinar la relación existente entre los riesgos mitigados y el total de riesgos identificados.	
Estimar el esfuerzo (personas-mes): Permite estimar el esfuerzo de las personas por mes.	

**Otras que consideres importantes y que faltan:**

-----  
 -----  
 -----.

**12 - Sugerencias que desee hacer:**

-----  
 -----  
 -----.

## **Anexo Nro. 2: Entrevista desarrollada para los Trabajadores de Calidad UCI que Atienden el Área de Métricas.**

- 1) ¿Por qué son importantes las métricas?
- 2) ¿Cómo se ven afectados los proyectos con la no aplicación de métricas?
- 3) ¿Por qué es importante que cada proyecto seleccione individualmente métricas para aplicar en su marco de trabajo?
- 4) ¿Cómo se debe llevar a cabo el proceso de selección de métricas en los proyecto?
- 5) ¿Qué herramientas existen que permitan la gestión de las métricas?
- 6) ¿Qué se debe tener en cuenta a la hora de elegir métricas para aplicar en los proyectos?
- 7) ¿Existe un Plan de Medición correctamente descrito y detallado para los proyectos en la UCI?
- 8) ¿Qué implica que la UCI este optando por el nivel 2 de CMMI?