

MODELO DE EVALUACION DE CALIDAD DEL SOFTWARE DE GESTION

REINALDO SIERRA PRIETO



**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
MAESTRÍA EN INGENIERÍA AREA DE INFORMATICA
BUCARAMANGA
2008**

MODELO DE EVALUACION DE CALIDAD DEL SOFTWARE DE GESTION

REINALDO SIERRA PRIETO

Proyecto de Investigación para Optar el Título de:

Magíster en Informática

Director

Msc. MANUEL GUILLERMO FLOREZ BECERRA



**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
MAESTRÍA EN INGENIERÍA AREA DE INFORMATICA
BUCARAMANGA**

2008

DEDICATORIA

A ti DIOS que me diste la oportunidad de vivir y regalarme un Familia maravillosa, en cabeza de mi mejor amiga, compañera y esposa, Luz Stella, por soportar estos años lejos de ella, por acompañarme en los buenos y malos momentos y por ayudarme a que este momento llegara.

No puedo dejar de lado a mis dos Hijos, Sergio David y Juliana, quienes me prestaron este tiempo para poder salir adelante, tiempo que no podré devolver y que cada momento que hablaba con ellos me preguntaban “Papi cuando regresas para quedarte” y que hoy al fin les puedo decir, regrese para quedarme siempre.

El presente trabajo de investigación está dedicado a toda mi Familia y a todos aquellos que con su apoyo moral, económico me alentaron en esta Maestría.

REY.

AGRADECIMIENTOS

- A la Universidad del Quindío por su colaboración y apoyo para poder realizar esta Maestría y especialmente a la Facultad de Ciencias Económicas y Administrativas.
- A los que leyeron la tesis, porque con sus comentarios y observaciones me permitieron enriquecerla, como los Profesores Luís Carlos Gomez y Fernando Rojas de la Universidad Industrial de Santander, José Fernando Echeverry Murillo de la Universidad del Quindío y Luís Chaparro de la Universidad de Boyacá.
- A Msc. Manuel Guillermo Florez Becerra, mi Director, por el tiempo que dedico a la lectura de la tesis, observaciones apropiadas, por su guía y apoyo constante.
- Al Ing. Hernando Prieto Martínez por sembrar esa semilla de la CALIDAD en mi pregrado y hoy poder decir que esa semilla creció y dio un fruto que se llama "Modelo de Evaluación de Calidad".
- A mis compañeros Genny, Maria Fernanda, Emma Isabel, Elkin y Luís Enelso, que con sus aportes y comentarios en seminario III y IV, enriquecieron y colaboraron de una u otra forma en el desarrollo de esta Tesis
- A la Familia Prieto Alzate, por acogerme durante este tiempo en su seno familiar y no ser tan triste la lejanía de mi Esposa e Hijos.

TABLA DE CONTENIDO

NOTA DE ACEPTACIÓN	iii
DEDICATORIA	iv
AGRADECIMIENTOS	v
TABLA DE CONTENIDO	vi
LISTA DE TABLAS	viii
LISTA DE FIGURAS	ix
RESUMEN	x
ABSTRACT	xi
INTRODUCCION	12
1. PLAN FUNDAMENTAL	13
1.1 SITUACIÓN PROBLEMA.....	13
1.1.1 Contexto.....	14
1.1.2 Identificación y descripción del problema	14
1.1.3 Causas de las debilidades de la calidad del software	16
1.1.4. Consecuencias.....	17
1.1.5. Alternativas de solución.....	18
2. OBJETIVOS DE LA INVESTIGACIÓN	20
2.1. OBJETIVO GENERAL.....	20
2.2. OBJETIVOS ESPECIFICOS.....	20
3. JUSTIFICACIÓN	21
4. MARCO DE REFERENCIA	23
4.1 ANTECEDENTES HISTORICOS.....	23
4.2 ESTADO DEL ARTE	25
4.4 MARCO CONCEPTUAL.....	39
4.4.1. Calidad	39
4.4.2 Métrica:	40
4.4.3 Métrica Directa:	40
4.4.4 Métrica Indirecta:.....	40
4.4.5 Modelos de calidad del software, Modelo factores/criterios/métricas de McCall.....	40
4.4.6 Modelo de Dromey	42
4.5. MARCO LEGAL.....	43
5. ALCANCES Y LIMITACIONES	48
5.1 ALCANCES	48
5.2. LIMITACIONES	48
6. FORMULACIÓN DEL PROBLEMA	50
7. HIPOTESIS	50
7.1. VARIABLES.....	50
7.1.2. Definición nominal y operacional, indicadores de las variables.....	50
8. METODOLOGIA y TECNICAS	56
8.1. TIPO DE ESTUDIO	56
8.1.1. Exploratorio	56
8.1.2. Descriptivo.....	56
8.2. METODOS USADOS	56
8.2.1. Modelo empleado para la Evaluación.....	56

8.2.2. Inductivo.....	58
8.2.3. Deductivo.....	58
8.2.4. Analítico.....	58
8.2.5. Síntesis.....	58
8.3. TECNICAS.....	59
8.3.1. Recolección de información.....	59
8.3.1.2. Entrevista a usuarios finales.....	59
8.3.2. De Campo.....	59
8.3.2.1. Diseño de muestreo.....	59
8.3.2.2. Ilustración del Modelo.....	60
9. ANALISIS DEL SISTEMA.....	61
9.1 SISTEMA ACTUAL.....	61
9.2 ANALISIS DE REQUERIMIENTOS.....	62
9.2.1 Determinación del Requerimiento.....	62
9.3 ESTUDIO DE VIABILIDAD.....	63
9.3.1 Viabilidad Operativa.....	63
10. PROPUESTA DE MODELO.....	64
10.1 MODELO DE EVALUACIÓN A USAR.....	64
10.1.1 Modelo.....	65
10.2 SELECCION DE FACTORES / CARACTERISTICAS.....	68
10.2.1 Modelo Conceptual UML para la Estándar 9126.....	72
10.2.2 Modelo Conceptual UML para Métricas.....	73
10.3 FORMULACION DEL MODELO.....	74
10.3.1 Definición de Perfiles en el Modelo.....	80
10.4 VALORACION DEL MODELO.....	81
10.4.1 Valoración (Resultados – Aceptación o Rechazo).....	84
10.5 CASO DE ESTUDIO.....	84
10.5.1 Análisis de Resultados.....	88
10.6 VALIDACIÓN DEL MODELO.....	91
11. CONCLUSIONES.....	93
11.1 LOGROS ALCANZADOS.....	93
11.2 APRENDIZAJE OBTENIDO.....	94
11.3 FUTUROS TRABAJOS.....	95
11.3.1 Recomendaciones.....	95
REFERENCIAS.....	96
ANEXOS.....	102
ANEXO 1- ORGANIZACIONES.....	102
ANEXO 2- CATALOGO DE ESTÁNDARES.....	103
ANEXO 3 - ENCUESTAS.....	106
ANEXO 4 – CARACTERÍSTICAS, SUBCARACTERÍSTICAS Y MÉTRICAS.....	110
ANEXO 5. CASO DE ESTUDIO, EVALUACIONES.....	114

LISTA DE TABLAS

Tabla 1. Factores de Calidad, según Modelo de McCall	41
Tabla 2. Agrupamiento de las características de calidad según Dromey	42
Tabla 3. Características de Calidad de diferentes Mod./Estánd. de Calidad del Software	64
Tabla 4. Cantidad de las características de Calidad de los Modelos/Estándares de	65
Tabla 5. Criterios Relevantes de una Buena Calidad “Usuario”	68
Tabla 6. “ILIDAES” de la Calidad del Software	69
Tabla 7. Matriz de Criterios Relevantes vs. Ilidades.....	70
Tabla 8. Ilidades – Características de Alto Nivel.....	71
Tabla 9 Característica de Funcionalidad en el Modelo de Evaluación de Calidad.....	75
Tabla 10. Característica de Fiabilidad en el Modelo de Evaluación de Calidad.....	76
Tabla 11. Característica de Usabilidad el Modelo de Evaluación de Calidad	77
Tabla 12. Característica de Mantenibilidad el Modelo de Evaluación de Calidad.....	78
Tabla 13. Característica de Portabilidad el Modelo de Evaluación de Calidad.....	79
Tabla 14. Característica de Eficiencia el Modelo de Evaluación de Calidad.....	80
Tabla 15. Roles de Acuerdo a las Características.....	81
Tabla 16. Tipos de Métrica	81
Tabla 17. Caso de Estudio Producto A, Evaluador Ingeniero 1	85
Tabla 18. Caso de Estudio Producto B, Evaluador Ingeniero 1	86
Tabla 19. Caso de Estudio Producto A, Evaluador Ingeniero 2.....	87
Tabla 20. Caso de Estudio Producto B, Evaluador Ingeniero 2.....	87
Tabla 21. Resultados de la Evaluación de calidad para el producto A y B	88
Tabla 22. Resultados de la Evaluación de Calidad de las subcaracterísticas para el producto A y B	88
Tabla 23. Validación del Modelo - Coeficiente de Correlación	92

LISTA DE FIGURAS

Figura 1. Niveles de Madurez de un proceso de Software	28
Figura 2 . PSP (Process Software Personal)	29
Figura 3. Visiones de los Factores de Calidad según el Modelo de McCall.....	41
Figura 4. Matriz de Factores de Calidad según Dromey (traducción).....	43
Figura 5. Calidad en el Ciclo de Vida según ISO/IEC 9126-1.....	45
Figura 6. Relación Métricas del Modelo / Atributos en ISO/IEC 9126-1	46
Figura 7. Modelo de Calidad Interna y Externa de ISO/IEC 9126-1	47
Figura 8. SISTEMA ACTUAL: Descripción selección del software.....	61
Figura 9. Modelo de Evaluación (Tomado [41], Traducido y Adaptado de ISO/IEC 9126)67	
Figura 10. Modelo Conceptual UML para la ISO/IEC 9126 (tomado y traducido de ISO/IEC 9126)	72
Figura 11. Modelo Conceptual UML para Métricas (Adaptado de Zuse, H., 1998)	73
Figura 12. Elementos del Modelo de Evaluación de la Calidad.....	74
Figura 13. Gráfica de Comparación de Subcaracterísticas Producto A y B.....	89
Figura 14. Gráfica de Subcaracterísticas Evaluador 1 y 2.....	90
Figura 15. Gráfica de Características Evaluador 1 Producto A y B	91

RESUMEN

TITULO: MODELO DE EVALUACIÓN DE CALIDAD DEL SOFTWARE DE GESTIÓN.*

AUTOR: REINALDO SIERRA PRIETO**

PALABRAS CLAVE: Calidad del Producto Software, Modelos de Evaluación, Modelos de Calidad, Norma ISO 9126, Características de Calidad, Evaluación de calidad del software.

DESCRIPCIÓN:

El propósito de este trabajo de investigación es proponer un modelo de evaluación de la calidad del software de gestión, guiado por la ISO/IEC 9126-1:2001, donde se logren beneficios que permita a los representantes de las empresas determinar el grado de calidad conociendo un juicio de las características de calidad del software a adquirir. El software como producto es el resultado de una inversión y por lo tanto se espera un beneficio económico de su uso y una satisfacción de los requerimientos.

Inicialmente, se presentó la situación actual para la adquisición de software de gestión, donde se identificó el problema realizándose un diagnóstico, definiendo debilidades, causas, consecuencias y alternativas de solución. También se presentaron los alcances y limitaciones de la investigación y como la calidad del software puede ser medida por medio de características, subcaracterísticas y métricas de calidad del software. Por tanto, se debe tener en cuenta factores como son: Funcionalidad, Usabilidad, Fiabilidad, Mantenibilidad, Flexibilidad, Portabilidad, tal como lo expone la norma ISO 9126. Con base en los factores anteriormente mencionados, se investigaron, ordenaron las características, las subcaracterísticas, como también se establecieron los valores mínimos aceptables para un "Modelo de Evaluación de Calidad del Software de Gestión", además, para ejemplificar el modelo se presentó un caso de estudio.

* Trabajo de Investigación

** Facultad de Físico Mecánicas, Programa de Posgrado: Maestría en Ingeniería Área de Informática.
Director: Ing. Msc. Manuel Guillermo Florez Becerra

ABSTRACT

TITLE: MODEL OF EVALUATION QUALITY OF THE MANAGEMENT SOFTWARE *

AUTHOR: REINALDO SIERRA PRIETO**

KEYWORDS: Quality Software Product, Evaluation Models, Quality Models, ISO 9126, Quality Characteristics, Evaluation of Quality Software.

DESCRIPTION:

The purpose of the this research is to propose a model of the evaluation quality of the management software, where benefits are obtained that allow the representatives of the companies to determine the degree of the quality knowing the I diagnose technician of the characteristics of the quality software to acquire. The software like product is the result of an investment and therefore it is expected an economic benefit of its use and a satisfaction of the requirements. .

Initially, the current situation was presented for the acquisition of management software, where the problem was identified being carried out a diagnosis, defining weaknesses, causes, consequences and alternative of solution. Also the reaches and limitations of the investigation were presented and as the quality software it can be measured by means of characteristic, sub characteristics and metric of quality of the software. Therefore, it should be kept in mind factors like they are: Functionality, Usability, Reliability, Maintainability, Flexibility, Portability, just as it exposes it the norm ISO 9126. With base in the previously mentioned factors, they were investigated, they ordered the characteristics, the subcharacteristics, as well as the acceptable minimum values settled down for a Model of Evaluation Quality of the Management Software, also, to illustrate the pattern a case of study it was presented.

* Research Work

** Faculty Physical and Mechanical Engineering, Postgraduate Program: Masters in Engineering Area of Computer science. Director: Engineer. Msc. Manuel Guillermo Florez

INTRODUCCION

La calidad se considera actualmente uno de los activos principales con los cuales una firma puede realzar su posición competitiva. Ésta es una razón por la que la calidad ha llegado a ser esencial para asegurarse de que los productos existentes en una compañía resuelven las necesidades de clientes. Una innovación reciente en el área de los sistemas es el desarrollo de un sistema de mecanismos y de modelos para la calidad de evaluación. Diversos modelos de la calidad fueron estudiados: McCall (Mc Call y otros en 1977), Dromey (Dromey, 1996) e ISO 9126 (ISO/IEC 9126, 1991 y 2001) en un intento por identificar los aspectos presentes en estos modelos que se juzguen importantes en un modelo de calidad.

Se propone un modelo guiado por la ISO 9126-1 que refleja las características esenciales de la calidad de un producto software, y estas se pueden aplicar al software de gestión, más específicamente al software contable.

1. PLAN FUNDAMENTAL

1.1 SITUACIÓN PROBLEMA

El desarrollo de la informática y en especial del software, se ha convertido en la herramienta que conduce a la toma de decisiones comerciales, sirve como base de investigación científica y moderna de resolución de problemas de ingeniería y es el factor clave que diferencia los productos y servicios modernos.

El software como producto es el resultado de una inversión y por lo tanto se espera un beneficio económico de su uso y una satisfacción de los requerimientos. Luego, se hace necesario un sistema sólido de control de calidad en el software que permita evaluar la concordancia con los requisitos funcionales y de rendimientos establecidos, los estándares de desarrollo explícitamente documentados y las características implícitas que se espera de todo software desarrollado profesionalmente.

La calidad del software puede ser medida, esto indica que, para su control se definen parámetros, indicadores o métricas de medición de la calidad del software se deben tener en cuenta algunos factores como son:

- Usabilidad.
- Fiabilidad.
- Corrección.
- Integridad.
- Mantenibilidad.
- Flexibilidad.
- Reusabilidad
- Portabilidad.
- Escalabilidad.
- Seguridad.

Por lo tanto, el desafío es uno solo. Al obtener calidad con énfasis en la evaluación de un Producto Software, se garantiza a los usuarios la adquisición de software de calidad, además de asegurar a la empresa o al mismo usuario que el producto software cumple con los factores mencionados.

Con base en los factores anteriormente mencionados, se seleccionarán las características que harán parte del **“Modelo de Evaluación de Calidad de Software de Gestión”**, con el fin de asegurar que las características que definen

de manera individual cada producto software satisfagan los requerimientos del sistema y las necesidades del usuario.

1.1.1 Contexto.

Se desarrollará un “**Modelo de Evaluación de Calidad de Software de Gestión**”, con el fin de asegurar que las características que definen de manera individual cada producto software satisfagan los requerimientos del sistema y las necesidades del usuario.

El modelo servirá como herramienta de soporte para gerentes, ingenieros y administradores de áreas de sistemas informáticos, conocer una opinión técnica sobre las características de calidad de un producto software antes de tomar una decisión de adquisición y así poder determinar la aceptación o rechazo del mismo.

1.1.2 Identificación y descripción del problema

En la actualidad existen diversos tipos de software, como son: software de sistemas, software de desarrollo, software de gestión, software de ingeniería y científico, software empotrado, software de computadoras personales y software de inteligencia artificial, entre otros; los cuales pueden ser sometidos a un control de calidad específica (ya sea procedimental u orientado a objetos), evaluando sus características se podrá conocer la calidad del producto software con respecto a los requerimientos de funcionamiento y necesidades del usuario ^{2 3}

Algunos procesos y metodologías desarrolladas con características de calidad son:

- W. Edwards ⁴, desarrolló una metodología basada en información estadística que permitía determinar el comportamiento de los procesos, a la que denominó “Control Estadístico de Calidad (CEC)”.
- Herramientas CASE ⁵ las cuales aceleran el proceso de desarrollo de sistemas y mejoran la calidad de los mismos, automatizando una o más fases del ciclo de vida de sistemas.
- Despliegue de la función de calidad: Es una técnica de gestión de calidad que traduce las necesidades del cliente en requisitos técnicos del software, hace énfasis en lo que resulta valioso para el cliente y después despliega estos

² Weyuker, E.J. (1988). Evaluating software complexity measures.

³ IEEE Transactions on Software Engineering. 14(9). pp.1357-1365.

⁴ Cruz Ramírez, José, “Historia de la calidad” en EXCELLENTIA, pp. 8-14

⁵ Análisis y diseño de sistemas de información. Tercera Edición Jeffrey L. Whitten. Pag. 172.

valores a lo largo del proceso de ingeniería. Originalmente desarrollado en Japón a principios de los años 70. *

De acuerdo a lo expuesto anteriormente, estos han sido diseñados para evaluar la calidad del software en su proceso de desarrollo, pero no para la evaluación de un producto software terminado. Por lo tanto se ha enfocado el interés en el desarrollo de un Modelo de evaluación de calidad del software de Gestión **(MECS)**.

Se determinó que el tipo de software al cual se va a evaluar su calidad, será el software de gestión, debido a que el procesamiento de información comercial constituye la mayor de las áreas de aplicación del software, tanto en su desarrollo como en su adquisición. [1]

El software de gestión, es aquel software que utilizan las empresas para la automatización de sus procesos de gestión y tiene unas características específicas: - El software de gestión debe adaptarse a una regulación jurídica y administrativa concreta.

- El software de gestión en su origen esta fragmentado funcionalmente, según el área de la empresa a la que se dirige, siendo abundantes las aplicaciones estrictamente departamentales. Esta característica esta evolucionando muy rápidamente hacia productos orientados a resolver de forma integrada los procesos de gestión. Las grandes empresas han sido pioneras en la introducción de este tipo de software y de su mano empresas de ámbito global se han introducido en este segmento de mercado (empresas del sector salud y educativo).
- En la misma medida que la empresa es una realidad cambiante, necesita que su software acompañe esta evolución. Por una parte el crecimiento de volumen de actividad supone en muchos casos la reforma de los procesos de gestión. También los cambios organizativos, la ampliación de actividades y nuevos productos. Por todos estos motivos el software de gestión está en permanente evolución y requiere de mantenimiento para adecuarse en cada momento a la realidad empresarial.

Como ejemplo de software de gestión se tienen ^{*} Helisa, software Administrativo y de Gestión de cualquier tipo de empresa. Heinsohn, solución empresarial integrada para la gestión administrativa, financiera para el sector privado y público.

* En 1972, el astillero de la Mitsubishi en Kobe hace un gran avance en los conceptos modernos de calidad, profundizando y centrandos los conceptos del Hoshin Kanri. Esto resultó en un enfoque ampliado de la Calidad desde el diseño de un producto hasta su consumo o uso, lo que llamaron Despliegue de la Función de Calidad o QFD (Quality Function Deployment), que derivaría posteriormente en el concepto de Concurrent Engineering

* Catalogo de Software y Guía de Servicios Informáticos. 15ª Edición 2006. Guía completa de información de los principales productos, servicios y empresas proveedoras en Colombia.

Sistema Uno, Software Integrado de Información empresarial. Novasoft, Software de Gestión Integrado, que integra la gestión financiera, administrativa y Comercial. Xenco, software de gestión administrativa y financiera, con sus productos Safix-ERP. Donde las soluciones de gestión antes mencionadas, poseen los sistemas de inventario, nómina, cartera, contabilidad, presupuesto, facturación, etc.

La garantía de calidad de software es una actividad de protección que se aplica a lo largo de todo el proceso de ingeniería de software, por lo tanto, el desconocimiento de los procesos, métodos, herramientas, revisiones técnicas formales que son aplicadas durante el proceso del software, originan deficiencias en la calidad del software.

1.1.3 Causas de las debilidades de la calidad del software

Factores^{*} de software que conllevan a que este posea ciertas debilidades, como la falta de:

- a) Corrección, se debe a que no se realizó una adecuada fase de análisis en donde se determinara con exactitud los requerimientos del cliente, lo cual traslada el problema a la fase de diseño en donde a través de distintas técnicas y procesos se define el sistema con el suficiente detalle como para permitir su realización física. A demás es probable que en la fase de codificación los programadores incurran en errores lógicos que se traducen en resultados incorrectos en el momento de ejecutar el software.
- b) Integridad, que se origina en una mala gestión en fase de diseño y desarrollo, debido a que no se realizan las validaciones necesarias para los diferentes tipos de datos, mala definición de los usuarios que los usan, concesión de permisos y en general toda la seguridad de la información que contiene el sistema.
- c) Flexibilidad, esta relacionado con la dificultad de mantenimiento, y se debe fundamentalmente a la falta de modularidad y soporte adecuado para realizar los cambios que requiera el programa.
- d) Portabilidad, es causada en las fases de diseño y desarrollo por no establecer un sistema abierto que establezca conectividad con otros entornos (Hardware y Software).
- e) Fiabilidad, causada por la falta de concordancia con los requisitos debido a Defectos de diseño e implementación.
- f) Seguridad, causada por la deficiencia en la disponibilidad de mecanismos que controlen o protegen los programas y los datos.
- g) Escalabilidad, Es causada por un análisis y diseño que no permite los tratos con los aspectos adaptables a nuevos requisitos conforme cambian las necesidades del negocio o actuaciones para satisfacer las nuevas

^{*} Una de las definiciones más difundidas es la de Mc Call (1977), que especifica una serie de factores, como se expone en el capítulo 4.4.5

necesidades. No se podrá incluir la adición un nuevo tipo de deducción (ejemplo, en un programa de la nómina) existente o agregar una interfaz a un nuevo sensor en un orden y sistema de información de mando.

Como también se encuentra dificultad en:

- h) Mantenimiento se debe a que en el diseño no se haya tenido en cuenta la Modularidad del sistema, y que no exista el soporte necesario para modificar los programas adecuadamente.
- i) El uso es causada básicamente por un deficiente diseño de interfaces de usuario, además de no existir capacitación adecuada y oportuna sobre el manejo del sistema.
- j) Auditoria se origina por deficiencias en la implementación del sistema de control realizado durante la fase de análisis y diseño.
- k) La reusabilidad causada por la incompatibilidad del software en el tipo de herramienta bajo la cual fue desarrollado y por el alcance que pueda tener el programa en las funciones que va a realizar.

1.1.4. Consecuencias

Las consecuencias que se derivan de la falta de calidad en el software son:

La falta de Corrección: Hace que el programa no opere correctamente proporcionándole al usuario datos que no son correctos.

Dificultad de mantenimiento: La dificultad en el mantenimiento del software conduce a que no se puede corregir un programa si en el se encuentra un error, si no se puede adaptar a los cambios del entorno, o mejorar si el cliente desea un cambio de requisitos.

Falta de integridad: Cuando el software adolece de seguridad, puede ser vulnerado en cualquiera de sus componentes: programas, datos y documentos, ocasionando pérdidas económicas y pérdida de la reserva de la información.

Dificultad en el uso: Esta causa obstáculos en el usuario en el manejo del sistema, el tiempo requerido para llegar a ser moderadamente eficiente en el sistema se hace mayor reduciendo su productividad, generando a los usuarios poca disposición hacia el sistema.

Dificultad de auditoria: Hay que determinar la facilidad con la que se puede comprobar el cumplimiento de los estándares de calidad control y legalidad.

Falta de flexibilidad: Es el esfuerzo requerido para transferir el programa y/o un entorno de sistemas a otro.

La no reusabilidad: El grado en que un programa (o partes de un programa) se pueden rehusar en otras aplicaciones. Esto va relacionado con el empaquetamiento o alcance de las funciones que realiza el programa.

La falta de fiabilidad: Es la probabilidad de operación libre de fallos (concordancia con los requisitos del software) de un programa en un entorno determinado y durante un tiempo específico.

Falta de seguridad: Produce que los datos y programas sean vulnerables.

La Falta de escalabilidad: Produce aspectos que no permiten aumentar la funcionalidad del software o actuación para satisfacer las nuevas necesidades.

1.1.5. Alternativas de solución

La alternativa para la evaluación de la calidad del software es:

Proponer un modelo guiado por la ISO/IEC 9126-1, que permita definir que características se deben tener en cuenta para evaluar la calidad del software de gestión como PRODUCTO terminado; donde se puedan establecer los valores de satisfacción o insatisfacción de cada característica y de el producto software en su totalidad, con el fin de servir de soportes a gerentes, ingenieros y administradores de áreas de sistemas informáticos de las empresas conocer el nivel de aceptación de la calidad al momento de adquirir software.

Para una solución confiable a este problema, se considera que la alternativa más óptima es Desarrollar un Modelo que permita definir que características se deben tener en cuenta para evaluar la calidad del software de gestión como PRODUCTO terminado. Es decir, aquellos factores de calidad del software (Corrección, mantenimiento, integridad, facilidad de uso, auditoria, flexibilidad, portabilidad, reusabilidad y fiabilidad y escalabilidad), con el fin de servir de soporte a los gerentes de empresas en la toma de decisiones en el momento de adquirir software.

Además, como se enunció en la descripción del problema, “que el tipo de software al cual se va a evaluar su calidad, será el software de gestión”; entre otros argumentos se tiene que dicho modelo facilita a los gerentes de las empresas la posibilidad de mejorar sus empresas, de acuerdo a sus objetivos estratégicos, a los mercados, a los objetivos de las mismas y a sus posibilidades.

Dicho Modelo evita que se produzcan costes financieros de repeticiones de trabajo, entre los cuales se tienen: costes de corrección de errores antes y después de instalar el software en producción, pérdidas de productividad debido a la falta de calidad del software y gastos innecesarios de mantenimiento y no lograr satisfacer al usuario.

Y en cuanto a que las empresas sean aquellas del sector público, hoy en día estas empresas tienen como finalidad lograr una certificación de calidad que les permita mejorar sus procesos organizacionales y aumentar su competitividad. Esto se desarrolla dentro de un contexto económico y empresarial donde el software que se adquiere o se desarrolla influya en dicha certificación.

Respecto de la importancia del problema, se debe tener en cuenta que el software juega un papel muy importante para el desarrollo de las organizaciones. Día tras día son liberados para su uso distintos tipos de programas para diferentes clases de clientes, los hay para cada necesidad de tal manera que resulta difícil imaginar alguna situación en la que el software no estuviera presente, dado que es uno de los componentes básicos de la tecnología que se involucra en las empresas, no sólo como soporte a los procesos de negocio, productivos y administrativos, sino como parte integral de las estrategias corporativas para la generación de ventajas competitivas.

2. OBJETIVOS DE LA INVESTIGACIÓN

2.1. OBJETIVO GENERAL

Proponer un Modelo para la evaluación de la calidad del Software de Gestión, a través de un proceso guiado por la ISO/IEC 9126-1:2001, y permita a los representantes de las empresas conocer un diagnóstico sobre las características de la calidad del software, para determinar la aceptación o rechazo del producto software.

2.2. OBJETIVOS ESPECIFICOS

- Diagnosticar y analizar la situación actual de la Calidad del software de Gestión desde el punto de vista de Software como producto terminado, por medio del estudio realizado a instituciones representativas en la colectividad empresarial
- Seleccionar los factores y características de calidad concernientes al software de gestión. De la siguiente manera:
 - Obtener las características de calidad asociadas al modelo de evaluación de calidad del software de gestión
 - Ordenar las Características por importancia
 - Elegir o definir los valores para cada Característica
 - Establecer los valores objetivo (finales o directos, intermedios o predictivos)
 - Establecer los valores mínimos aceptables en el modelo de evaluación.Con el fin de obtener una evaluación que permita determinar la calidad del software de gestión.

3. JUSTIFICACIÓN

- En el mundo cambiante actual, las empresas exigen agilidad en el manejo de la información de los procesos y aprueban inversiones en sistemas para obtener beneficios de negocio. Por lo tanto lo realizan no para obtener un nuevo sistema, sino para lograr beneficios que el sistema promete, estos beneficios pueden ser obtenidos mediante el aseguramiento de la calidad
- Con el desarrollo acelerado del software de Gestión, la variedad de Empresas que lo ofrecen, y la falta de un Modelo que permita evaluar la calidad de dicho software como producto terminado; se hace necesario contar con una Modelo que le permita a los representantes de las empresas e ingenieros de sistemas encargados de áreas de informática, conocer un diagnostico técnico sobre las características de la calidad del software, para determinar la aceptación o rechazo del producto software.
- El desarrollo de un modelo de evaluación de la calidad del software de Gestión permitirá un cumplimiento en las necesidades del usuario y satisfacción de los requerimientos ya que puede conocer la aceptación de cada una de las características y así determinar la aceptación o rechazo del producto software
- La mayor parte del software de gestión que se ofrece en el mercado no contempla los estándares de calidad ISO 9000, de la IEEE y de IEC, su diseño es regido comúnmente por criterios basados en la experiencia y la concepción personal de los desarrolladores de software.
- En la actualidad no existen modelos, sistemas que permitan evaluar la calidad del software de gestión, que generen resultados acerca de las características del mismo, lo cual dificulta obtener una visión clara con relación a la calidad del producto software y que este no brinde un beneficio económico de su uso.
- Los proyectos de desarrollo y adquisición de software en Colombia, como herramienta vital de competitividad en una empresa del sector público, requiere de una cuidadosa puesta en marcha, de un proceso ordenado y metódico que dé resultados favorables tanto para el contratista como para el contratante.
- El software como producto terminado es uno de los elementos de un sistema de información, es el resultado de una inversión y por ende, se espera un beneficio económico de su uso. Quien lo usa, por su parte tiene requerimientos que esperan sean satisfechos; es decir, tiene expectativas de calidad.

- El modelo de evaluación puede generar ahorros significativos tanto en los costos indirectos, como en los costos directos, porque las fallas en el sistema a causa del software serán menores y garantizarán mejores resultados.
- Estudios en Europa señalan que el costo de la NO-CALIDAD asciende en un 10% a un 20% de los costos operativos (mano de obra y materiales) según el rubro de la empresa. La introducción de sistemas de calidad ha llevado a reducir en el primer año este costo de un 6 a un 15% convergiendo asintóticamente de un 3 a 6% en el largo plazo. [7] [1]
- Aportar un modelo que servirá de soporte en futuras investigaciones en proyectos relacionados con la evaluación de la calidad del software.

4. MARCO DE REFERENCIA

4.1 ANTECEDENTES HISTORICOS

A. EL CONTROL DE CALIDAD: Fue desarrollado 40 años atrás en los Estados Unidos por un modesto especialista en el tema W. Edwards Deming.

Durante los años 50, Deming comenzó predicando que el trabajo con la calidad era la clave del éxito en los negocios, para lo cual desarrolló una metodología basada en información estadística que permitía determinar el comportamiento de los procesos, a la que denominó “Control estadístico de Calidad (CEC)”.

Cuando el Dr. Deming del Control Estadístico de Calidad asesoró al Japón entre 1949 y 1959 en la implementación de su teoría de que “si el Japón sigue asimilando y desarrollando en esta forma el concepto de Control de Calidad, dentro de varias décadas esta nación estará inundando con sus mercancías el mundo”, esta era una opinión de minoría; ni siquiera los mismos japoneses la creyeron.

B. Juran 1970, adecuación (del producto) al uso, adaptabilidad de Uso, esto implica dos parámetros: calidad de diseño y calidad de Conformidad, Es decir Adaptable a la necesidad de los usuarios.

C. Mc Call en 1977, Descompone el concepto de calidad en tres usos o capacidades importantes para un producto de software, así: operación, revisión y transición, modelo que será expuesto en el capítulo 4.4.5

D. Crosby 1979, conformidad de los requerimientos cero defectos, la calidad debe ser medible y predecible y con factores como ausencia de defectos, satisfacción al usuario y conformidad con los requerimientos.

E. ISO/IEC 9126 de 1991, cumplimiento de requisitos. Grado en el que un conjunto de características inherentes cumple con los requisitos

F. Dromey 1996, partiendo del modelo de Mc Call y de la ISO 9126 de 1991, plantea la calidad del producto por medio de la definición de subcaracterísticas que pueden ser medidas y evaluadas como características. Este modelo se expone en el capítulo 4.4.6

En el Departamento del Quindío en sus empresas, no existen en la actualidad proyectos relacionados con la evaluación de la calidad del software de gestión ⁸, pero nacional e internacionalmente, se han llevado a cabo estudios y proyectos dirigidos al control y evaluación de la calidad del software en su fase de diseño y desarrollo.

Los problemas en el software han sido estudiados, analizados y se le ha tratado de dar soluciones por más de 25 años. La famosa conferencia de finales de la década de los 60 sobre la “Crisis del Software” (que dio origen a la ingeniería de Software), fue tal vez el primer enfoque serio de un problema que en 1991 todavía seguía sin resolverse a pesar de ya existían algunos estándares para el software como producto final, fue así entonces como la IEEE siguió investigando y generando estándares que abarcaran tanto clasificación de estándares para anomalías de software IEEE 1044 – 1993, planes de seguridad IEEE 1228 - 1994, metodologías de métricas de software IEEE 1061 - 1998, entre otros adquiriendo una aproximaciones a la calidad. ([ver anexo No. 1](#)).

Desde entonces el garantizar su calidad se ha convertido en un precepto gerencial.

La aplicación de nuevas herramientas y técnicas han dado como resultado una mayor eficiencia en la producción y mantenimiento del software en un ambiente competitivo en el cual se incrementa el uso de las computadoras al mismo tiempo que el costo por unidad cae dramáticamente.

Herramientas Case: (Ingeniería de sistemas asistida por computador).

⁹ CASE es la aplicación de tecnología informática a las actividades, las técnicas y la metodología propias del desarrollo de sistemas. Las herramientas CASE son programas que automatizan o apoyan una o más fases del ciclo de vida del desarrollo de sistemas, su propósito es acelerar el proceso de desarrollo de sistemas y mejorar la calidad de los mismos.

La tecnología CASE se remonta a los primeros años de la década del 70. El proyecto ISDOS bajo la dirección del Doctor Daniel Teichrowe de la Universidad de Michigan, desarrolló un lenguaje llamado PSL para la descripción de problemas de usuario y las necesidades de solución de un sistema de información en un diccionario computarizado.

⁸ CONTRALORIA DEPARTAMENTAL DEL QUINDIO- DEPARTAMENTO DE SISTEMAS - 2005

⁹ Análisis y diseño de sistemas de información. Tercera Edición Jeffrey L. Whitten. Pag. 172.

Con el nombre de PSA se creó un producto asociado cuyo fin era analizar la relación de problemas y necesidades de una completa y persistente PSL/PSA funcionaban en grades computadoras que consumían valiosos recursos de máquina. Pocas compañías podían dedicar recursos a PSL/PSA.

El Autentico despegue de CASE llegó con la apropiación de las computadoras personales IBM. No mucho después en 1984 una compañía naciente llamada Index Technology creó una herramienta de software para PC denominada EXELERATOR. Sobre su enorme éxito se cimentó la tecnología CASE y hoy día se pueden encontrar en el mercado cientos de productos CASE suministrado por diferentes proveedores de sistemas.

Hoy día, casi todos los productos CASE trabajan en computadoras personales y estaciones de trabajo inteligente. En Algunos entornos, estas estaciones de trabajo comparten en red las herramientas CASE, las informaciones y los periféricos. Es importante advertir que la moderna tecnología CASE aun es muy joven, pero está perfeccionándose a gran velocidad. Cada mes surgen nuevos productos.

Actualmente existen algunas herramientas CASE dirigidas a la depuración de programas, al análisis y diseño de sistemas, implantación, soporte, gestión y estimación.

4.2 ESTADO DEL ARTE

A. Herramientas y modelos que contribuyen a mejorar la calidad del software.

Desde hace unos años, se ha generalizado el uso de analizadores estáticos de código como el LOGISCOPE desarrollado por la empresa francés VERILOG. Esta herramienta permite la medición automática de métricas que evalúan características del código fuente tales como la comprensibilidad, la facilidad de modificación, la estabilidad y la facilidad de realización de pruebas.¹⁰ Estas métricas son comparadas con los parámetros establecidos por la norma ISO 9126 definida por la ISO [10].

El realizar esta herramienta ha sido tradicionalmente difícil ya que los desarrolladores de aplicaciones han carecido de mecanismos de medición automática de métricas sobre la calidad del software y además hasta hace poco no había un parámetro estándar que evaluará estas métricas.

¹⁰ DUTSCH S. Maichael. Software Quality Engineering. PrenticeHall 1998

Despliegue de la Función de Calidad (DFC)¹¹

Es [12] una técnica de gestión de calidad que traduce las necesidades del cliente en requisitos técnicos de software. Originalmente desarrollado en Japón y usado por primera vez en Mitsubishi Heavy Industries Ltda. A principios de los años 70, DFC “se concentra en maximizar la satisfacción del cliente” para conseguirlo. DFC hace énfasis en entender lo que resulta valioso para el cliente y después desplegar estos valores a lo largo del proceso de Ingeniería. En realidad, el DFC se extiende a todo proceso de ingeniería.

SQA Suite [13]

Es el único producto totalmente integrado de la industria que permite efectuar pruebas automatizadas de aplicaciones cliente/servidor a nivel empresarial.

SQA Suite está compuesto por tres productos SQA líderes en el mercado, SQA Robot, SQA Manager, y SQA Load Test, ofreciendo la única solución integrada para realizar las pruebas automatizadas de las aplicaciones cliente/servidor en el ambiente multiplataforma Windows.

SQA Robot

Es el software de prueba de automatización más potente del mercado. Utiliza Object - Oriented Recording y Object Testing de SQA para la prueba de aplicaciones Windows a nivel del objeto. Posee integración con las herramientas de desarrollo cliente/servidor más populares usadas hoy en día, como: Power Builder, SQL Windows, Centura y Visual Basic. SQA Robot ofrece un entorno de programación integrado que permite a los usuarios grabar, programar y depurar scripts en un entorno integrado.

Cuando la primera versión de la aplicación ha sido terminada, los desarrolladores y/o el equipo de control de calidad utiliza el SQA Robot para crear y ejecutar las pruebas automatizadas que validarán la aplicación. Los resultados son guardados automáticamente dentro del repositorio de pruebas del SQA.

SQA Suite Object Testing™ es una tecnología que permite probar mucho más que el IGU de una aplicación llegando a todas las propiedades de los objetos que la componen. Esto es muy importante dentro de las herramientas de desarrollo que existen hoy en día – como Power Builder, Visual Basic, Oracle Developers 2000, SQL Windows, Centura y Delphi – donde la productividad ha sido dramáticamente aumentada a través del uso de objetos embebidos.

¹¹ Quality Assurance Resources Online. <http://w.w.w.quality.org/qc/homepage/html>

SQA Manager ¹⁴

[14] Basado en un repositorio de pruebas en red, SQA Manager ofrece funciones completas para la programación y desarrollo, información y análisis, así como gestión de errores (Defect Management). SQA Manager implementa un modelo activo de volumen de trabajo, notificando a los usuarios vía correo electrónico cualquier cambio en el estado de los ítems de SQA Test Repository.

Mientras los desarrolladores refinan las especificaciones funcionales, el equipo de control de calidad (QA), utiliza el SQA Manager para definir los requerimientos de prueba y control de calidad basados en la información obtenida a través del SQA Process.

SQA Manager analiza los resultados para identificar, dar prioridades y asignar los errores detectados a los programadores que serán los encargados de corregirlos. A medida que los errores detectados van pasando por los distintos miembros del equipo de pruebas y desarrollo, SQA Manager enviará mensajes vía e-mail automáticamente para notificar a cada uno de ellos sobre los cambios y estado de los errores que se van detectando. Cada nueva versión de la aplicación se vuelve a probar (pruebas de regresión), validándose así que los errores detectados hayan sido corregidos y que no se hayan incorporado nuevos errores durante este proceso.

Modelo de madurez de capacidad:

La ¹⁵ finalidad del modelo de madurez de capacidad, que ahora en adelante será nombrada como CMM, es ayudar a las organizaciones de software para que usen el CMM como guía para mejorar la madurez en el proceso del software.

Los fundamentos del CMM sirven para generar herramientas de diagnóstico las cuales son usadas para la evaluación del proceso del software y para la evaluación de la capacidad del software. La evaluación del proceso del software, consiste en tener un equipo de profesionales de software entrenados para determinar el estado del actual proceso del software en una organización, para

¹⁴ CARDENAS Tangarife José Joaquín. Calidad en Ingeniería del Software. Empresa Nacional de Telecomunicaciones Telecom. Santa fe de Bogotá 1998. "Se encargará de la elaboración de estrategia de pruebas, Plan de pruebas y ejecución de casos de pruebas".

¹⁵ An experiment measuring the effects of personal software process (PSP) training - Software Engineering, IEEE , Vol. 27, No. 5, May 2000.

- (Rational Unified Process) to Comply with CMM (Capability Maturity Model) Levels 2 and 3
Lisandra V. Manzoni and Roberto T. Price, Member, IEEE. IEEE Transactions on Software Engineering, Vol. 29, No. 2, February 2003

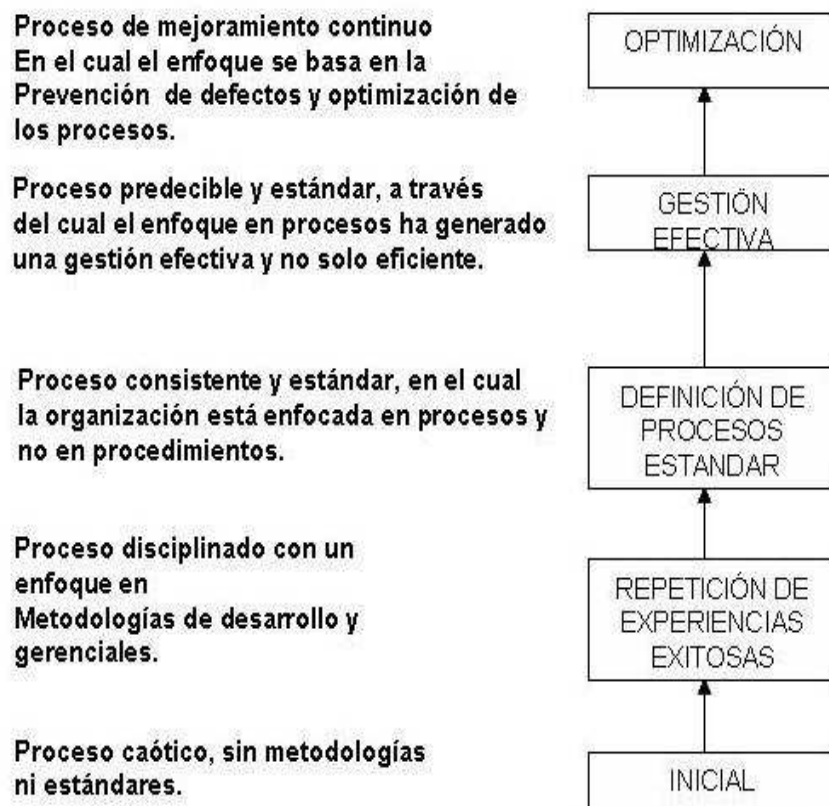
determinar la alta prioridad de las características del proceso del software en la organizacional en el mejoramiento del proceso de software.

La evaluación de la capacidad de software consiste en tener un equipo entrenado de profesionales para identificar contratistas que son calificados para desarrollar un trabajo de software o para monitorear el proceso de software usado.

Los cinco niveles de madurez definen una escala ordinal para medir la madurez del proceso de software en una organización y para evaluar su capacidad del proceso de software. Los niveles también ayudan a la organización a priorizar sus esfuerzos de mejoramiento.

Cada nivel de madurez, provee una capa en el proceso de mejoramiento continuo de la empresa. Cada nivel compromete una serie de metas, que cuando son satisfechas, estabilizan un importante componente del proceso de software. (Figura No. 1)

Figura 1. Niveles de Madurez de un proceso de Software



Proceso personal de software: ¹⁶

El proceso de software es la secuencia de pasos requeridos para desarrollar o mantener software. Un proceso de software adecuado es diseñado y presentado, cuando se da la definición de guías de ingeniería de software de cómo se trabaja. Un equipo que sigue diferentes procesos, o algo más común no usa procesos definidos forman lo que se llaman un pobre equipo, pues aunque tienen la intención y esfuerzo necesitado, no hablan con la misma base. En el lado opuesto, un equipo que sigue la misma definición de procesos, puede coordinar fácilmente el trabajo que es realizado individualmente por los miembros y precisa su proceso.

El PSP [15], es un proceso diseñado para ayudar a controlar, manejar, y aprobar la forma de trabajar.

A continuación se exponen los puntos al acercamiento del PSP: Figura No. 2

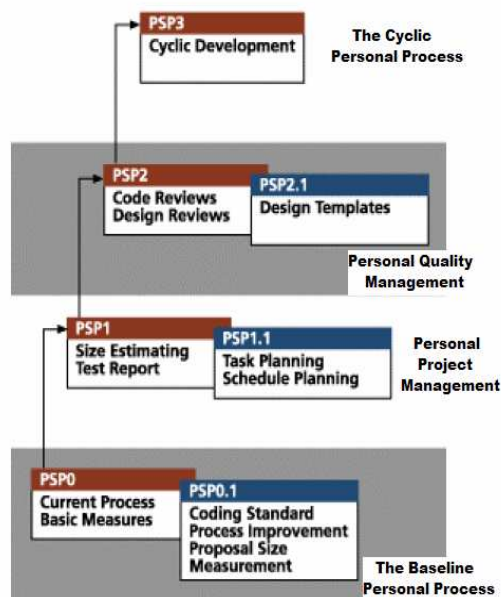


Figura 2 . PSP (Process Software Personal)

Tomado de Calidad del Producto. Grupo Alarcos ESI – UCLM. [31]

¹⁶ An experiment measuring the effects of personal software process (PSP) training - Software Engineering, IEEE , Vol. 27, No. 5, May 2000.

- (Rational Unified Process) to Comply with CMM (Capability Maturity Model) Levels 2 and 3
Lisandra V. Manzoni and Roberto T. Price, Member, IEEE. IEEE Transactions on Software Engineering, Vol. 29, No. 2, February 2003

PSP0 y PSP0.1 – Baseline Personal Process

El Baseline Personal Process (PSP0 – Current process, PSP0.1 - Coding Standard process improvement. Size measurement) provee una introducción a PSP y establece una base inicial de tamaño histórico, tiempo y datos de defectos. El paso inicial de PSP consiste en establecer una línea de base que incluye mediciones y un formato de informe. Provee una base para la medición del progreso y un fundamento definido a mejorar. PSP0 es el proceso actual que los Ingenieros utilizan para escribir software. PSP0 pasa a PSP0.1 por medio del agregado de código estándar, medición del tamaño y mejoramiento del proceso.

Los Ingenieros escriben 3 programas en este nivel y utilizan sus métodos actuales pero dentro de la estructura de los 6 pasos de la línea de base de procesos. Los pasos de la línea de base del PSP son:

- 1- Planeamiento – Planear el trabajo y documentar el plan
- 2- Diseño – Diseñar el programa
- 3- Codificación – Implementar el diseño
- 4- Compilación – Compilar el programa y determinar/eliminar todos los defectos Encontrados.
- 5- Prueba – Probar el programa y determinar / eliminar todos los defectos encontrados

PSP1 y PSP1.1 – Personal Project Management

PSP1 (Size estimating. Test report) y PSP1.1 (Task planning. Schedule planning) están enfocados respecto de las técnicas de administración de proyectos personales y hace una introducción a la estimación del tamaño y esfuerzo, planeamiento programado, y métodos de seguimiento de lo programado. Las estimaciones de tamaño y esfuerzo son realizadas usando el método PROBE (Proxy Based Estimating). A través de este método, los Ingenieros usan el tamaño relativo de un proxy para hacer una estimación inicial y utilizan los datos históricos para convertir el tamaño relativo del proxy a LOC (líneas de código). Los objetos (clase, función o procedimiento) son ejemplos de proxy. Utilizando PROBE, la estimación del tamaño es realizada por medio de la identificación de todos los objetos que deben ser desarrollados. Se determinan el tipo y el tamaño relativo de los objetos. El tipo hace referencia a la categoría general del componente (computacional, input / output, control logic, etc). Los rangos del tamaño relativo en el PSP son: very small, small, medium, large y very large. El tamaño relativo es convertido a LOC (líneas de código) usando una tabla de rango de tamaño basada en los datos de tamaño histórico del proxy. El tamaño estimado del nuevo código desarrollado es la suma de todos los objetos nuevos, además de las actualizaciones del código base existente. El tamaño y esfuerzo del tamaño son estimados usando el método estadístico de regresión lineal.

PSP2 y PSP2.1 – Personal Quality Management

PSP2 (Code reviews. Design reviews) y PSP2.1 (Design templates) agregan métodos de administración de la calidad a PSP: revisiones de código y diseño personal, notación de diseño, técnicas de verificación de diseño, y medidas para administrar la calidad del proceso y del producto.

El objetivo de la administración de la calidad en el PSP es encontrar y eliminar todos los defectos antes de la primera compilación. La medida asociada a este objetivo es el yield, el cual se define como el porcentaje de defectos encontrados antes de la compilación y que fueron eliminados antes de la compilación. Un yield de 100% ocurre cuando todos los defectos encontrados antes de la compilación son eliminados antes de la misma.

PSP3 – Cyclic Personal Process

Este proceso (PSP3 – Cycle development) está dirigido a la necesidad de escalar eficientemente sin sacrificar la calidad o productividad. La productividad se define respecto de un rango de tamaño (mínimo, máximo). PSP3 maneja este límite de escalabilidad por medio de una estrategia de desarrollo cíclico donde los grandes programas son particionados para el desarrollo e integración de los mismos. PSP3 introduce diseño de alto nivel, revisión de diseño de alto nivel, planeamiento cíclico y ciclos de desarrollo basados en el proceso PSP2.1. Usando PSP3, los Ingenieros particionan el proyecto en una serie de ciclos de PSP2.1 e integran y prueban la salida de cada ciclo. Debido a que los programas producidos con PSP2.1, son de alta calidad, los costos de integración y de prueba son menores.

PSP3 introduce métodos a utilizar cuando se desarrollan programas a gran escala. En PSP2, la propuesta es subdividir el proceso personal del desarrollo de grandes programas en partes de un tamaño determinado. Estos grandes programas son diseñados para ser desarrollados en pasos incrementales. La forma de hacer esto es a través de PSP3 usando el proceso de desarrollo cíclico. El proceso cíclico PSP3 puede ser un elemento efectivo para un proceso de desarrollo a gran escala solo si cada incremento de software es de alta calidad.

Quality Software ¹⁷:

Es un artículo, donde se hace un análisis del desarrollo del software, específicamente en la industria, como es la industria petrolera.

¹⁷ Journal Quality Software Ms Ron J West IEEE 29-35 , 1999

SOFTWARE QUALITY ASSURANCE PLANNING (IEEE) ¹⁸

Los acercamientos a la convicción de Calidad de Software buena practican en el apoyo de IEEE Std. 730 - 1989. Estas prácticas se dirigen hacia el desarrollo y mantenimiento de software crítico, eso está, dónde el fracaso pudiera dañar la seguridad o causa las pérdidas financieras grandes. Esta guía explica y clarifica los volúmenes de cada sección de un SQAP de que satisface los requisitos IEEE Std 730-1989. La guía reemplaza IEEE Std 983-1986 y no constituye los requisitos extensos que aquéllos declararon en IEEE Std 730-1989. Una organización puede exigir la complacencia con IEEE Std 730-1989 sin seguir esta guía completamente.

Aspectos metodológicos del desarrollo y reutilización de componentes de software:¹⁹

Los beneficios de la reutilización de componentes y activos de software han atraído la atención de un número considerable de empresas encargadas de desarrollar sistemas de software. Por este motivo, ha surgido recientemente un interés por crear y utilizar modelos de procesos para desarrollo de software que contemplen la reutilización. Aunque el concepto de reutilización no es nuevo, recientemente ha emergido, como elemento central de este, el concepto de componente de software reutilizable. Las características de los componentes de software reutilizables han obligado a crear nuevos modelos de desarrollo de software basados en la reutilización.

Estos procesos buscan la solución a un problema. Esta solución no es única, es por esto que los métodos deben ir evolucionando y adaptándose a las nuevas necesidades de los diferentes tipos de usuarios y los nuevos conceptos de la Ingeniería de Software.

Otra de las deficiencias encontradas en los métodos basados en la reutilización es que no especifican de forma detallada el ciclo de vida de un componente de software reutilizable. Estos métodos se centran en la reutilización del componente y no en su desarrollo individual.

Adquisición de software en la Universidad de los Andes – Guía metodológica ²⁰

¹⁸ Software Engineering Standards Collection”, 1991. IEEE.

¹⁹ Tesis de Grado de la Universidad de los Andes-Venezuela Elaborado por: Ing. Vanessa Hamar , Mérida de 2003.

²⁰ Adquisición de software en la Universidad de los Andes guía metodológica, tesis de grado, Adriana Gomez Muñoz, noviembre de 2002, Universidad del los Andes, Santa fe de Bogota.

En este trabajo, los autores abordan el problema de adquisición de software desde el punto de vista de Control organizacional y auditoria de sistemas. Para lo cual se realiza una revisión y evaluación de como las organizaciones hoy en día realizan el proceso de adquisición en general. Además, en dicho trabajo para definir la guía metodológica de adquisición de software se basan en la cadena del valor para reflejar las soluciones a las necesidades de la organización, por medio de preguntas como: "...Cada organización debe preguntarse así misma: ¿qué software necesitamos? La respuesta siempre será valiosa para asegurar que la compra y utilización de software sea eficiente y efectiva.

En la actualidad parte del valor agregado de una organización se ve representado en (el alto nivel de entrenamiento de) su personal, herramientas tecnológicas, sistemas e información.

Todas las mediciones, aunque dependen de las actividades, responsabilidad y desempeño del personal, también deben tener en cuenta el papel que juegan las herramientas tecnológicas en el día a día y su impacto. Cuando se habla de herramientas tecnológicas, se hace referencia a la capacidad y estado de los servidores y aditamentos físicos; infraestructura de los canales de comunicación utilizados y el software, constituido por todas aquellas aplicaciones desarrolladas o adquiridas por la organización para cumplir con las metas organizacionales.

Al hablar de software, se ingresa en un campo amplio en el cual actividades como: levantamiento de requerimientos, búsqueda en el mercado de soluciones, realización de términos de referencia, evaluación de propuestas, adquisición, implementación, capacitación, puesta en producción y seguimiento del software, son eslabones de una cadena. Algunas partes de la cadena pueden cambiar si se trata de adquirir un producto desarrollo a la medida y en casa, o hecho a la medida y fuera de casa, o una solución genérica y adaptable a la organización.

El problema de la adquisición de software y aplicaciones para la Universidad de Los Andes, radica en que cada unidad de la organización es autónoma para la compra de los mismos. Lo anterior genera y hace que surjan demoras e inconvenientes dentro del funcionamiento normal de la organización, ya que la infraestructura se puede ver afectada por la adquisición de herramientas sin el debido conocimiento de la misma; como ejemplo de esto tenemos el caso de la Facultad de Administración, la cual adquirió como plataforma de trabajo en grupo Exchange de Microsoft, sin tener en cuenta la seguridad de la Universidad, la red y hasta el momento no han logrado poner a funcionar en la totalidad la solución.

Surge entonces la pregunta: ¿Cómo se puede escoger correctamente un software que responda a las necesidades de la organización de forma armónica y siendo parte activa del engranaje que forma la infraestructura informática de la empresa?

Hay que revisar y evaluar los procesos y/o metodologías de adquisición de software que se utilizan, qué tan representativos son para las necesidades de la empresa, quiénes participan en el proceso de adquisición, cómo se evalúa el nuevo producto, etc.

Se debe tener en cuenta que la Universidad está compuesta por dos tipos de población en su estructura organizacional, a saber: Las dependencias administrativas (Recursos Humanos, Planta Física, DTI, etc), y las dependencias o departamentos académicos (Facultades y departamentos). Dentro de algunas dependencias académicas se encuentran centros de cómputo, los cuales influyen en la adquisición de software.

Aplicar los conocimientos adquiridos en la Especialización de Sistemas de Control Organizacional y de Gestión, para generar una solución viable al problema de adquisición de software para la Universidad de Los Andes. El problema se abordará desde el punto de vista de la Auditoria de Sistemas y del Control Organizacional.”

Usando Modelos de Calidad en la Selección de Paquetes de Software ²¹

Los autores de la Universidad Politécnica de Cataluña, España, Xavier Franch y Juan Pablo Carvallo, debido al crecimiento de componentes software ya desarrollados y de índole comercial, que se conocen como componentes COTS (commercial off-the-shelf), plantean metodología para el manejo de la construcción de la calidad partiendo de la ISO/IEC 9126-1. Donde se describen los factores de calidad de paquetes del software y declararan los requisitos al seleccionar un paquete del software en un contexto particular, a partir de 6 pasos y de esa clasificación expresar los requisitos de calidad y modelar las necesidades para obtener el paquete. Es importante informar que hay tipos diferentes de requisitos, como directivo, político, y, requisitos de calidad. Los requisitos de calidad son a menudo difíciles verificar. Esto es en parte debido a su naturaleza, como la falta de estructuras y descripciones extendidas de dominios del paquete (es decir, categorías de paquetes del software como los sistemas de ERP - de su traducción Planificación de Recursos Empresariales). Esta ausencia estorba la descripción exacta de paquetes del software y la declaración precisa de requisitos de calidad, y la selección del paquete consecuentemente global y confianza en el resultado del proceso. Nuestra metodología por construir la calidad estructurada planea que las ayudas resuelven esta desventaja.

Modelos de calidad del software de diseño a la arquitectura ²²

²¹ Traducción del artículo original “Using quality models in software package selection”, fuente Base de Datos IEEE UIS, Franch, X.; Carvallo, J.P. Software, IEEE Volume 20, Issue 1, Jan/Feb 2003 Page(s): 34 – 41.

La autora del artículo, presenta como objetivo principal del trabajo realizado “ El proponer una mejora del proceso de diseño arquitectónico mediante la aplicación de un enfoque basado en la norma ISO a la calidad, el enfoque SQUID de las especificaciones de calidad, la planificación, el control y la evaluación, y el modelado de El software de medición de apoyo a las actividades para asegurar la calidad del software. En este trabajo se utiliza principalmente para SQUID especificación de la calidad. Con el fin de aplicar nuestro enfoque, vamos a examinar varios métodos de diseño arquitectónico y de utilizar uno de ellos como un estudio de caso.

Un aspecto importante del diseño arquitectónico es que, por un lado, los requisitos de calidad de la gran influencia de la arquitectura del software. Por otra parte, los diversos requisitos tienen que ser "equilibrado" durante el proceso de diseño. Sólo recientemente la importancia de un diseño explícito de la arquitectura del software ha crecido considerablemente para la construcción de sistemas fiables de evolución.”

Características de la Calidad para la arquitectura de Software ²³

Es un trabajo que se refiere a la especificación de requisitos de calidad para la arquitectura del software, teniendo como técnica o basándose en la norma ISO 9126-1. Las autoras realizan una adaptación de la norma ISO 9126-1 para un modelo de calidad para la arquitectura del software. Además, se exponen “Los requisitos de calidad, capturados como requisitos no funcionales en las primeras etapas del desarrollo de software, en gran medida la influencia de la arquitectura de software de sistema. Sin embargo, también el sistema básico de abstracciones que son los requisitos funcionales, desempeñan un papel importante en la definición de la arquitectura inicial”

Atributos de Calidad para Componentes COTS ²⁴

Con la aparición de empresas desarrolladoras de software que ofrecen componentes COTS, se presenta la necesidad de evaluar dichos componentes con calidad para verificar que cumplen con los requisitos de calidad de los usuarios. Evaluación que se logra con una lista de atributos aplicables a esos

²² Traducción del título del artículo original “Quality Models to Design Software Architecture” fuente Base de Datos IEEE UIS Francisca Losavio, Central University of Venezuela, Caracas, Venezuela. Journal of Object Technology, Vol. 1, no. 4, September-October 2002

²³ Traducción del artículo original “Quality Characteristics for Software Architecture” Losavio Francisca, Chirinos ledy, Levy Nicole. Central University of Venezuela, Caracas, Venezuela. Université de Versailles St.-Quentin France. Journal of Object Technology, Vol. 2, No. 2, March-April 2003. Fuente http://www.jot.fm/issues/issue_2003_03/article2.pdf.

²⁴ Para conocer más sobre “Atributos de Calidad para Componentes COTS” presentado por Manuel F. Bertoa y Antonio Vallecillo, se consigue dicho artículo en www.lcc.uma.es/~av/Publicaciones/02/AtributosCalidadCOTS.pdf.

componentes COTS, que permiten medir varias propiedades funcionales. Los autores exponen que “ el modelo de calidad propuesto se contrasta también con los ya existentes, cuyo carácter demasiado generalista dificulta su implantación y utilización efectiva para el caso concreto de los componentes COTS.”, además los autores afirman que las características de calidad de los componentes se basan en la norma internacional ISO 9126.

Un Catálogo de Factores de Calidad para la Definición de Requisitos No-Técnicos en la Selección de Componentes COTS ²⁵

Los autores de la Universidad Politécnica de Cataluña, España, Xavier Franch y Juan Pablo Carvalho, presentan en el trabajo un catálogo como una estructura Jerárquica de características, subcaracterísticas y atributos análoga a la establecida en el estándar ISO/IEC 9126-1, proponiendo extender dicho estándar con factores no-técnicos siguiendo el marco del estandar, extensión que es la base para obtener modelos más exhaustivos en procesos concretos de selección ya que dispone de más de 180 factores en total. Finalmente, en el artículo también se muestra cómo generar y depurar requisitos no-técnicos partiendo del catálogo propuesto.

Un modelo de calidad basado para la descripción y evaluación de paquetes de software ²⁶

Los autores de la Universidad Politécnica de Cataluña, España, Xavier Franch y Juan Pablo Carvalho, definen que una selección de paquetes de software a partir de las necesidades de los usuarios es una tarea central en la ingeniería de software. Selección de paquetes inadecuado pueden comprometer los procesos de negocio y pueden interferir negativamente en el funcionamiento de la organización que participan. El éxito de la selección de paquetes está en peligro a causa de muchos factores, uno de los más importantes es la ausencia de descripciones estructuradas de ambas características del paquete de usuario y los requisitos de calidad. Por lo tanto, los autores en el documento proponen una metodología para describir los factores de calidad de paquetes de software utilizando el ISO/IEC 9126 como marco. A raíz de esta norma, los atributos de un determinado software de dominio y se identifican como una jerarquía estructurada, y las cifras son elegidos por ellos. Los paquetes de software en este ámbito puede ser descrito en un modelo uniforme y completa. Por lo tanto, la selección de los paquetes puede ser mejorada mediante la transformación de los requisitos

²⁵ El artículo “Un Catálogo de Factores de Calidad para la Definición de Requisitos No- Técnicos en la Selección de Componentes COTS, se puede consultar en http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER06/carvalho.pdf.

²⁶ Se traducción del título del artículo original “A Quality-Model-Based Approach for Describing and Evaluating Software Packages”. fuente Base de Datos IEEE de la UIS. Franch, X.; Carvalho, J.P. Universidad politecnica de Cataluña, Proceedings of the IEEE Joint International Conference on Requirements Engineering (RE'02) 1090-705X/02 \$17.00 © 2002 IEEE

de calidad de usuario en las necesidades expresadas en términos de la calidad del modelo de atributos. Ellos ilustran el enfoque presentado con un modelo de calidad para el servidor de correo de dominio.

El modelo de software de la calidad de los productos²⁷

El modelo de calidad de los productos de software se define y el autor formula por asociación un conjunto de propiedades de calidad de la ejecución de cada una de las formas estructurales que se utilizan para definir las declaraciones y la declaración de los componentes de un lenguaje de programación. Estas portadoras de propiedades de calidad son, a su vez vinculadas con el alto nivel de calidad de los atributos de la Norma Internacional de Evaluación de Productos Software ISO - 9126. El modelo se apoya en la creación de software de calidad, la definición de los idiomas específicos de codificación de las normas, la clasificación sistemática de defectos de calidad, y el desarrollo de código automatizado de los auditores para detectar defectos en el software.

Modelo de Agregación Basado en un Sistema Neurodifuso para un proceso de evaluación de calidad de software²⁸

Los autores establecen un sistema neurodifuso con los lineamientos generales a partir de las seis características establecidas por la ISO 9126, donde a partir de la especificación de los criterios de evaluación para cada atributo cuantificable se debe utilizar una función neurodifusa, con la finalidad de obtener un valor real, para lo cual es necesario definir por lógica difusa variables lingüísticas, valores lingüísticos, funciones de pertenencia y un dominio de valores discretos, esto permite que haciendo uso de la técnica de la lógica difusa se facilita la representación subjetiva de la calidad del software complementando de esta manera el proceso de cuantificación al modelo propuesto.

4.3 MARCO TEORICO

En primer lugar se tuvo en cuenta las asesorías y los apuntes de la Dra. Wilma Duarte, Dr. Julio Pedraza (docentes UIS) y asesoría del Licenciado Juan de Jesús Castillo (U Quindío), ya que a través de sus lineamientos y de sus conocimientos se establecen las pautas teóricas sobre metodología en el desarrollo del proyecto de investigación y se tomó como referencia las conferencias sobre metodología de la

²⁷ Traducción del título del artículo original "A MODEL FOR SOFTWARE PRODUCT QUALITY" fuente Base de Datos IEEE UIS, Dromey R.G., Software Quality Institute. Griffith University.

²⁸ Tomado de http://oomethod.dsic.upv.es/anonimo/..%5Cfiles%5CInConferenceArticle%5C2002_CLEI.pdf Condory – Fernandez Nelly, Belenguer F. Jorge, Albiol Manoli

Investigación suministradas por los docentes, porque en ellos está consignada la base teórica necesaria para dar cumplimiento a estos requisitos metodológicos. La teoría de la parte metodológica del proyecto está soportada sobre las asesorías y apuntes del Licenciado Juan de Jesús Castillo, Dr. Julio Pedraza y la Dra Wilma Duarte (UIS) con el espacio académico Seminario de la Investigación. [29] [30]

Los fundamentos teóricos se basan en libros como: análisis y diseño de sistemas de información, ingeniería del software, revista nacionales e internacionales, auditoria informática, Journals y documentos extractados de la red de Internet.

Sobre el libro ingeniería del software ³¹ se hizo referencia a los siguientes capítulos y páginas así: capítulo No. 8, control de calidad del software, paginas 121 a 123. "Bases teóricas sobre calidad, definida como una característica o atributo de algo, características mensurables de un programa.

Capitulo No-14, métodos de diseño, paginas 250 a 274, "Se obtuvieron los conocimientos básicos para realizar la fase de diseño como son: diseño arquitectónico que permitió desarrollar la arquitectura modular y representar las relaciones de control entre los módulos, diseño de interfaz el cual se refiere a la relación hombre máquina y el diseño procedimental el cual define los detalles de los procedimentales sin ambigüedades en un lenguaje natural".

Capítulo No-18, métricas técnicas del software, paginas 346 a 347. "Fueron tomados los conceptos básicos sobre los diferentes atributos que afectan la calidad del software, se concentran en tres aspectos importantes de un producto software: sus características operativas, su capacidad de cambios y su adaptabilidad a nuevos entornos."

De la revista ACIS ³², paginas 4 a 13. "El paradigma de la calidad, que trata de buscar no solamente su conceptualización sino los elementos para referenciar si estos conceptos se están llevando a la vida real en forma tangible, parametrizada y dinámica. El problema de la calidad aparentemente se ha convertido en un problema de medición."

De las paginas 49 a 55, "En el cual se formulan una serie de conclusiones y recomendaciones acerca de la calidad del software."

Del libro Auditoria informática ³³. "Se obtuvieron algunos conceptos básicos sobre las normas de auditoria informática desde el punto de vista del auditor tales como

³¹ Ingeniería del Software. Cuarta Edición. Roger S. Pressman. España.

³² Revista ACIS. Calidad del Software. Julio-Septiembre/97. Bogotá Colombia.

³³ Auditoria informática un enfoque operacional. José Dagoberto Pinilla. Editorial Ecoe. Santafé de Bogotá. 1992.

normas de auditoría operacional, procedimientos de auditoría informática, seguridad informática y la auditoría en el desarrollo de sistemas.”

De los documentos extraídos de la red Internet ³⁴. “Todos los temas sobre la calidad total en informática, y calidad del software. Se recurrió a este medio de información considerando que en la red se encuentra información actualizada y de buena calidad.”

4.4 MARCO CONCEPTUAL

4.4.1. Calidad: No es otra cosa que satisfacción del cliente. Un producto o servicio de calidad, es aquel que cumple las expectativas del cliente y que satisface sus necesidades con una relación costo/beneficio.

El término calidad en este documento siempre hará referencia a los atributos de la calidad software, es decir, el conjunto de cualidades que la caracterizan y que determinan su utilidad y su existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad, auditoría e integridad.

La no funcionalidad en el software, hace que el usuario no acepte el producto por no estar en capacidad de cumplir con los requerimientos exigidos.

La corrección se da cuando el usuario descubre defectos en el software, proporcionándole datos que no son correctos.

La dificultad de mantenimiento conduce a que no se puede corregir un programa si se encuentra un error en él, si no se puede adaptar a los cambios del entorno o ser mejorado si el cliente desea un cambio de requisitos.

La falta de integridad en el software se da cuando adolece de seguridad y puede ser vulnerado en cualquiera de sus componentes: programas, datos y documentos, ocasionando pérdidas económicas y de información.

La dificultad de auditoría, determina la facilidad con la que se puede comprobar el cumplimiento de los estándares de calidad, control y legalidad.

La falta de flexibilidad determina el esfuerzo requerido para modificar un programa operativo.

³⁴ <http://www.rspa.com>. <http://www.spc.ca/spc/metrvr.htm>, <http://www.pressman5.com>

La falta de portabilidad es el esfuerzo requerido para transferir el programa y/o un entorno de sistema a otro.

La reusabilidad indica el grado en que un programa o sus partes se pueden reusar en otras aplicaciones.

La escalabilidad, indica aquellos aspectos que permiten aumentar la funcionalidad del software o actuación para satisfacer las nuevas necesidades.

El estándar de calidad se refiere a la manera o proceso que debe ser seguido para construir y mantener el software. Por lo tanto enumeran y describen prácticas ingenieriles comúnmente aceptadas, que la experiencia ha demostrado hace más factible el desarrollo de productos de buena calidad de una manera consistente.

4.4.2 Métrica: * (medida) es un valor numérico o nominal asignado a *características* o *atributos* de un *ente* computado a partir de un conjunto de *datos* observables y consistentes con la intuición. Una métrica puede ser *directa* o *indirecta*, *interna* o *externa* –a veces, dentro de esta última categoría, se la suele llamar también objetiva o subjetiva.

4.4.3 Métrica Directa: ** es lo resultante de una correspondencia o mapeo directo entre un *atributo* de un *ente* (del dominio empírico) y el valor (del dominio numérico), y que sirve como referencia para describir y evaluar aspectos o situaciones del mundo empírico. Los atributos se miden generalmente por medio de métricas directas (aunque no siempre).

4.4.4 Métrica Indirecta: Es la resultante de una correspondencia o mapeo entre relaciones de dos o más *atributos*, o de un atributo compuesto (del dominio empírico) y el nuevo valor (del dominio numérico), o sea son cálculos entre dos o más métricas.

4.4.5 Modelos de calidad del software, Modelo factores/criterios/métricas de McCall^{35 36}

Descompone el concepto de calidad en tres usos o capacidades importantes para un producto de software, así: (Figura 3).

➤ Operación

* Tomado de http://eisc.univalle.edu.co/materias/Material_Desarrollo_Software/Metricas4.pdf

** Ibidem

³⁵ Calidad en Proyectos Informáticos - <http://lisisu02.usal.es/~mmoreno/APITema5.pdf>

³⁶ Análisis y diseño de aplicaciones informáticas de Gestión. Piattini G. Mario

- Revisión
- Transición

³⁷Cada capacidad se descompone en una serie de *factores* que determinan la calidad en cada una de ellas.

Tabla 1. Factores de Calidad, según Modelo de McCall
(Resumen del Modelo)

Operación:	Revisión	Transición
Facilidad de uso	Facilidad de prueba	Reusabilidad
Integridad	Facilidad de mantenimiento	Portabilidad
Eficiencia	Flexibilidad	Interoperabilidad
Corrección o exactitud		Modularidad
Fiabilidad		
Escalabilidad		

A su vez, en una serie de *critérios* o propiedades que determinan su calidad: Los criterios pueden ser evaluados mediante un conjunto de métricas para cada criterio deben fijarse unos valores máximo y mínimo aceptables.

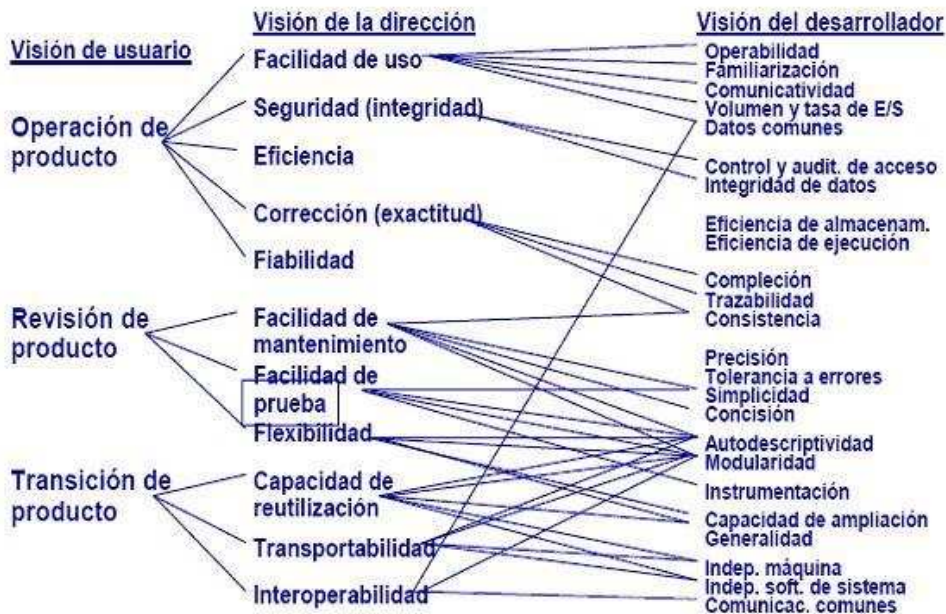


Figura 3. Visiones de los Factores de Calidad según el Modelo de McCall

³⁷ Calidad del producto.

4.4.6 Modelo de Dromey

³⁸El modelo de Dromey trabaja con una estructura que permite tener un modelo de calidad práctico para evaluar las etapas de Determinación de los requerimientos, diseño e implementación de los productos de software.

El Modelo de Dromey, plantea la calidad del producto por medio de la definición de subcaracterísticas que pueden ser medidas y evaluadas como características. Además que, permite entender la relación entre los atributos (características) y los sub atributos (subcaracterísticas) de calidad.

Dromey propone 3 modelos para cada etapa del proceso de desarrollo, a saber: (1) modelo de requerimientos, (2) modelo de diseño y (3) modelo de calidad de la implementación.

Las características de calidad planteadas en el Modelo de Dromey son: Eficiencia, Confiabilidad, Facilidad de mantenimiento, Portabilidad, Facilidad de uso y Funcionalidad que de una u otra manera tienen relación con lo planteado con las características de Modelo ISO/IEC 9126-1. Por lo tanto, estas características pueden ser agrupadas de acuerdo a diversos aspectos a tener en cuenta en la implementación: corrección, aspectos internos, aspectos del contexto y aspectos descriptivos. (Tabla 2)

ETAPA	PROPIEDADES DEL PRODUCTO	CARACTERISTICAS DE CALIDAD
Implementación	Corrección	Funcionalidad, Reutilización, Fiabilidad
	Aspectos Internos	Mantenibilidad, Reutilización, Eficacia
	Aspectos de Contexto	Mantenibilidad, Reutilización, Portabilidad, Fiabilidad
	Aspectos Descriptivos	Mantenibilidad, Reutilización, Portabilidad, Usabilidad

Tabla 2. Agrupamiento de las características de calidad según Dromey

Este modelo presenta una matriz propuesta por Dromey, donde relaciona las características de calidad respecto de la Norma ISO 9126-1. Dicha matriz presenta un mapeo entre las características del producto y los atributos de alto nivel, el cual es utilizado en las evaluaciones del producto de software (Figura 4).

³⁸ Estudio Comparativo de los modelos y estándares de Calidad del Software

	Propiedades de Exactitud							Propiedades de Estructurales							Propiedades Modulares					Propiedades Descriptivas						
	Calculable	Completo	Asignado	Preciso	Inicializado	Progresivo	Variable	Consistente	Estructurado	Resuelto	Homogéneo	Efectivo	No Redundante	Directo	Ajustable	Rango Independiente	Utilizado	Parametrizado	Acoplado	Encapsulado	Cohesivo	Genérico	Abstracto	Eficiente	Documentado	Descriptivo
Funcionalidad																										
Fiabilidad																										
Usabilidad																										
Eficacia																										
Mantenibilidad																										
Portabilidad																										
Reusabilidad																										

Figura 4. Matriz de Factores de Calidad según Dromey (traducción)

Los pasos para la aplicación del modelo de Dromey son: (1) Seleccionar el conjunto de atributos que se necesitan evaluar, (2) Realizar una lista de todos los componentes o módulos del sistema, (3) Identificar las propiedades de calidad de cada componente, (4) Determinar cómo afecta cada propiedad en los atributos de calidad y (5) Evaluar el modelo de calidad.

4.5. MARCO LEGAL

Respecto a las reglamentaciones vigentes para realizar un control de Calidad en el Software, se tienen:

Norma Colombiana 2-634³⁹: La cual facilita la comunicación internacional en el procedimiento de la información, presentando términos y definiciones de conceptos relevantes al campo de procesamiento de información e identificando relaciones entre las entradas, como también, la habilidad de un sistema de computador para operar correctamente aunque una o más de las partes de sus componentes no estén en funcionamiento.

Esta norma también presenta artículos como:

³⁹ Sistema de Control de Calidad en el Software – Icontec. Esta norma Colombiana editada por el Instituto Colombiano de normas Técnicas (ICONTEC) y es una homologación de la norma del International Organization For Standardization (ISO) 2-382/8 que trata sobre Control, Integridad y Seguridad

- a) Calidad de los datos.
- b) Prueba de operación, desempeño, aceptación.
- c) Seguimiento de fallas.
- d) Protección y seguridad

Norma Colombiana “Sistemas de Calidad” * Partiendo desde el punto de vista de software como un servicio que satisface las expectativas del cliente, ICONTEC elaboró una norma que abarca la estructura de la organización, responsabilidades, procesos y recursos para implementar la gestión de calidad en las operaciones de servicios, la cual está identificada bajo el número 1-0004.

Estándar ISO 9000: Las siglas ISO corresponden a la Organización Internacional de Normalización y proviene de la partícula griega “ISO” que significa “igual”. La ISO fue creada en 1947 y actualmente está constituida por más de 110 países. Se trata de una norma de ASEGURAMIENTO DE LA CALIDAD. Es decir, mediante la aplicación de esta norma pretende asegurar, frente a terceros, que se tiene la capacidad de mantener determinada calidad en los productos o servicios. Ha sido adoptada por los estándares nacionales de la mayoría de los países manufactureros alrededor del mundo. EL ISO 9000 consiste de cuatro partes: ISO 9001, ISO 9002, ISO 9003, ISO 9004.

- **EL ISO 9000:** Se Refiere a la administración de la calidad y estándares de aseguramiento de la calidad – guías para la selección y uso.
- **ISO 9001:** Hace referencia a sistemas de calidad – modelos para asegurar la calidad en diseño, desarrollo, producción instalación y servicio.
- **ISO 9003:** Sistemas de calidad. Modelos para asegurar la calidad en la inspección final y pruebas.
- **ISO 9004:** Elemento de la administración de la calidad y sistemas de calidad.
- **ISO/IEC 9126-1:2001 – Quality Model**

Esta parte de la ISO 9126 ⁴⁰ describe el modelo de calidad del producto de software. La primera parte del modelo especifica 6 características de calidad

* Icontec definió los sectores para poder normalizar, y entre ellos TECNOLOGÍA DE LA INFORMACIÓN. Generalidades, caracteres y codificación de la información, desarrollo de programas, interconexión de sistemas, redes, microprocesadores, terminales y otros equipos periféricos, interfaces, dispositivos de almacenamiento de datos, aplicaciones de información tecnológica

⁴⁰ ISO/IEC, ISO/IEC 9126-1:2001 Software Engineering – Product Quality, Part 1: Quality Model, Secretaría General de ISO, Ginebra, 2001

interna y externa, las cuales están divididas en subcaracterísticas, son manifestadas externamente cuando el software es utilizado como parte de un sistema, y son un resultado de atributos internos del software.

La calidad externa evalúa que el software satisfaga las necesidades del usuario teniendo en cuenta las condiciones especificadas.⁴¹ Esta calidad es medible en el comportamiento del producto. La calidad interna evalúa el total de atributos que un software debe satisfacer teniendo en cuenta condiciones especificadas. Esta calidad es medible a partir de las características intrínsecas.

Las características definidas son aplicables a todo tipo de software. Las características y subcaracterísticas proveen una terminología consistente respecto de la calidad del producto del software.

Esta Norma permite especificar y evaluar la calidad del software desde distintas perspectivas, las cuales están asociadas a la adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad, y auditoría del software. Puede ser usada por desarrolladores, evaluadores independientes y grupos de aseguramiento de la calidad responsable de especificar y evaluar la calidad del software.

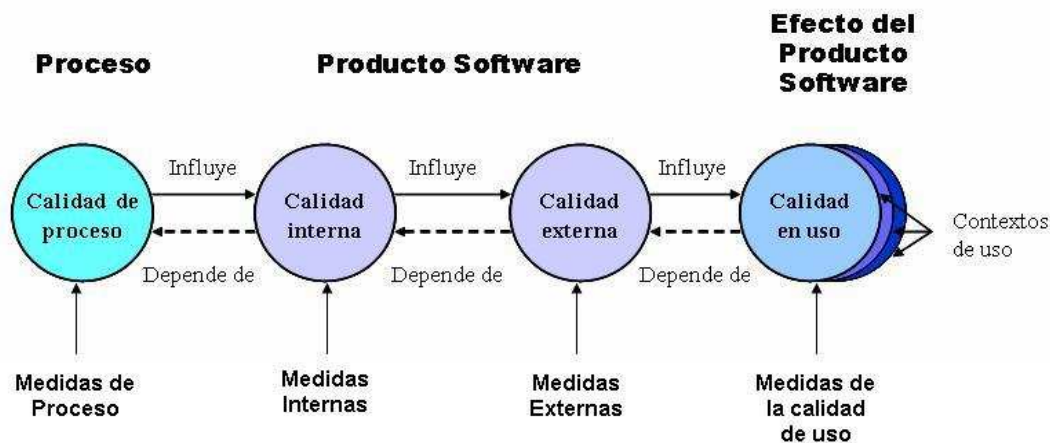


Figura 5. Calidad en el Ciclo de Vida según ISO/IEC 9126-1
Tomado y adaptado de Calidad del Producto. Grupo Alarcos ESI – UCLM. [37]

⁴¹ ISO/IEC, ISO/IEC 9126

* Ibidem

La evaluación de los productos de software que satisfacen las necesidades de calidad del software es uno de los procesos del ciclo de vida de desarrollo del software. La calidad del producto de software puede ser evaluada por medio de la medición de atributos internos, externos o a través de la calidad en uso (Figura 5). El objetivo es que el producto tenga el efecto requerido en un contexto particular de uso. La calidad del proceso contribuye a mejorar la calidad del producto, y la calidad del producto contribuye a mejorar la calidad en uso. Evaluar y mejorar la calidad de un proceso contribuye a mejorar la calidad del producto; y esto contribuye a mejorar la calidad en uso. De manera similar, evaluar la calidad en uso puede mejorar la calidad del producto, y evaluar un producto puede mejorar un proceso.

Las necesidades de calidad del usuario contribuyen a especificar los “requisitos de calidad externa”, los cuales contribuyen a especificar los “requisitos de calidad interna”. La “calidad interna” indica la existencia de “calidad externa” y ésta indica la existencia de “calidad en uso” (Figura 6).

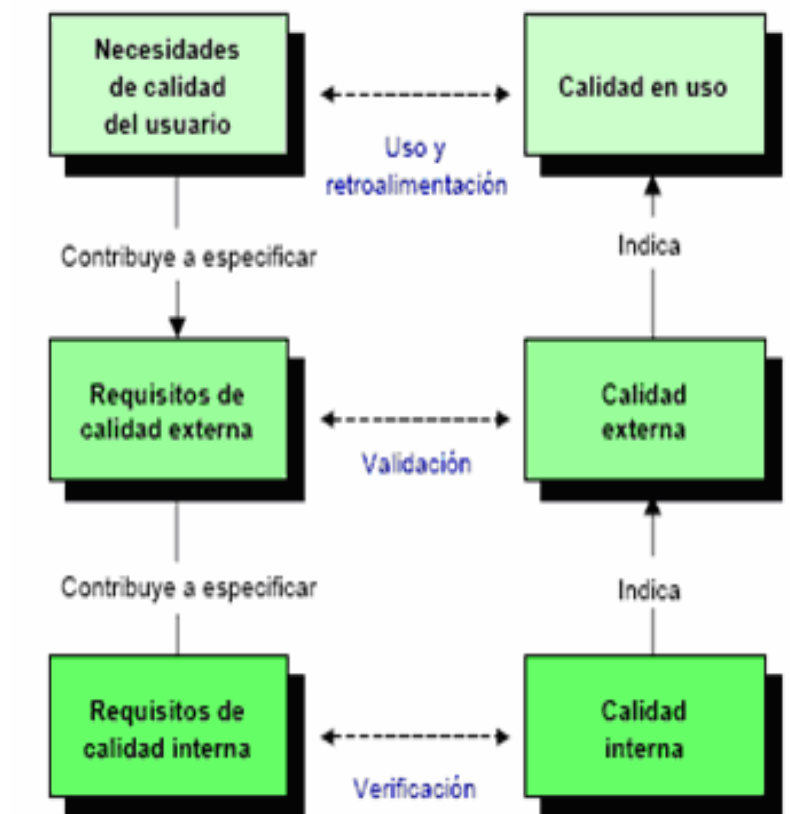


Figura 6. Relación Métricas del Modelo / Atributos en ISO/IEC 9126-1
Tomado de Calidad del Producto. Grupo Alarcos ESI – UCLM. [37]

El modelo de calidad de ISO 9126-1 establece 3 niveles: (1) Característica, (2) Subcaracterística y (3) Métricas. Existen métricas internas y externas. Las métricas internas pueden ser aplicadas a un software no ejecutable durante el diseño y la codificación. Las métricas externas se utilizan en el software ejecutable.

El modelo de calidad interna y externa está formado por las siguientes características: Funcionalidad, Confiabilidad, Facilidad de uso, Eficiencia, Facilidad de mantenimiento y Portabilidad (Figura 7)⁴²

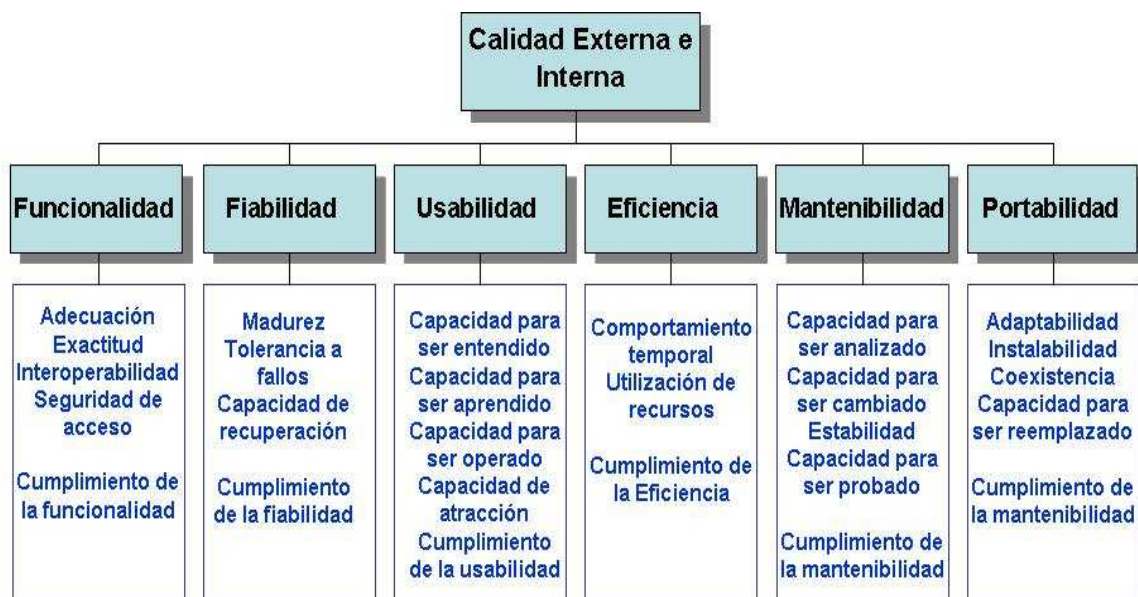


Figura 7. Modelo de Calidad Interna y Externa de ISO/IEC 9126-1

Tomado de Calidad del Producto. Grupo Alarcos ESI – UCLM. [37]

⁴² Calidad del Producto ISO/IEC 9126-1

5. ALCANCES Y LIMITACIONES

5.1 ALCANCES

Al realizar una investigación sobre los factores que determinan la calidad del software y las normas y procedimientos que la rigen, se pretende proponer un Modelo que presente las características de la calidad del Producto de software de Gestión.

El Modelo que se desarrollará solamente tomará la parte de la ISO 9126-1, que describe el modelo de calidad del producto de software en las características de calidad externa, donde evalúa que el software satisfaga las necesidades del usuario teniendo en cuenta las condiciones especificadas. Esta calidad es medible en el comportamiento del producto.

Lo anterior debido a que las características de dicha norma son los atributos que se desean evaluar en la Calidad del Producto de Software como son la Usabilidad, Fiabilidad, Corrección, Integridad, Mantenibilidad, Flexibilidad, Reusabilidad, Portabilidad, Escalabilidad y Seguridad; pero desde el punto vista del usuario y el Ingeniero de sistemas de la empresa que asesora a la alta dirección en lo referente al concepto técnico que se realiza antes de la toma de decisión para la comprar o adquisición del software de gestión.

5.2. LIMITACIONES

El modelo está limitado a no tener las características Internas de la Calidad del Producto de Software, ya que estas características miden el software en sí mismo y puede ser aplicada a un software no ejecutable (como una especificación o código fuente) durante el diseño, la codificación e implementación, como las sentencias de código de fuente, control de gráficos, flujo de datos y estados de representación de procesos.

El modelo de evaluación de Calidad de Software, en esta primera fase y/o propuesta, está centrado solamente para evaluar el software Contable, tomándose para ejemplarizar el modelo una empresa del Sector público; donde se evaluará los atributos de calidad del producto de software de este tipo de software de gestión en una forma de cuestionario tanto al ingeniero de sistemas y usuario final en referencia con los atributos expuestos en el estándar Norma ISO 9126-1, dichas personas responderán en forma objetiva sobre el modelo, y el modelo se encargará de definir una calificación a los atributos de calidad, que le permiten a la

empresa conocer un diagnóstico técnico de sus características de calidad que apoye una decisión de aceptación o rechazo del software que se está evaluando.

Debido a la gran diversidad de Software Contable en lo referente a códigos, arquitecturas, plataformas, interfaces, el modelo no evaluará el software de gestión extrayendo de ellos información en forma automática que alimenten el modelo para permitir que este evalúe sus atributos. Por lo tanto, el modelo presentará como entrada las características a los diferentes atributos de calidad de Producto de software que se deben tener en cuenta para que un software contable sea de calidad.

Se consideran diferentes módulos del modelo, pero solamente se tendrá en cuenta el módulo de evaluación del ingeniero (evaluación atributos de operación). De esta forma, se presentará una visión general de la operatividad del Modelo de evaluación de la calidad del software de gestión, específicamente en el software contable. Así mismo, se deja abierta la posibilidad de que se le agreguen otros módulos que evalúen otros tipos de software.

6. FORMULACIÓN DEL PROBLEMA

¿Existen Modelos que permitan evaluar las diferentes características que afectan la calidad del software de gestión?

7. HIPOTESIS

Las diferentes características que afectan la calidad del software de gestión, se pueden evaluar a través de un Modelo que permita medir cada uno de las características de la calidad del software de gestión como producto terminado.

7.1. VARIABLES

Cada variable tendrá un Índice, el cual es un valor de preferencia dado en los niveles de Satisfactorio e insatisfactorio, que para el modelo será de Aceptado o rechazado respectivamente pero estos niveles se definen así:

Insatisfactorio estará comprendido de 0% a 40%, Marginal estará comprendido desde 40% a 60 % y Satisfactorio estará comprendido desde 60 a 100%. Lo anterior con el fin de dar cumplimiento a la escala satisfacción que se usará para evaluar cada componente de las subcaracterísticas (escala de likert que se expone en el capítulo 10.4) ya que dicha escala provee un nivel de Aceptación (totalmente de acuerdo) y nivel neutro (marginal) y un nivel de rechazado (totalmente en desacuerdo) .

7.1.2. Definición nominal ** y operacional, indicadores de las variables

a. Revisión: Se define nominalmente como la acción de revisar y/o observar detalladamente un objeto.

* Es el esquema de categorías de aceptabilidad que se adopta con respecto al estándar IEEE 1061, donde indica que los valores para clasificar el software se da en categorías de “aceptable”, “marginal” e “inaceptable”, y los niveles de satisfacción o niveles de aceptabilidad según la ISO-9126 son “satisfactorio” e Insatisfactorio”

** Definición Nominal es el significado del término o la descripción del concepto. La definición Operacional, es el significado del término aplicado a algo específico, para el caso, con relación al software. Hernández Sampieri, R.; Fernández Collado, C.; Baptista Lucio, P. *Metodología de la Investigación*. Segunda Edición. Santa Fé de Bogotá, Colombia. Mc Graw Hill. 1998.

VARIABLES

Mantenimiento:

- Def. Nominal: Efecto de mantener o mantenerse, lleva a cabo su función mediante revisiones periódicas que permiten identificar aquellos fallos que pueden originar averías.
- Def. Operacional: Facilidad de mantenimiento: Es la facilidad con que una modificación puede ser realizada. La facilidad para localizar y arreglar un programa si se encuentra un error, se puede adaptar si su entorno cambia, o mejorar si el cliente desea un cambio de requisitos.
- Indicador: Valor numérico representado porcentualmente, indicando el grado de calidad de este atributo.

Flexibilidad:

- Def. Nominal: Propiedad de los cuerpos en virtud de la cual se curvan mientras dura la acción de una fuerza exterior aplicada a ellos. Que cede o se acomoda fácilmente al dictamen de otro. Propiedad de un cuerpo que permita ser deformado y doblado sin que llegue a romperse.
- Def. Operacional: Es la facilidad para modificar un programa, adaptándolo para realizar tareas específicas. Además que permita su fácil aprendizaje operativo.
- Indicador: Valor numérico representado porcentualmente, indicando el grado de calidad de este atributo

b. Operación: Se define nominalmente como la ejecución de una cosa.

Operación: Se define operacionalmente como el Factor de calidad del software que se concentra en las características operativas del mismo.

VARIABLES.

Dificultad en el uso.

- Def. Nominal: Complicación al ejercitar o practicar una cosa.
- Def. Operacional: Es la dificultad para aprender, operar, preparar los datos de entrada e interpretar las salidas de un programa. Es también, un intento de cuantificar lo amigable que puede ser con el usuario el sistema, reduciendo su productividad y generando a los usuarios poca disposición hacia el sistema.
- Indicador: Valor numérico representado porcentualmente, indicando el grado de calidad de este atributo.

- Índice: Es el valor de preferencia será uno de los tres niveles de aceptabilidad, esto es, Insatisfactorio (0 a 40%), marginal (desde 40 a 60%), y satisfactorio (desde 60 a 100%)

Insatisfactorio (0 a 40%): El software presenta dificultad de interacción con el usuario, no presenta mensajes de error significativos, no utiliza en forma eficiente el espacio disponible en la pantalla, la información que muestra no es apropiada al contexto además no presenta un buen diseño para la entrada de datos.

Marginal (desde 40 a 60%): El software requiere más herramientas que eviten al usuario memorizar información, proporciona algunas ayudas que facilitan la consulta. Su formato aun es inconsistente. Solo algunas veces presenta mensajes de error significativo y en algunas interfaces hace uso de una buena distribución de la pantalla. En variadas ocasiones no minimiza el número de acciones de entrada de datos.

Satisfactorio (desde 60 a 100%): El software presenta buenas herramientas de interacción general como ayudas sensibles al contexto, un formato consistente, buena interfaz de usuario, buena visualización de la información y hace uso de un adecuado diseño para la entrada de datos.

Integridad.

- Def. Nominal: Aquello en que no falta ninguna de sus partes. De una perfecta probidad incorruptible.

- Def. Operacional: se refiere a la seguridad de que una información no ha sido alterada, borrada, reordenada, copiada, etc., bien durante el proceso de transmisión o en su propio equipo de origen

- Indicador: Valor numérico representado porcentualmente, indicando el grado de calidad de este atributo.

0% Insatisfactorio: El software no presenta ningún proceso de verificación y autenticación de usuarios y de identidad de usuarios.

100% satisfactorio: El software presenta procesos de verificación y autenticación de usuarios y de identidad de usuarios.

Corrección.

- Def. Nominal: Acción o efecto de corregir. Alteración hecha en una obra para mejorarla.

- Def. Operacional: Especifica hasta donde satisface un programa su especificación y logra los objetivos propuestos por el cliente.

- Indicador: Valor numérico representado porcentual mente, indicando el grado de calidad de este atributo.

- Índice: Es el valor de preferencia será uno de los tres niveles de aceptabilidad, esto es, Insatisfactorio (0 a 40%), marginal (desde 40 a 60%), y satisfactorio (desde 60 a 100%)

Insatisfactorio (0 a 40%): El software no satisface los requerimientos del usuario. Se debe realizar nuevamente la fase de análisis.

Marginal (desde 40 a 60%): Se debe realizar una nueva revisión a la fase de análisis debido a que no ha cumplido con la totalidad de los requerimientos.

Satisfactorio (desde 60 a 100%): El software cumple con los requerimientos de funcionamiento.

Fiabilidad.

- Def. Nominal: Es la habilidad de una persona o sistema para realizar y mantener sus funciones en las circunstancias normales, así como en las circunstancias hostiles o inesperadas.

- Def. Operacional: Fiabilidad: Determina hasta donde se puede esperar que un programa lleve a cabo su función pretendida con la exactitud requerida, la probabilidad de operación libre de fallos de un programa en un entorno determinado y durante un tiempo específico.

- Indicador: Valor numérico representado porcentualmente, indicando el grado de calidad de este atributo.

Insatisfactorio (0 a 40%): El software arroja datos que no son confiables y presenta una alta frecuencia de inconsistencias . .

Marginal (desde 40 a 60%): El software con mucha frecuencia arroja datos que no son confiables presentando un nivel grave de inconsistencias.

Satisfactorio (desde 60 a 100%): El software arroja datos confiables.

Seguridad.

- Def. Nominal: Calidad de seguro, certeza, confianza. Aquel o Aquello en que se puede confiar en absoluto.

- Def. Operacional: Es la disponibilidad de mecanismos (físicos y lógicos) que controlan o protegen los programas o los datos de riesgos potenciales que pueden producir un impacto negativo en el software y hacer que falle el sistema.

- Indicador: Valor numérico representado porcentualmente, indicando el grado de calidad de este atributo.

Insatisfactorio (0 a 40%): El software no cuenta con procedimientos de control que protejan el equipo, los programas y los datos de riesgos potenciales que incidan en el funcionamiento del software .

Marginal (desde 40 a 60%): El software cuenta con algunos procedimientos de control que protegen los programas y los datos de riesgos potenciales que incidan en el funcionamiento del software, como controles de entrada, proceso y salida.

Satisfactorio (desde 60 a 100%): El software cuenta con procedimientos de control para acceso a los datos y que protegen los programas y los datos de riesgos que incidan en el funcionamiento del software, como controles de entrada, proceso y salida. Además, el sistema Cuenta con controles de operación, documentación.

c. Transición: Se define nominalmente como la acción y efecto de pasar de un estado a otro. Modo de pasar de un razonamiento a otro.

Transición: Se define * operacionalmente como una capacidad de la calidad del software que se concentra en la capacidad de reusabilidad, portabilidad e interoperabilidad de nuevos entornos

VARIABLES

Portabilidad.

- Def. Nominal: facilidad de un objeto de ser trasladado de un sitio a otro.

- Def. Operacional: Es la facilidad para transferir el programa de un entorno de sistema hardware y/o software a otro.

- Indicador: Valor numérico representado porcentualmente, indicando el grado de calidad de este atributo.

Insatisfactorio (0 a 40%): El producto software no tiene la capacidad para ser adaptado e instalado en diferentes entornos especificados, como no coexistir con otro software para compartir recursos comunes.

Marginal (desde 40 a 60%): El software tiene la capacidad para ser adaptado o instalado en diferentes entornos especificados o de coexistir con otro software para compartir recursos comunes.

* Modelo de Mc Call (1979). PIATTINI G. Mario, Calvo José A. Análisis y Diseño de Aplicaciones Informáticas de Gestión.

Satisfactorio (desde 60 a 100%): El software tiene la capacidad para ser adaptado e instalado en diferentes entornos especificados, como la de coexistir con otro software para compartir recursos comunes

Reusabilidad.

- Def. Nominal: característica de un objeto o sistema que le permite ser usado varias veces.
- Def. Operacional: Determina hasta donde se puede emplear un programa o partes de él en otras aplicaciones, con relación al empaquetamiento o alcance de las funciones que realiza el programa.
- Indicador: Valor numérico representado porcentualmente, indicando el grado de calidad de este atributo.

Insatisfactorio (0 a 40%): No permite utilizarse el software en otra aplicación

Marginal (desde 40 a 60%): Permite utilizar alguna parte del software en otra aplicación, pero con restricciones en sus funciones.

Satisfactorio (desde 60 a 100%): Permite reutilizar el producto software en otra aplicación sin ningún tipo de restricción en su funcionamiento.

Escalabilidad

- Def. Nominal: característica de un objeto o sistema que le permite ser modificado aumentando sus servicios y operaciones.
- Def. Operacional: Son los aspectos adaptables de mantenimiento del software, es decir, mientras se aumenta la funcionalidad del software o actuación para satisfacer las nuevas necesidades agregando una interfaz o hardware haciéndolo fácil de modificarlo para agregar las nuevas capacidades.
- Indicador: Valor numérico representado porcentualmente, indicando el grado de calidad de este atributo.

Insatisfactorio (0 a 40%): No permite ser actualizado para poder procesar más transacciones por medio de nuevos procesadores, dispositivos y almacenamiento que se pueden implementar fácil y transparentemente

Marginal (desde 40 a 60%): Permite realizar algunas actualizaciones para procesar algunas transacciones pero pierde la calidad en sus servicios

Satisfactorio (desde 60 a 100%). Permite ser actualizado para poder procesar más transacciones por medio de nuevos procesadores, dispositivos y almacenamiento que se pueden implementar fácil y transparentemente o permite hacerse mas grande sin perder calidad en sus servicios

8. METODOLOGIA y TECNICAS

8.1. TIPO DE ESTUDIO

8.1.1. Exploratorio

El tipo de investigación empleado en este estudio es Exploratorio⁴³, porque suministra conocimientos parciales o generales los cuales permiten obtener una solución aproximada del problema.

En la fase de análisis y pruebas se pretendió obtener una visión general de tipo aproximativo respecto a una determinada realidad y dado que el tema de investigación había sido poco explorado y aun sobre él era difícil formular hipótesis precisas o de cierta generalidad. Además se trata sobre el estudio de un fenómeno novedoso del cual no se disponen de suficientes recursos para emprender un trabajo más profundo.

8.1.2. Descriptivo

También es un tipo de investigación descriptiva, porque se preocupa por suministrar descripciones, en lo posible completas y definidas sobre la realidad en la deficiencia de la calidad del software y sus características.

8.2. METODOS USADOS

8.2.1. Modelo empleado para la Evaluación

El modelo usado para la evaluación de calidad será guiado con base en la ISO 9126-1^{*}, estándar que permite evaluar la calidad del producto de software [40]. Donde La calidad del producto será la Calidad externa, la cual evalúa que el software satisfaga las necesidades del usuario teniendo en cuenta las condiciones especificadas. Esta calidad es medible en el comportamiento del producto.

El desarrollo de la fase de análisis, se realiza utilizando el Modelo de Programación Orientado a Objetos⁴⁴, el cual se centra en el proceso de reunión de requisitos para comprender la naturaleza del modelo.

⁴³ Sabino A. carlos. El proceso de la Investigación Edit. El Cid. Bogotá 1976 Pag 60 a 62

* Objetivo General planteado

⁴⁴ Braude E. Ingeniería de Software, Una Perspectiva orientada a Objetos

Pero, ¿qué es un modelo? La respuesta es bien sencilla, **un modelo es una simplificación de la realidad**. El modelo proporciona los planos tanto del análisis como del diseño ya que este puede estar entre todo el ciclo de vida del software, proporcionando una visión general del Modelo, ya que con ellos se consiguen cuatro objetivos:

- Los modelos ayudan a visualizar cómo es o queremos que sea un prototipo.
- Los modelos permiten especificar la estructura o el comportamiento de un prototipo.
- Los modelos proporcionan plantillas que nos guían en la construcción de un prototipo.

Por lo anterior y con el fin de generar el Modelo de Evaluación de Calidad del Software de Gestión, este se trabajara con base en UML, que proporciona las herramientas necesarias para poder obtener los planos del software y abarca todas las fases del ciclo de vida de un proyecto, soporta diferentes maneras de visualización dependiendo de quién tenga que interpretar los planos y en que fase del proyecto se encuentre.

Además que la orientación a objetos ofrece ventajas como:

- *Mantenibilidad* (facilidad de mantenimiento). Los programas que se diseñan utilizando el concepto de orientación a objetos son más fáciles de leer y comprender y el control de la complejidad del programa se consigue gracias a la ocultación de la información que permite dejar visibles sólo los detalles más relevantes.
- *Modificabilidad* (facilidad para modificar los programas). Se pueden realizar añadidos o supresiones a programas simplemente añadiendo, suprimiendo o modificando objetos.
- *Reusabilidad*. Los objetos, si han sido correctamente diseñados, se pueden usar numerosas veces y en distintos proyectos.
- *Fiabilidad*. Los programas orientados a objetos suelen ser más fiables ya que se basan en el uso de objetos ya definidos que están ampliamente testados.

Donde los cuatro aspectos expresados anteriormente son parte de los factores que se van a evaluar en la investigación.

8.2.2. Inductivo

En la fase de análisis se definen las diferentes características que determinan la calidad del software de gestión, los cuales pueden ser aplicables en la evaluación de la calidad de todo tipo de software. De esta forma se ha hecho una generalización o inducción tendiente a darle el carácter de ley general.

8.2.3. Deductivo.

La calidad del software como característica, está presente en cualquier tipo de software, y esta a su vez esta determinada por los diferentes factores que la componen.

8.2.4. Analítico

Durante la determinación del problema de investigación sobre la calidad del software se consideró como ámbito general todos los tipos de software existentes en el medio, lo cual haría del proyecto algo complejo; para evitarse ésta complejidad se delimito el proyecto en el Departamento del Quindío definiéndose orientar el proceso de investigación a Evaluar la Calidad del Software de Gestión.

Además, es analítico en la definición de los diferentes valores que determinan en forma cuantitativa los atributos de la calidad del software. Como también en el momento de definir la población y las técnicas de recolección de información con el fin de delimitar el problema de investigación.

8.2.5. Síntesis

Los usuarios nunca deben recomendar la compra de paquetes de aplicaciones sin antes usar el producto, evaluar su documentación, considerar los aspectos de formación y valorar los costos (de adquisición y mantenimiento).

Al Formular un Modelo de evaluación de la calidad del software de gestión, permitirá a las empresas contar con una Modelo para determinar la aceptación y/o satisfacción de las características antes de tomar la decisión de adquirir un software de buena calidad.

Esta investigación esta orientada únicamente a la evaluación de la calidad del software de gestión y queda como propuesta para su refinamiento y proyección a otros tipos de software.

8.3. TECNICAS

8.3.1. Recolección de información

8.3.1.1. Encuesta a Ingenieros de Sistemas

Para la recolección de la información, se empleará el método de encuesta dirigida y estructurada a ingenieros, con el objetivo de obtener información específica sobre la calidad del software de gestión.

Se determinó realizar una encuesta ya que permite obtener información de un grupo socialmente significativo de personas acerca del problema en estudio para luego, mediante un análisis de tipo cuantitativo, sacar las conclusiones que correspondan con los recogidos.

Las encuestas se realizarán a 15 Ingenieros de Sistemas que permanentemente interactúan y desarrollan sistemas de información, a través de cuestionarios con formato fijo, los cuales contienen preguntas con respuestas específicas por parte de las personas encuestadas. El cuestionario consta de preguntas con elecciones múltiples y preguntas con respuestas de opinión.

8.3.1.2. Entrevista a usuarios finales

La entrevista utilizada es del tipo no estructurada informal de gran utilidad en estudios exploratorios y recomendable cuando se trata de abordar realidades poco conocidas por los investigadores, en donde se reduce a una simple conversación sobre el tema de estudio, dando completa sensación al entrevistado de que puede hablar libremente, alentándolo y estimulándolo cautamente, para evitar influirlo con nuestras actitudes.

Se realizaron entrevistas de tipo informal⁴⁵ a 20 usuarios casuales (que utilizan solo el sistema de manera ocasional), y 5 usuarios dedicados (aquellos invierten una cantidad de tiempo considerable en la utilización de determinados programas) a los cuales se les solicitara la opinión, acerca de los factores que tienen más incidencia, sobre la calidad del software de gestión.

8.3.2. De Campo

8.3.2.1. Diseño de muestreo

⁴⁵ Jeferey I. Nhitten. Análisis y Diseño de Sistemas de Información. Editorial Irwin. España 1996. Pag .707

Esta conformada de una parte por todas aquellas personas involucradas en el problema que pueden proporcionar información relativa al análisis y de otra, por aquellas personal que directamente resultaran beneficiadas con el desarrollo de esta investigación.

La población a tener en cuenta para esta investigación en el municipio de Armenia Quindío es:

- Usuarios finales
- Ingenieros de Sistemas con experiencia en sistemas de información.

Muestra: Debido a que el tamaño de la población que sirve como fuente esta representada por 25 (veinticinco) empresas que poseen un área de sistemas (ya sea esta división, departamento o sección dentro del organigrama de la empresa) se ha decidido utilizar una muestra representativa del 60% (15 empresas).

8.3.2.2. Ilustración del Modelo

El modelo de evaluación de calidad se ejemplifica con un caso de estudio, donde se da a conocer la aplicabilidad del modelo usando las características definidas, los pasos propuestos para la evaluación de la calidad.

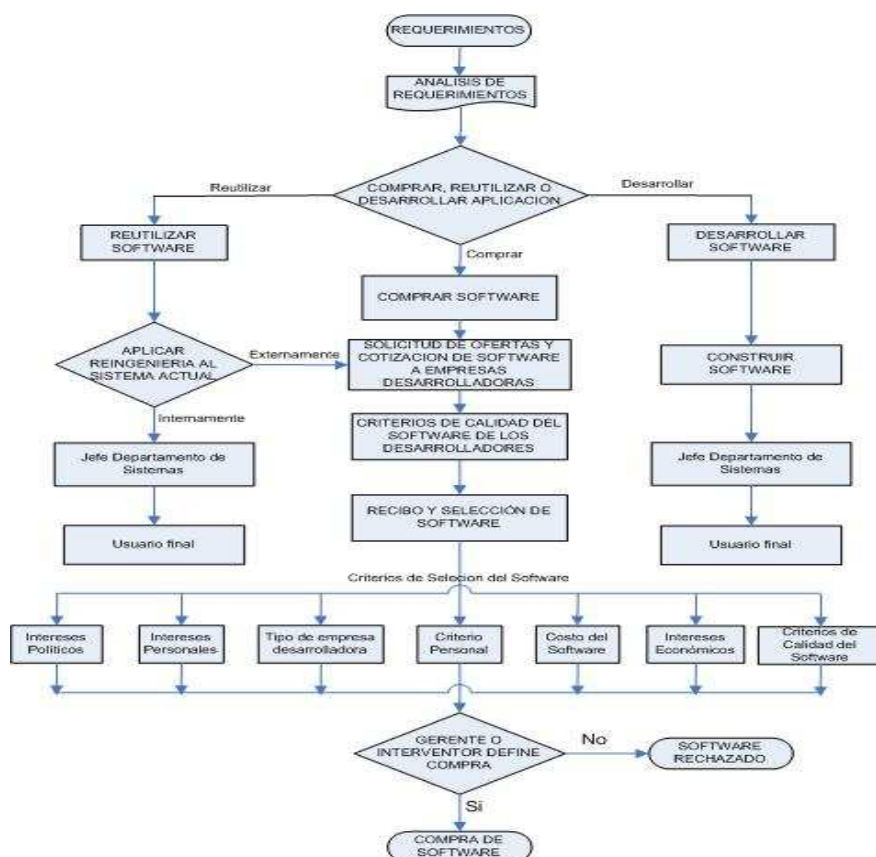
9. ANALISIS DEL SISTEMA

El desarrollo de la fase análisis se centra en el proceso de reunión de requisitos para comprender la naturaleza del Modelo.

9.1 SISTEMA ACTUAL

En la actualidad no se cuenta con un sistema o modelo que permita realizar el proceso de selección y evaluación del software de gestión, razón por la cual no se presenta el diagrama general del sistema actual, pero se presenta un esquema en el cual se hace una descripción general de la forma en que se realiza el proceso actual de selección de software.

Figura 8. SISTEMA ACTUAL: Descripción selección del software
(Diseño del Autor)



9.2 ANALISIS DE REQUERIMIENTOS

De acuerdo a los resultados obtenidos en la recolección de la información, a través de encuestas y entrevistas sobre los estándares, herramientas y criterios más importantes para la evaluación de la calidad del software de gestión, realizadas a ingenieros de sistemas y usuarios finales directos, (ver anexo 3) se concluye que:

- No existe en la actualidad criterios definidos en los usuarios directos e indirectos, sobre las normas, tendencias, métodos, herramientas, estrategias de prueba y factores de calidad en el momento de evaluar un software.
- No existe en la actualidad en el Departamento del Quindío, un Modelo que permita evaluar la calidad del software de gestión como producto terminado.
- Los ingenieros de sistemas están de acuerdo en que los atributos más importantes a tener en cuenta en el momento de evaluar la calidad del software de gestión, son las interfaces amigables, su funcionalidad, la facilidad de uso, la seguridad, la modularidad, el soporte, y el tiempo de respuesta, entre otros.
- Los usuarios finales están de acuerdo en que los atributos más importantes a considerar en un software de gestión, son: su buen funcionamiento, la facilidad de uso, mensajes claros y coherentes, interfaces gráficas, buena distribución de la pantalla de manera tal que la información aparezca siempre en las mismas zonas, tamaño del texto adecuado y que el programa permita entenderse fácilmente.

9.2.1 Determinación del Requerimiento

Para evaluar la calidad de software de gestión, se hace necesario Proponer un modelo guiado por la ISO/IEC 9126-1:2001 donde se logren beneficios que el sistema promete como producto terminado y permita a los representantes de las empresas, determinar el grado de la Calidad del software de acuerdo a las características de funcionalidad, fiabilidad, usabilidad, eficiencia, Mantenibilidad, Portabilidad.

Los resultados del Modelo permitirán a gerentes de empresas y a ingenieros de sistemas administradores de sistemas de información, contar con una herramienta útil de aceptación de las características de calidad del software como producto terminado antes de tomar una decisión de adquisición del software. Además exigirá a las empresas desarrolladoras de software, regirsen por las normas de calidad existentes, en procura de un aseguramiento de calidad al proceso de construcción del software y por ende al producto de software.

9.3 ESTUDIO DE VIABILIDAD

9.3.1 Viabilidad Operativa

Esta viabilidad establece una medida del correcto funcionamiento de la solución a la evaluación de la calidad del software de gestión, además determina el grado de aceptabilidad del Modelo por parte de los ingenieros, usuarios finales y gerentes o representantes de empresas:

Dentro de la viabilidad se tienen en cuenta dos aspectos:

- a. ¿Vale la pena desarrollar Modelo que evalúe la calidad del software, funcionara la solución propuesta al problema?
 - Si vale la pena, porque el Modelo proporciona a los gerentes, administradores y a los ingenieros de sistemas, información útil, precisa, pertinente y a tiempo sobre la calidad de un software de gestión previamente evaluado con el fin de adquirirlo.
 - El Modelo propuesto, permitirá obtener controles adecuados que garanticen la seguridad y la precisión de los resultados, pero considerando los criterios éticos al momento de evaluar el software el ingeniero de sistemas y el usuario final.
 - El modelo por su flexibilidad, ofrecerá la posibilidad de ampliarlo a la evaluación de otros tipos de software. Además su adaptabilidad permitirá la actualización del modelo.

- b. Qué opinan los gerentes, usuarios finales y los ingenieros de Sistemas sobre el Modelo Propuesto?
 - Los gerentes, usuarios finales e ingenieros de sistemas consideran que existen buenas expectativas con respecto a la implementación de un Modelo que evalúe la calidad del software de gestión como producto terminado, ya que contarán con una herramienta que les servirá de solución en el proceso de selección.
 - Es posible que por celos profesionales los ingenieros de sistemas se puedan negar a usar el modelo, debido a que su creatividad y capacidad intelectual en desarrollo de software se evaluara por medio de un modelo.

El problema puede ser superado demostrándoles a los ingenieros las bondades del modelo y creándoles conciencia de que existen otras herramientas que también hacen este tipo de evaluación.

10. PROPUESTA DE MODELO

Construir un modelo de calidad es bastante complejo y es usual que estos modelos descompongan la calidad del producto software jerárquicamente en una serie de características y subcaracterísticas que pueden usarse como una lista de comprobación de aspectos relacionados con la calidad.

10.1 MODELO DE EVALUACIÓN A USAR

Como base y de acuerdo a lo planteado en el Objetivo General, el modelo está guiado por la ISO/IEC 9126-1:2001, es importante no dejar de lado los modelos existentes planteados en el marco conceptual, para así poder determinar que la ISO/IEC 9126-1:2001, es la guía más exacta para proponer el modelo.

Teniendo en cuenta los Modelos y Estándares definidos, las características de calidad planteadas en los diferentes modelos se presentan en el siguiente cuadro (Tabla 3).

Tabla 3. Características de Calidad de diferentes Mod./Estánd. de Calidad del Software

CARACTERÍSTICAS DE CALIDAD	DROMEY	McCall	ISO-9126
Facilidad de Uso	X	X	X
Integridad		X	X
Corrección		X	
Confiabilidad	X	X	X
Eficiencia	X	X	X
Facilidad de Mantenimiento	X	X	X
Facilidad de Prueba		X	X
Flexibilidad		X	
Facilidad de Reutilización		X	
Interoperabilidad		X	X
Portabilidad	X	X	X
Fácil de Entender		X	X
Fácil de Modificar		X	X
Funcionalidad	X	X	X
Facilidad de Soporte			X
Trazabilidad			X
Conformidad			X

Análisis

Se puede observar que las 6 primeras características planteadas en la tabla 4 (Facilidad de Uso, Confiabilidad, Eficiencia, Facilidad de mantenimiento, Portabilidad y Funcionalidad) coinciden con las características planteadas en ISO/IEC 9126-1:2001, que es la guía del proyecto . Ver tabla 4.

Tabla 4. Cantidad de las características de Calidad de los Modelos/Estándares de Calidad del Software

CARACTERÍSTICAS DE CALIDAD	Cant.
Facilidad de Uso	3
Confiabilidad	3
Eficiencia	3
Facilidad de Mantenimiento	3
Portabilidad	3
Funcionalidad	3
Integridad	2
Fácil de Modificar	2
Interoperabilidad	2
Facilidad de Prueba	2
Flexibilidad	1
Facilidad de Reutilización	1
Fácil de Entender	1
Corrección	1
Facilidad de Soporte	1
Trazabilidad	1
Conformidad	1

Luego el esquema del modelo de proceso de evaluación, dado por el estándar ISO/IEC 9126 se puede aplicar en forma apropiada para el producto de software como lo es el software de gestión.

10.1.1 Modelo

En el Modelo de Evaluación se tiene tres fases o etapas, como se aprecia en la figura 9:

- Definición de Requerimientos de Calidad: Fase donde se especifican los requerimientos en términos de las características prescritas en este estándar. Las mismas sirven de entrada al proceso, al igual que todas las necesidades explícitas e implícitas del usuario, y produce como salida una especificación de

requerimientos de calidad. Los requerimientos expresan la demanda del contexto para el producto de software contable bajo evaluación.

- Preparación de la Evaluación: Fase donde se tienen tres pasos generales, a saber: 1) Selección de las Métricas de Calidad: Es donde los evaluadores deben decidir y seleccionar a las métricas debido a que las características definidas en el estándar son de alto nivel y dependen del rol, o sea quien las evalúa. 2) Definición de los Niveles de Puntaje: * se definen los niveles de satisfacción que para nuestro caso es Insatisfactorio, Marginal y Satisfactorio, expresando así el nivel de satisfacción o aceptabilidad, comprendido entre 60% y el 100%, pero dentro de ese nivel de satisfacción de aceptado entre el rango de 40 al 60% es un nivel de aceptabilidad Marginal, o sea que se encuentra en la margen mínima de aceptación; y Insatisfactorio o Rechazado, comprendido entre el 0% y el 40% el cual es un nivel de aceptabilidad pobre; pero cada métrica se evalúa en una escala de de 1 a 5, donde 1 es el mínimo valor y 5 el máximo valor. 3) Definición de los Criterios de Valoración: los evaluadores deben definir procedimientos para resumir los resultados de las diferentes características. Por ejemplo, dice el estándar, se pueden usar tablas de decisión o promedios. El procedimiento puede incluir otros aspectos

* Es el esquema de categorías de aceptabilidad que inicialmente planteó el estándar IEEE 1061 “Metodología para un estándar de calidad de Métricas de Software”, donde indica que los valores para clasificar el software se da en categorías de “aceptable”, “marginal” e “inaceptable”, y los niveles de satisfacción o niveles de aceptabilidad según la ISO-9126 son “satisfactorio” e Insatisfactorio, los cuales se usan para la evaluación de producto software

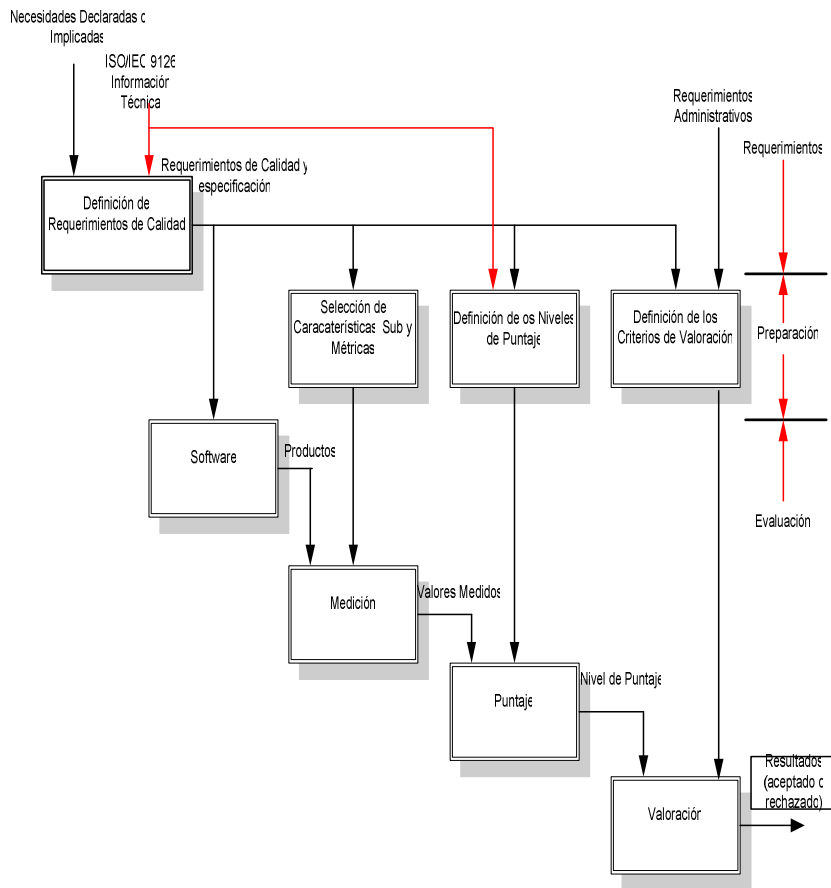


Figura 9. Modelo de Evaluación (Tomado [41], Traducido y Adaptado de ISO/IEC 9126)

- Procedimiento de la Evaluación: En la fase de *Procedimiento de Evaluación* se especifican tres pasos generales, a saber: 1) *Medición*; 2) *Puntaje*, y 3) *Valoración*. En la actividad de *Medición*, se aplican las métricas seleccionadas al producto de software contable, obteniendo como salida un valor en la escala de medición. En la actividad de *Puntaje*, se determina el nivel de puntaje alcanzado (Nivel de Aceptación) a partir del valor determinado en la actividad previa. La *Valoración* es la última actividad del modelo de proceso de evaluación de ISO 9126.

El resultado es una declaración de aceptabilidad o satisfacción, insatisfacción o rechazo de las características de calidad del producto de software. De este modo, la calidad resultante puede ser comparada con otros aspectos como costo (según los criterios), viabilidad financiera y económica y viabilidad legal* .

10.2 SELECCION DE FACTORES / CARACTERISTICAS

La Calidad aceptable no es definida por ingenieros, simplemente es la capacidad para el uso. La calidad es medible por la persona que usa el software, para una verdadera prueba, esta es realizada después de que el software se ha entregado como producto terminado.

Para una selección satisfactoria, se toman los resultados de la Encuesta a los Usuarios Finales (ver anexo 3), estas determinaron que entre los criterios que consideran más relevantes en una buena calidad del software de gestión están en la tabla 5:

Tabla 5. Criterios Relevantes de una Buena Calidad “Usuario”

- El software debe tener facilidad de uso.
- Facilidad de Corrección
- Facilidad mantenimiento
- Facilidad para los procesos de Auditoria
- Ser flexible
- Ser fiable
- Ser seguro en su información y el programa
- Modular
- Se deben utilizar en lo posible interfaces amigables
- Distribución de la información, de tal manera que esta aparezca en las mismas zonas
- El programa debe entenderse fácilmente.
- Garantizar la fiabilidad de los datos
- Velocidad de respuesta
- Cumplimiento adecuado de los requerimientos de usuario
- Variedad y facilidad de interpretación en la presentación de reportes e informes
- Control en la entrada de datos no validos
- Manuales de usuario sencillos
- No deben existir bloqueos frecuentes en el sistema.
- Ambiente de soporte completo y fácil de Usar

* La investigación sobre otros aspectos como costos, viabilidad financiera y económica, y viabilidad legal para evaluación de software sería motivo de otro trabajo de tesis *per se*

De acuerdo a los resultados de las encuestas y a lo que requiere el usuario del software de gestión como producto terminado y cruzarlas con las "ILIDADES" o características generales de la calidad del software, expuestas en la siguiente tabla:

Tabla 6. "ILIDAES" de la Calidad del Software

- La Exactitud:	- Cumple exactamente con los requerimientos, obedece a los requerimientos?
- Eficiencia:	- Cuantos recursos son necesarios?
- Expansibilidad:	- Es fácil expandir el software?
- Flexibilidad:	- Es fácil cambiar el software?
- Integridad:	- Es seguro es software?
- Interoperabilidad:	- Es fácil de acoplar el software con otros?
- Mantenibilidad:	- Es fácilmente de reparar el software?
- Manejabilidad:	- Es fácilmente manejable?
- Portabilidad:	- Es fácil de transportar a otros sistemas?
- Usabilidad:	- Es fácil de usar?
- Fiabilidad:	- Fallará a menudo?
- Reusabilidad:	- Se puede usar en otros sistemas
- Seguridad:	- Previene o controla?
- Durabilidad:	- Puede el software sobrevivir a un daño o contingencia?
- Verificabilidad:	- La comprobación de sus procesos son fáciles?

Luego se realiza un cruce una matriz, como se muestra en la tabla 7, de las características generales o llidades, los criterios relevantes, así:

Tabla 7. Matriz de Cr terios Relevantes vs. Iidades

CARACTERISTICAS GENERALES - IIDADES	Exactitud	Eficiencia	Expansibilidad	Flexibilidad	Fiabilidad	Integridad	Interoperabilidad	Mantenibilidad	Manejabilidad	Portabilidad	Usabilidad	Reusabilidad	Seguridad	Durabilidad	Verificabilidad
CRITERIOS RELEVANTES															
El software debe tener facilidad de uso en varios ambientes															
Facilidad de Correcci�n															
Facilidad de Mantenimiento															
Administraci�n y gesti�n de Seguridad															
Ser Flexible															
Ser fiable															
Tolerante a Fallos															
Administraci�n de Seguridad															
Modular															
Se deben utilizar en lo posible Interfaces amigables															
Distribuci�n de la Informaci�n de tala manera que aparezca en las mismas zonas															
El programa debe entenderse f�cilmente															
Garantizar la fiabilidad de los datos															
Velocidad de Respuesta															
Cumplimiento de acuerdo a los requerimientos del usuario															
Variedad y facilidad en la presentaci�n de los diferentes reportes e informes															
Control en la entrada de datos no v�lidos															
Manuales de usuario sencillos															
No deben existir bloqueos frecuentes en el sistema															
Ambiente de soporte y f�cil de usar															

Hasta ahora se ha observado que los enfoques utilizados para la evaluaci n, no se han concentrado en medir a la calidad como una caracter stica de muy alto nivel* que contiene y describe a otras caracter sticas como *usabilidad*, *funcionalidad*, *eficiencia*, *confiabilidad*, u otras como *mantenibilidad* y *portabilidad*, que son consideradas caracter sticas de alto nivel.

Entonces de acuerdo a lo anterior, se pueden cruzar esos criterios obtenidos y refinar las "IIDADES" para acoplarlas en la Norma mencionada y al Modelo; obteniendo as  las caracter sticas de alto nivel, seg n tabla 8; permitiendo un modelo flexible ya que debe:

* Existen caracter sticas generales (las Iidades) y las caracter sticas de alto nivel que involucran las generales en seis caracter sticas , las cuales son las que expresa la ISO/IEC 9126

Primero, permitir agregar o sacar características, subcaracterísticas de un modo modular, conforme a las necesidades específicas del perfil o perfiles de usuario, del Software de Gestión, y del contexto organizacional.

Segundo, porque dependiendo del Software, criticidad del software y necesidad de precisión, debe permitir ajustar los criterios de evaluación y/o procedimientos de escrutinio.

Tabla 8. Ilidades – Características de Alto Nivel

CARACTERISTICAS GENERALES - ILIDAES	Exactitud	Eficiencia	Expansibilidad	Flexibilidad	Fiabilidad	Integridad	Interoperabilidad	Mantenibilidad	Manejabilidad	Portabilidad	Usabilidad	Reusabilidad	Seguridad	Durabilidad	Verificabilidad
CRITERIO RELEVANTES															
El software debe tener facilidad de uso en varios ambientes															
Facilidad de Corrección															
Facilidad de Mantenimiento															
Administración y gestión de Seguridad															
Ser Flexible															
Ser fiable															
Tolerante a Fallos															
Administración de Seguridad															
Modular															
Se deben utilizar en lo posible Interfaces amigables															
Distribución de la Información de tala manera que aparezca en las mismas zonas															
El programa debe entenderse fácilmente															
Garantizar la fiabilidad de los datos															
Velocidad de Respuesta															
Cumplimiento de acuerdo a los requerimientos del usuario															
Variedad y facilidad en la presentación de los diferentes reportes e informes															
Control en la entrada de datos no válidos															
Manuales de usuario sencillos															
No deben existir bloqueos frecuentes en el sistema															
Ambiente de soporte y fácil de usar															
	FUNCIONALIDAD	EFICIENCIA	FUNCIONALIDAD	MANTENIBILIDAD	FIABILIDAD	FUNCIONALIDAD	FUNCIONALIDAD	MANTENIBILIDAD	USABILIDAD	PORTABILIDAD	USABILIDAD	PORTABILIDAD	FUNCIONALIDAD	FUNCIONALIDAD	FIABILIDAD

Como se puede observar en la tabla 8, existen características que se presentan más veces (ver última fila), por ejemplo: Funcionalidad se presenta 6 veces y eficiencia una vez, por lo tanto la funcionalidad estará en el orden primero y la eficiencia en el orden sexto o último.

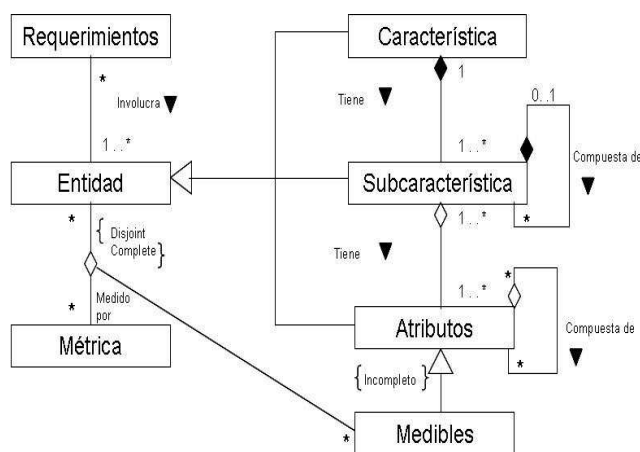
Definidas entonces las características de alto nivel que se acoplan a la guía planteada para la propuesta del modelo de calidad, se puede proponer este.

10.2.1 Modelo Conceptual UML para la Estándar 9126

La idea principal detrás de esta norma es la definición de un modelo de calidad y su uso como un almacén para la evaluación de software. Un modelo de calidad se define por medio de las características y sus subcaracterísticas en que se descompone. La ISO/IEC 9126 con sus seis características de alto nivel: la funcionalidad, la fiabilidad, la usabilidad, la mantenibilidad, la portabilidad y la eficiencia. Luego un modelo conceptual en UML que refina las características es como se aprecia en la figura 10, donde los estados de asociación se pueden definir de la siguiente manera:

Para una métrica medible, los estados de asociación pueden ser diferentes dependiendo de la subcaracterística y de la característica que la posea ya sea esta por agregación por referencias o composición por valor.

Figura 10. Modelo Conceptual UML para la ISO/IEC 9126 (tomado y traducido de ISO/IEC 9126)*



* ISO/EIC IS 9126-1 "Information Technology-Software Product Evaluation: quality characteristics and guidelines for their use",2001.

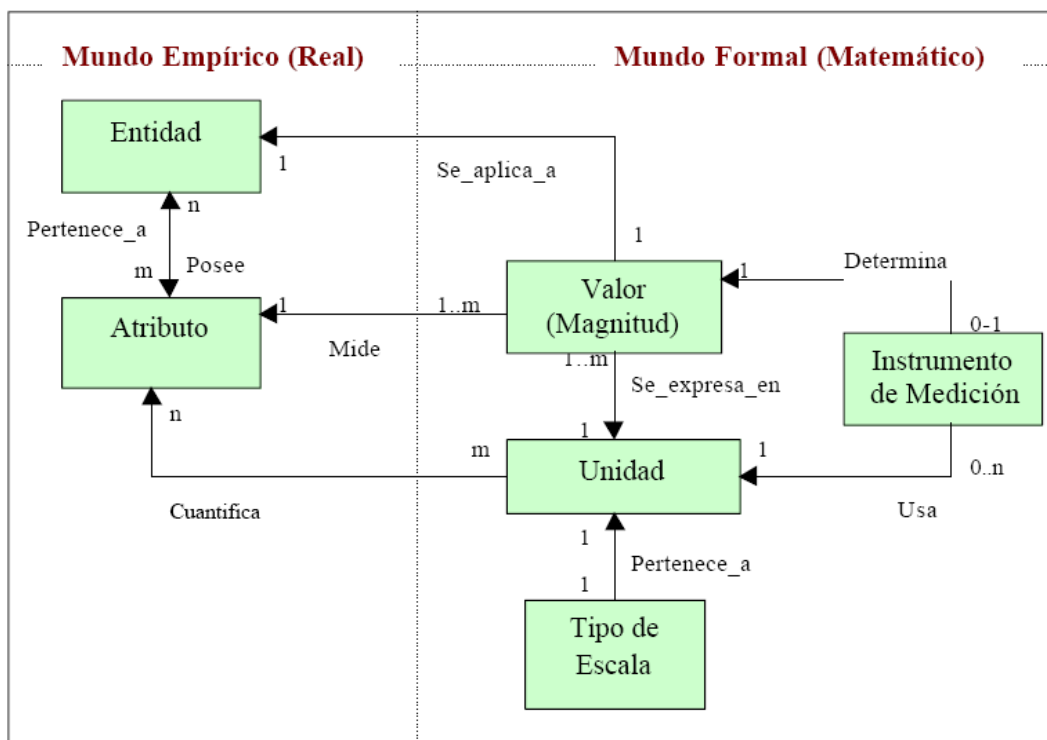
-----* Zuse, H., 1998, "A Framework of Software Measurement", Walter de Gruyter, Berlín-NY.

10.2.2 Modelo Conceptual UML para Métricas

La figura 11 muestra el esquema para métricas, una métrica es la resultante de una correspondencia entre un atributo de un ente (del dominio empírico) y el valor o magnitud (del dominio formal, numérico), y que sirve como referencia para describir y explicar aspectos o situaciones del mundo empírico. Los atributos se miden generalmente por medio de métricas directas (aunque no siempre) y las características y subcaracterísticas se miden por medio de métricas**.

En la figura se observan dos mundos o dominios modelados: el empírico y el formal. En el primer mundo (que tiene un impacto empírico, perceptible sobre los usuarios) se modela y formula el sistema relacional empírico; en el segundo mundo, se modela y formula el sistema relacional formal (numérico y, en algunos casos simbólico) y los mecanismos para ayudar en la determinación de valores. A propósito, uno de los aportes de esta investigación es determinar ese Instrumento de Medición en el Mundo Formal y poderlo aplicar en el Mundo real.

Figura 11. Modelo Conceptual UML para Métricas (Adaptado de Zuse, H., 1998)*



** El lector para conocer definiciones de métricas, métricas directas e indirectas se debe remitir al numeral 4.4.2, donde se realiza una definición de estos conceptos.

* Zuse, H., 1998, "A Framework of Software Measurement", Walter de Gruyter, Berlín-NY.

10.3 FORMULACION DEL MODELO

En este capítulo se presenta el planteamiento del Modelo de Evaluación de Calidad del Software de Gestión, donde se parte de las características de alto nivel planteadas por la ISO/IEC 9126-1: Funcionalidad, Fiabilidad (confiabilidad), Usabilidad (facilidad de Uso), Mantenibilidad, Portabilidad, Eficiencia, dadas en su orden de acuerdo a la tabla 8. Por lo tanto a continuación se muestra una estructura del Modelo de Evaluación ver Figura 12.

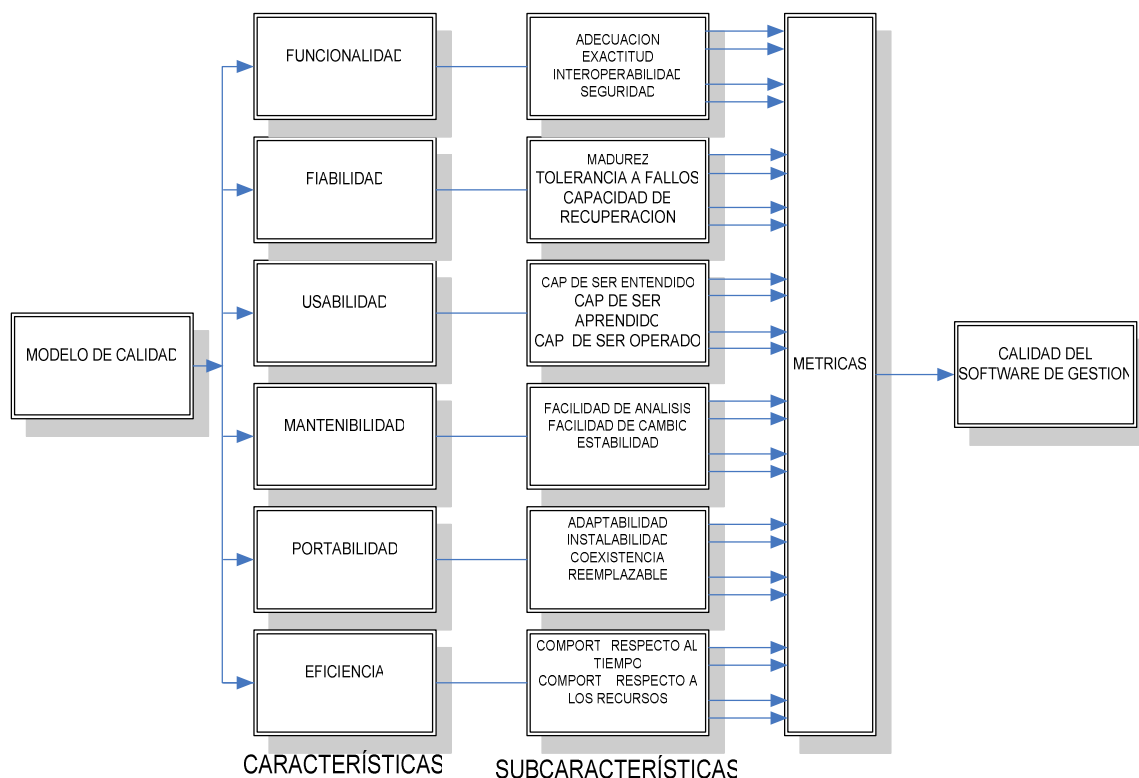


Figura 12. Elementos del Modelo de Evaluación de la Calidad

A continuación cada característica es descrita con sus correspondientes subcaracterísticas, como sus respectivas atributos de cada subcaracterística asociadas a obtener el Modelo de Evaluación de Calidad.

Funcionalidad, es la cantidad de posibilidades o funciones que satisfacen las necesidades específicas e implícitas cuando el software se usa bajo condiciones específicas. Uno de los problemas más difíciles que encuentra un líder de proyecto es saber cuánta funcionalidad es suficiente, entonces aparece la presión por más funciones, conocida como “adornitis” (requerimientos desmedidos), donde sus consecuencias son malas para proyectos internos y peor para productos comerciales. La Funcionalidad posee las siguientes subcaracterísticas:

- Adecuación: Capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.

- Exactitud (precisión): Capacidad del producto software para proporcionar los resultados o efectos correctos o acordados, con el grado necesario de precisión.

- Interoperabilidad: Capacidad del producto software para interactuar con uno o más sistemas especificados.

- Seguridad de acceso: Capacidad del producto software para proteger información y datos de manera que las personas o sistemas no autorizados no puedan leerlos o modificarlos, al tiempo que no se deniega el acceso a las personas o sistemas autorizados.

- Cumplimiento de la funcionalidad: Capacidad del producto software para adherirse a normas, convenciones o regulaciones en leyes y prescripciones similares relacionadas con funcionalidad.

Tabla 9 Característica de Funcionalidad en el Modelo de Evaluación de Calidad

CARACTERISTICA	Concepto	SUBCARACTERISTICA	Pregunta
FUNCIONALIDAD	Las Funciones y propiedades satisfacen las necesidades explicitas e implícitas, es el qué?	Adecuación	Tiene el conjunto de funciones apropiadas para las tareas especificadas?
		Exactitud	Hace lo que fue acordado en forma esperada y correcta?
		Interoperabilidad	Interactúa con otros sistemas especificados?
		Seguridad de Acceso	Previene Accesos no autorizados a los datos y programas?
		Cumplimiento de la Funcionalidad	Esta de acuerdo con las leyes o normas y estándares, u otras prescripciones?

Fiabilidad, o confiabilidad del software es la probabilidad que un programa realice su objetivo satisfactoriamente (sin fallos) en un determinado periodo de tiempo y en un entorno concreto (denominado perfil operacional)". No hay duda que la fiabilidad de un programa de computadora es un elemento importante de su calidad general. Si un programa falla frecuentemente en su funcionamiento, no importa si el resto de los factores de calidad son aceptables. La fiabilidad del software, a diferencia de otros factores de calidad, puede ser medida o estimada mediante datos históricos o de desarrollo. La Fiabilidad posee unas subcaracterísticas, a saber:

- Madurez: Es la Capacidad del producto software para evitar fallar como resultado de fallos en el software.

- Tolerancia a fallos: Capacidad del software para mantener un nivel especificado de prestaciones en caso de fallos de software o de infringir sus interfaces especificadas.

- Capacidad de recuperación: Es la capacidad del producto software para reestablecer un nivel de prestación especificada y de recuperar los datos directamente afectados en caso de fallo.

- Cumplimiento de la fiabilidad: Capacidad del producto software para adherirse a normas, convenciones o regulaciones relacionadas con al fiabilidad.

Tabla 10. Característica de Fiabilidad en el Modelo de Evaluación de Calidad

CARACTERISTICA	Concepto	SUBCARACTERISTICA	Pregunta
<i>FIABILIDAD(Confianza)</i>	Puede mantener el nivel de rendimiento bajo ciertas condiciones?	Madurez	Con que frecuencia presenta fallas por defectos ?
		Tolerancia a fallos	Si suceden fallas como se comporta en cuanto a su actuación?
		Capacidad de Recuperación	Es capaz de recuperar datos en caso de fallas?
		Cumplimiento de Fiabilidad	Esta de acuerdo con las leyes o normas y estándares?

USABILIDAD, Es la Capacidad de un producto de software de ser atractivo, que la gente puede aprender a usar el software y aplicarlo para resolver problemas. También incluye la facilidad de instalación, operación y monitoreo. La definición acentúa los diferentes niveles de experiencia de los usuarios potenciales. Este requerimiento representa uno de los mayores retos para los diseñadores de software interesados en facilidad de uso. La facilidad de uso no es un atributo inherente al software, no puede especificarse independientemente del entorno de uso y de los usuarios que vayan a utilizar el sistema. La facilidad de uso no puede definirse como un atributo simple de un sistema, pues implica distintos aspectos dependiendo del tipo de sistema. Las subcaracterísticas de la “Facilidad de Uso” son:

- Comprensibilidad o capacidad para ser entendido: Es la Capacidad del producto software que permite al usuario entender si el software es adecuado y cómo puede ser usado para unas tareas o condiciones de uso particulares.

- Facilidad de aprendizaje o capacidad para ser aprendido: Es cuando aquel producto software le permite al usuario aprender sobre su aplicación.

- Operatividad o Capacidad para ser operado: Capacidad del producto software que permite al usuario operarlo y controlarlo.

- Capacidad de atracción: Es la Capacidad del producto software para ser atractivo al usuario desde el punto de vista de Explicitud, Adaptabilidad al usuario, Claridad, Facilidad de ayuda, Atractivo y Amistoso al usuario.

- Cumplimiento de la usabilidad: Capacidad del producto software para adherirse a Normas, convenciones, guías de estilo o regulaciones relacionadas con la usabilidad.

Tabla 11. Característica de Usabilidad el Modelo de Evaluación de Calidad

CARACTERISTICA	Concepto	SUBCARACTERISTICA	Pregunta
USABILIDAD (Fiabilidad de Uso)	El software es fácil de usar y de aprender?	Capacidad de ser entendido	Es fácil de entender y reconocer la estructura, la lógica y su aplicabilidad?
		Capacidad de ser Aprendido	Es fácil de aprender a usar?
		Capacidad de ser operado	Es fácil de operar?
		Capacidad de Atracción	Es atractivo al usuario por su adaptabilidad, claridad y facilidad de ayuda?
		Cumplimiento de la Usabilidad	Esta de acuerdo con las leyes o normas y estándares?

Mantenibilidad, Es la facilidad con la cual se puede corregir un producto de software si se encuentra un error, adaptarlo si su entorno cambia, o mejorarlo si el cliente desea un cambio en los requisitos. El mantenimiento del software representa más esfuerzo que cualquier otra actividad de la Ingeniería del Software. Existen tres tipos de mantenimiento: (1) mantenimiento correctivo: corregir errores, (2) mantenimiento adaptativo: modificar el software de acuerdo con el entorno; y (3) mantenimiento perfectivo: añadir nueva funcionalidad. El mantenimiento preventivo no está tan extendido y consiste en cambiar el producto pensando en mejoras futuras. En las subcaracterísticas de la Facilidad de Mantenimiento, se tiene:

- Capacidad para ser analizado o Facilidad de análisis: Es la capacidad del producto software para serle diagnosticadas deficiencias o causas de los fallos en el software, o para identificar las partes que han de ser modificadas.

- Capacidad para ser cambiado o Facilidad de cambio: Capacidad del producto software que permite que una determinada modificación sea implementada.

- Estabilidad: Capacidad del producto software para evitar efectos inesperados debidos a modificaciones del software.

- Capacidad para ser probado o Facilidad de prueba: Capacidad del producto software que permite que el software modificado sea validado.

- Cumplimiento de la mantenibilidad: Capacidad del producto software para adherirse a normas o convenciones relacionadas con la mantenibilidad.

Tabla 12. Característica de Mantenibilidad el Modelo de Evaluación de Calidad

CARACTERISTICA	Concepto	SUBCARACTERISTICA	Pregunta
MANTENIBILIDAD	Es fácil de manejar y testear?	Capacidad para ser analizado	Es fácil diagnosticar una falla o identificar partes a modificar?
		Capacidad para ser cambiado	Es fácil de modificar y adaptar?
		Estabilidad	Existen riesgos o efectos inesperados cuando se realizan cambios?
		Capacidad para ser probado	Son fáciles de validar las modificaciones?
		Cumplimiento de la mantenibilidad	Esta de acuerdo con las leyes o normas y estándares, u otras prescripciones

Portabilidad, consiste en la facilidad de transportar productos de software a varios Ambientes de hardware y software. Esta característica se ocupa de variaciones no solo del hardware sino de algo más general, la combinación hardware-software, que es la máquina que en realidad programamos, que incluye el sistema operativo, el sistema gráfico, y otras herramientas fundamentales. En la portabilidad se tienen las siguientes subcaracterísticas:

- Adaptabilidad: Es cuando el producto software tiene la capacidad para ser adaptado a diferentes entornos especificados, sin aplicar acciones o mecanismos distintos de aquellos proporcionados para este propósito por el propio software considerado.

- Instalabilidad: Capacidad del producto software para ser instalado en un entorno Especificado.

- Coexistencia: Capacidad del producto software para coexistir con otro software Independiente, en un entorno común, compartiendo recursos comunes.

- Capacidad para reemplazar: Capacidad del producto software para ser usado en lugar de otro producto software, para el mismo propósito, en el mismo entorno.

- Cumplimiento de la portabilidad: Capacidad del producto software para adherirse a normas o convenciones relacionadas con la portabilidad.

Tabla 13. Característica de Portabilidad el Modelo de Evaluación de Calidad

CARACTERISTICA	Concepto	SUBCARACTERISTICA	Pregunta
<i>PORTABILIDAD</i>	Es fácil de transferir de un ambiente a otro?	Adaptabilidad	Es fácil de adaptar a otros entornos?
		Instalabilidad	Es fácil de instalar en el ambiente especificado?
		Coexistencia	Adhiere a los estándares y convenciones de portabilidad?
		Capacidad para ser reemplazado	Es fácil de usarlo en lugar de otro software para ese ambiente?
		Cumplimiento de la Portabilidad	Esta de acuerdo con las leyes o normas y estándares, u otras prescripciones

Eficiencia, La eficiencia es la habilidad del software para poner la cantidad mínima de demanda sobre los recursos de hardware como sea posible, tales como el tiempo de procesador, espacio ocupado en memorias internas o externas, ancho de banda usado en dispositivos de comunicación. Normalmente se identifica eficiencia como velocidad de ejecución. Esta interpretación no es correcta. El software es eficiente si realiza un uso racional de todos los recursos de hardware. La eficiencia incluye la utilización equilibrada de: (1) tiempo de CPU, (2) memoria principal, (3) memoria secundaria, (4) canales de entrada/salida, (5) velocidad de ejecución y (6) tiempo de respuesta.

El tema de la eficiencia debe estar balanceado con otros objetivos como la extensibilidad y la reutilización. Sin embargo, no hay que disminuir la importancia de la eficiencia puesto que nadie quiere estar esperando demasiado las respuestas del sistema o verse obligado a estar comprando más memoria para ejecutar un programa. Las subcaracterísticas de la Eficiencia son:

- Comportamiento con respecto al tiempo: Capacidad del producto software para proporcionar tiempos de respuesta, tiempos de proceso y potencia apropiados, bajo condiciones determinadas
- Comportamiento con respecto a los recursos: Capacidad del producto software para usar las cantidades y tipos de recursos adecuados cuando el software lleva a cabo su función bajo condiciones determinadas.
- Cumplimiento de la eficiencia: Capacidad del producto software para adherirse a normas o convenciones relacionadas con la eficiencia.

Tabla 14. Característica de Eficiencia el Modelo de Evaluación de Calidad

CARACTERISTICA	Concepto	SUBCARACTERISTICA	Pregunta
<i>EFICIENCIA</i>	Es rápido y minimiza en cuanto al uso de recursos bajo ciertas condiciones?	Comportamiento Con respecto al tiempo	Cual es el tiempo de respuesta y adecuación en la ejecución de una función?
		Comportamiento Con respecto a los recursos	Cuantos recursos usa y durante cuanto tiempo en una determinada función?
		Cumplimiento de la Eficiencia	Esta de acuerdo con las leyes o normas y estándares, u otras prescripciones?

10.3.1 Definición de Perfiles en el Modelo

Teniendo en cuenta el marco conceptual de Perfiles de Usuario de la ISO/IEC 9126; y particularmente, el estándar citado afirma que la relativa importancia de las características de calidad (como usabilidad, funcionalidad, confiabilidad, eficiencia, portabilidad, y mantenibilidad) varían dependiendo del punto de vista considerado y de la criticidad de los componentes del artefacto a evaluar. Por ejemplo, la visión del usuario, concierne al interés de los mismos en usar el software contable, como así también su rendimiento, su eficiencia, su facilidad de uso, entre otros aspectos. Los usuarios no están interesados en características internas o de desarrollo de los artefactos (sin embargo, atributos internos contribuyen a la calidad de uso).

En cambio, la visión de calidad del Ingeniero de Sistemas debe considerar no sólo los requerimientos del producto para la visión del usuario sino también la calidad para los productos intermedios resultantes de las actividades de la fase de desarrollo. Para poder evaluar a la calidad de los productos, los evaluadores deberán considerar las métricas apropiadas para la misma característica. Por otra parte, los desarrolladores están preocupados en características de calidad del producto como mantenibilidad y portabilidad.

Por lo tanto, como ya se tienen las características y subcaracterísticas definidas y de acuerdo a los alcances y limitaciones de la Investigación "...desde el punto vista del usuario y el Ingeniero de sistemas de la empresa que asesora a la alta dirección en la toma de decisión para la comprar o adquisición del software de gestión..", y donde se evaluará en forma de "... cuestionario tanto al ingeniero de sistemas y usuario final.." para lo cual es importante definir cual es el papel o el rol de cada uno como entrevistado con respecto a las características y subcaracterísticas como se ve en la tabla 15.

Tabla 15. Roles de Acuerdo a las Características

CARACATERISTICA	SUBCARACTERISTICA	Rol o Papel
<i>FUNCIONALIDAD</i>	Adecuación Exactitud Interoperabilidad Seguridad	<i>Usuario, Ingeniero de Sistemas</i> <i>Usuario, Ingeniero de Sistemas</i> <i>Usuario, Ingeniero de Sistemas</i> <i>Usuario, Ingeniero de Sistemas</i>
<i>FIABILIDAD</i>	Madurez Tolerancia a Fallos Capacidad de Recuperación	<i>Usuario, Ingeniero de Sistemas</i> <i>Usuario, Ingeniero de Sistemas</i> <i>Usuario, Ingeniero de Sistemas</i>
<i>USABILIDAD</i>	Capacidad de ser entendido Capacidad de ser Aprendido Capacidad de ser operado Capacidad de Atracción	<i>Usuario, Ingeniero de Sistemas</i> <i>Usuario, Ingeniero de Sistemas</i> <i>Usuario, Ingeniero de Sistemas</i> <i>Usuario, Ingeniero de Sistemas</i>
<i>MANTENIBILIDAD</i>	Capacidad para ser analizado Capacidad para ser cambiado Estabilidad Capacidad para ser probado	<i>Ingeniero de Sistemas</i> <i>Ingeniero de Sistemas</i> <i>Ingeniero de Sistemas</i> <i>Ingeniero de Sistemas</i>
<i>PORTABILIDAD</i>	Adaptabilidad Instalabilidad Coexistencia Capacidad para ser reemplazado	<i>Ingeniero de Sistemas</i> <i>Ingeniero de Sistemas</i> <i>Ingeniero de Sistemas</i> <i>Ingeniero de Sistemas</i>
<i>EFICIENCIA</i>	Comportamiento Con respecto al tiempo Comportamiento Con respecto a los recursos	<i>Ingeniero de Sistemas</i> <i>Ingeniero de Sistemas</i>

10.4 VALORACION DEL MODELO

Antes de entrar a valorar el Modelo, es importante definir el tipo de métrica a usar para poder estandarizar los resultados, para lo cual se tendrán dos tipos de métricas como son las de tipo rango y likert [46], como se define en la tabla 16.

Tabla 16. Tipos de Métrica

Tipo de Métrica	Valor	Valor Normalizado
Likert	Totalmente en desacuerdo	1
	En Desacuerdo	2
	Neutral (afirmación)	3
	De Acuerdo	4
	Totalmente de Acuerdo	5
Rango	$0 = < N < \text{Valor}$	1
	$\text{Valor} = < N < \text{Valor } w$	2
	$\text{Valor } w = < N < \text{Valor } x$	3
	$\text{Valor } x = < N < \text{Valor } y$	4
	$\text{Valor } y = < N < \text{Valor } z$	5

Para la valoración del modelo se tiene en cuenta las etapas definidas en el estándar y presentadas en la Figura 9. Modelo de Evaluación en la etapa de Preparación, a saber:

ETAPA 1: Definición de Requerimientos de Calidad: donde se especifican los requerimientos en términos de las características, subcaracterísticas y métricas a

usar, se eliminan o adicionan de acuerdo a los requerimientos, [ver anexo 4](#). Las mismas sirven de entrada al proceso, al igual que todas las necesidades explícitas e implícitas de cada uno de los evaluadores como definir los pesos que utilizarán en el modelo, si estos son concertados para evaluar el producto con iguales pesos o cada evaluador coloca los pesos de acuerdo a su criterio profesional, experiencia y requerimientos. Puede presentarse el caso de que ninguno de los evaluadores determine los pesos y a conveniencia de los mismos determinar que sea el Modelo que otorgue los pesos, para las características y subcaracterísticas.

ETAPA 2: Preparación de la Evaluación:

Paso 1: Selección de las Características, subcaracterísticas y métricas de Calidad: Es donde los evaluadores deben decidir y seleccionar la característica (que puede ser: una, varias o todas, ya que depende del rol de cada evaluador), la subcaracterística (que puede ser una, varias o todas de cada una de las características) y automáticamente se están eligiendo las métricas propuestas inicialmente en el modelo. [Ver anexo 4](#)

Paso 2: Definición de los Niveles de Puntaje: En este paso y considerando la norma ISO 9126-1, se deberá: a) establecer un valor especificado o peso por el evaluador para cada característica en un rango de de 0 a 100%, la sumatoria de los pesos de las características, no deberá exceder el 100% de las características seleccionadas dentro del modelo.

b) Establecer un peso especificado por el evaluador para cada subcaracterística en rango de 0 a 100%, donde la sumatoria de los pesos de las subcaracterísticas pertenecientes a cada característica, no deberá exceder el 100% dentro de la característica a la cual pertenece.

: - En caso que la característica y sus respectivas subcaracterísticas no desee ser evaluadas, el valor especificado por el evaluador será cero.

- Si el evaluador no desea asignar los pesos El modelo puede definir automáticamente el peso a cada subcaracterística como el promedio de las subcaracterísticas existentes en relación a un 100% dentro de la característica y de igual manera definir automáticamente el peso a cada característica como el promedio de las características existentes en relación a un 100% dentro del total del producto de software.

c) Definir los niveles de satisfacción; que para cada métrica es de un rango de 1 – 5⁴⁶, donde uno no cumple con la métrica y 5 cumple en su totalidad con la métrica. Al Final, la evaluación total de la calidad o la evaluación de calidad de cada característica (según el caso) del producto de software

⁴⁶ **La Escala de Likert**, Es un tipo de instrumento de medición o de recolección de datos que se dispone en la investigación. Es una escala para medir las actitudes. Consiste en un conjunto de ítems bajo la forma de afirmaciones o juicios ante los cuales se solicita la reacción (favorable o desfavorable, positiva o negativa) de los individuos.

contable, esta definida en un rango de Rechazado o Insatisfactorio entre 0 – 40.00), Marginal desde 40.0 hasta 60.00, que pertenece al rango de satisfactorio o Aceptación y Satisfactorio o Aceptación desde 60.00 – 100.00.

Paso 3: Definición de los Criterios de Valoración: El modelo define procedimientos para resumir los resultados de las diferentes características y subcaracterísticas. De acuerdo al estándar, se permiten diferentes criterios de valoración y para este modelo se usa el criterio y o procedimiento del promedio, así:

- a. Métricas: Se toma el peso dado por evaluador o el modelo de la subcaracterística, paso 2 de la etapa 2, a la cual pertenece la métrica y se divide entre el número de métricas existentes de dicha subcaracterística, obteniéndose así el peso (%) correspondiente a cada métrica dentro de cada subcaracterística.
 - Luego se toma el peso obtenido de la métrica (anterior) y se multiplica por el nivel satisfacción dado por el evaluador a dicha métrica, este producto se divide por el máximo valor de satisfacción de la métrica, que es 5. Obteniéndose la calidad de la métrica (%).
- b. Subcaracterística: Es la sumatoria de las calidades obtenidas en cada métrica. Obteniéndose la calidad de la subcaracterística en un porcentaje correspondiente al peso (%) dado inicialmente por el evaluador o el modelo en la subcaracterística.
- c. Característica: Es la sumatoria de las calidades obtenidas en cada subcaracterísticas pertenecientes a dicha característica. Obteniéndose la calidad de la característica en un porcentaje correspondiente al peso (%) dado por el evaluador o el modelo en la característica.
 - La Calidad de la característica con respecto al total de Características evaluadas (o 100% del producto de software contable), se obtiene de multiplicar la calidad de la característica ya obtenida por el peso (%) dado por el evaluador o el modelo.
- d. Calidad Total del Producto Software: Se obtiene de la sumatoria de las calidades de cada una de las características evaluadas.

ETAPA 3: Evaluación

Paso 1: Medición, se aplican las métricas seleccionadas al producto de software contable.

Paso 2: Puntaje, De acuerdo a la evaluación por cada uno de los evaluadores, se determina el nivel de puntaje alcanzado de acuerdo a los pesos dados a cada subcaracterística y característica y lo definido en el paso 2 y 3 de la etapa anterior.

Paso 3: *Valoración*. Se presenta en el siguiente ítem de este capítulo.

10.4.1 Valoración (Resultados – Aceptación o Rechazo)

Una vez evaluado el software contable y de acuerdo a los resultados arrojados por el modelo:

- El evaluador “Ingeniero de Sistemas”, evaluará todas las características.
- El evaluador “Usuario”, evaluará las características correspondientes a su perfil.
 - *En el caso que de que existan 2 o más evaluadores de cada perfil los resultados será el promedio de dichas evaluaciones.*
- Se determina si la calidad del Producto software en el rango establecido anteriormente: Aceptado o Satisfactorio y Rechazado o Insatisfactorio, ejemplo si la calidad total arrojada por el modelo es de 78.87%, se concluye que es Aceptado.
- Se determinan las calidades por Características; ejemplo, el software contable tiene un Calida Total de Aceptado (78.87%), pero su funcionalidad es del 80.1 (Satisfactorio).
- La Calidad del Software es del 80.1 (Aceptado), pero si tomamos Figura 09. Modelo de Evaluación del estándar, Requerimientos Administrativos (Costos) y el costo del software es muy elevado para el presupuesto de la empresa, ya es determinación de los evaluadores definir otros estudios de viabilidad, como un estudio de viabilidad financiera y económica (que no es competencia de este proceso de investigación) para la adquisición del software. Es importante saber, que el modelo es un elemento para determinar técnicamente la calidad del Producto desde sus características y con este concepto técnico ayudar en la determinación de adquisición del mismo.
- El Modelo está en la capacidad de generar gráficas estadísticas por Calidad Total, Características y subcaracterísticas de cada producto evaluado.

10.5 CASO DE ESTUDIO

Con el fin de ejemplificar el modelo, se presenta este caso de estudio, donde se plantea la aplicación de la norma ISO/IEC 9126-1, o sea el modelo de calidad propuesto y con base en ello se obtiene el software contable de la empresa A y B*, a los cuales se comenzarán a evaluar las características del producto de software con sus respectivas subcaracterísticas y métricas teniendo en cuenta las

* Se toman para el caso de estudio dos entidades del sector público, que poseen software contable como son la empresa A, Universidad del Quindío y la empresa B, la Secretaría de Educación Municipal de Armenia, usando el modelo se evalúa la calidad de los paquetes contables que dichas empresas poseen como son XENCO y FACIL respectivamente; las empresas solicitan confidencialidad en el manejo de la información y los datos arrojados por el modelo, por tal razón se seguirán llamando Producto A, Empresa A y Producto B, empresa B.

etapas que constituyen el desarrollo de esta aplicación. Para ello, se realizarán las siguientes etapas y pasos:

ETAPA 1: Definición de Requerimientos de Calidad: donde se especifican los requerimientos en términos de las características, subcaracterísticas y métricas a usar. En este caso se toma el software contable de la empresa A, B y un evaluador por perfil.

ETAPA 2: Preparación de la Evaluación:

Paso 1: Se seleccionaron todas las Características, subcaracterísticas y métricas de Calidad, propuestas inicialmente por el modelo para cada el perfil del evaluador, como se muestra en el anexo 4

Paso 2: Definición de los Niveles de Puntaje, donde cada Evaluador definió el peso correspondiente a cada característica y subcaracterística así: (tener en cuenta el anexo 5)

ETAPA 3: Evaluación, se realizaron las valoraciones y se aplicó el modelo para cada producto de la empresa A y B, así:

PRODUCTO: Empresa A
EVALUADOR: Ingeniero 1

Tabla 17. Caso de Estudio Producto A, Evaluador Ingeniero 1

CARACATERISTICA	Peso	SUBCARACTERISTICA	Peso	Calidad Parc - Subcar	Calidad Subcar	Calidad Caract	Calidad Total
<i>Funcionalidad</i>	10	1.1	40	37.54	90.04	9.00	90.92
		1.2	25	21.25			
		1.3	20	17.0			
		1.4	15	14.25			
<i>Fiabilidad</i>	15	2.1	30	25.50	90.00	13.50	
		2.2	40	36.00			
		2.3	30	28.50			
<i>Usabilidad</i>	10	3.1	35	31.50	85.00	8.50	
		3.2	30	22.00			
		3.3	35	31.50			
<i>Mantenibilidad</i>	25	4.1	25	21.67	91.67	22.92	
		4.2	25	22.50			
		4.3	25	25.00			
		4.4	25	22.50			
<i>Portabilidad</i>	20	5.1	40	40.0	95.00	19.00	
		5.2	20	18.0			
		5.3	25	25.0			
		5.4	15	12.0			
<i>Eficiencia</i>	20	6.1	50	50.0	90.00	18.00	
		6.2	50	40.0			

Se puede determinar por los resultados (tabla 17) arrojados por el modelo que la calidad producto corresponde aun 90.92 %, equivalente al Rango de Aceptable o

Satisfactorio. Analizando la Característica Funcionalidad, que obtiene una calidad del 90.04% correspondiente a ese 10% que le dio como peso el evaluador a la mencionada característica, que corresponde al rango de Satisfactorio, y que a su vez este 90.04% corresponde a un 9% del 100% del total de las características analizadas

PRODUCTO: Empresa B
EVALUADOR: Ingeniero 1

Tabla 18. Caso de Estudio Producto B, Evaluador Ingeniero 1

CARACATERISTICA	Peso	SUBCARACTERISTICA	Peso	Calidad Parc - Subcar	Calidad Subcar	Calidad Caract	Calidad Total
<i>Funcionalidad</i>	10	1.1	40	30.15	71.90	7.19	78.88
		1.2	25	18.75			
		1.3	20	11.00			
		1.4	15	12.00			
<i>Fiabilidad</i>	15	2.1	30	22.50	66.00	9.90	
		2.2	40	24.0			
		2.3	30	19.50			
<i>Usabilidad</i>	10	3.1	35	31.50	89.00	8.90	
		3.2	30	26.00			
		3.3	35	31.50			
<i>Mantenibilidad</i>	25	4.1	25	20.00	67.50	16.88	
		4.2	25	17.50			
		4.3	25	15.00			
		4.4	25	15.00			
<i>Portabilidad</i>	20	5.1	40	36.00	80.00	16.00	
		5.2	20	18.00			
		5.3	25	15.00			
		5.4	15	11.00			
<i>Eficiencia</i>	20	6.1	50	50.00	100.0	20.00	
		6.2	50	50.00			

Se puede determinar por los resultados (tabla 18) arrojados por el modelo que la calidad producto corresponde aun 78.88 %, equivalente al Rango de Aceptable. Si analizamos la Característica Funcionalidad esta obtiene una calidad del 71.90% correspondiente a ese 10% que le dio como peso el evaluador a la mencionada característica, que corresponde al rango de Aceptable, y que a su vez este 71.90% corresponde a un 7.19% del 100% del total de las características analizadas

PRODUCTO: Empresa A
 EVALUADOR: Ingeniero 2

Tabla 19. Caso de Estudio Producto A, Evaluador Ingeniero 2

CARACTERISTICA	Peso	SUBCARACTERISTICA	Peso	Calidad Parc - Subcar	Calidad Subcar	Calidad Caract	Calidad Total
<i>Funcionalidad</i>	15	1.1	45	40.85	80.10	12.01	88.52
		1.2	20	15.60			
		1.3	20	13.00			
		1.4	15	11.25			
<i>Fiabilidad</i>	10	2.1	30	22.50	80.00	8.0	
		2.2	40	32.00			
		2.3	30	25.50			
<i>Usabilidad</i>	15	3.1	30	27.00	91.00	13.65	
		3.2	30	28.00			
		3.3	40	36.00			
<i>Mantenibilidad</i>	25	4.1	25	21.25	85.00	21.25	
		4.2	25	22.50			
		4.3	25	20.00			
		4.4	25	22.50			
<i>Portabilidad</i>	25	5.1	40	40.00	94.42	23.60	
		5.2	25	23.75			
		5.3	20	18.67			
		5.4	15	12.00			
<i>Eficiencia</i>	10	6.1	60	60.00	100.0	10.00	
		6.2	40	40.00			

Se puede determinar por los resultados (tabla 19) arrojados por el modelo que la calidad producto corresponde aun 88.52 %, equivalente al Rango de Aceptable. Si analizamos la Característica Funcionalidad esta obtiene una calidad del 80.10% correspondiente a ese 15% que le dio como peso el evaluador a la mencionada característica, que corresponde al rango de Aceptable, y que a su vez este 80.10% corresponde a un 12.01% del 100% del total de las características analizadas

PRODUCTO: Empresa B
 EVALUADOR: Ingeniero 2

Tabla 20. Caso de Estudio Producto B, Evaluador Ingeniero 2

CARACTERISTICA	Peso	SUBCARACTERISTICA	Peso	Calidad Parc - Subcar	Calidad Subcar	Calidad Caract	Calidad Total
<i>Funcionalidad</i>	15	1.1	45	33.92	71.92	10.79	77.99
		1.2	20	15.00			
		1.3	20	11.00			
		1.4	15	12.00			
<i>Fiabilidad</i>	10	2.1	30	22.50	66.00	6.60	
		2.2	40	24.00			
		2.3	30	19.50			
<i>Usabilidad</i>	15	3.1	30	27.00	89.00	13.35	
		3.2	30	26.00			
		3.3	40	36.00			
<i>Mantenibilidad</i>	25	4.1	25	20.00	67.50	16.88	
		4.2	25	17.50			
		4.3	25	15.00			

		4.4	25	15.00		
<i>Portabilidad</i>	25	5.1	40	36.00	81.50	20.38
		5.2	25	22.50		
		5.3	20	12.00		
		5.4	15	11.00		
<i>Eficiencia</i>	10	6.1	60	60.00	100.00	10.00
		6.2	40	40.00		

Se puede determinar por los resultados (tabla 20) arrojados por el modelo que la calidad producto corresponde aun 78.87 %, equivalente al Rango de Aceptable. Si analizamos la Característica Funcionalidad esta obtiene una calidad del 71.90% correspondiente a ese 10% que le dio como peso el evaluador a la mencionada característica, que corresponde al rango de Aceptable, y que a su vez este 71.90% corresponde a un 7.19% del 100% del total de las características analizadas

Una vez terminada la evaluación por medio del modelo propuesto se tiene la siguiente tabla 21.

Tabla 21. Resultados de la Evaluación de calidad para el producto A y B

	<i>Producto A</i>	<i>Producto B</i>
<i>Evaluador 1</i>	90.92	78.87
<i>Evaluador 2</i>	88.52	77.99

Se puede concluir que el Software Contable Producto A de la empresa A, es de una calidad Aceptable o satisfactoria para los evaluadores y de igual manera para el Producto B perteneciente a la empresa B.

La Valoración del Modelo para los productos evaluados A y B, determina que el Producto A posee un nivel de satisfacción más alto que el Producto B.

10.5.1 Análisis de Resultados

Los resultados de la evaluación de calidad realizados por los dos evaluadores a cada uno de los productos de software contable, fueron comparados en este caso de estudio las subcaracterísticas y características de cada producto, perteneciente a cada evaluador, ejemplo, para el evaluador 1 en la tabla 22 así:

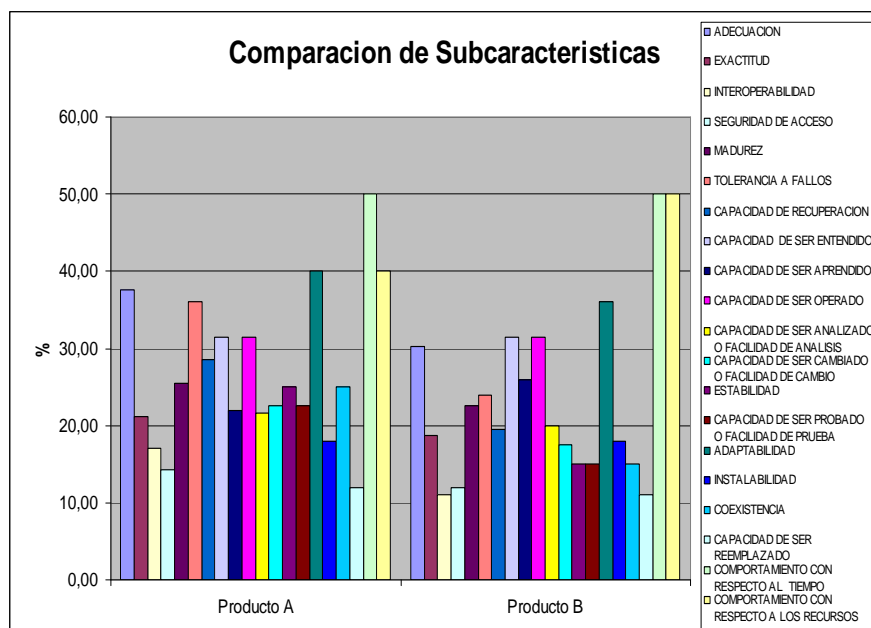
Tabla 22. Resultados de la Evaluación de Calidad de las subcaracterísticas para el producto A y B

SUBCARACTERISTICAS	PRODUCTO A	PRODUCTO B
ADECUACION	37.54	30.15
EXACTITUD	21.25	18.75
INTEROPERABILIDAD	17.0	11.00
SEGURIDAD	14.25	12.00

MADUREZ	25.50	22.50
TOLERANCIA A FALLOS	36.00	24.0
CAPACIDAD DE RECUPERACION	28.50	19.50
CAPACIDAD DE SER ENTENDIDO	31.50	31.50
CAPACIDAD DE SER APRENDIDO	22.00	26.00
CAPACIDAD PARA SER OPERADO	31.50	31.50
CAPACIDAD DE SER ANALIZADO	21.67	20.00
CAPACIDAD DE SER CAMBIADO	22.50	17.50
ESTABILIDAD	25.00	15.00
CAPACIDAD DE SER PROBADO	22.50	15.00
ADAPTABILIDAD	40.0	36.00
INSTALABILIDAD	18.0	18.00
COEXISTENCIA	25.0	15.00
CAPACIDAD DE SER REEMPLAZADO	12.0	11.00
COMPORT CON RESPECTO AL TIEMPO	50.0	50.00
COMPORT CON RESPECTO A LOS RECURSOS	40.0	50.00

Se puede observar que el producto A, posee una capacidad de prestación del servicio en caso de fallo del mismo en un porcentaje del 36% y el producto B, en este mismo ítem es del 24%; además, el Producto A, en el caso que se realicen modificaciones no presenta o evita que se presenten efectos inesperados debidos a las modificaciones realizadas, que es lo que se conoce como Estabilidad con un 25%, cosa que no ocurre con el producto B, que posee un 15%. Si se analiza la Coexistencia en ambos productos, el producto A tiene más capacidad (25%) que el producto B (15%) en compartir recursos comunes dentro del mismo entorno, o sea que esto influye que el Producto A sea más portable que el producto B. Gráficamente lo podemos observar en la Figura 13.

Figura 13. Gráfica de Comparación de Subcaracterísticas Producto A y B



Desde otro punto de vista, se puede apreciar en la Gráfica 14, las subcaracterísticas del mismo producto pero con el punto de apreciación de evaluadores distintos.

En la gráfica Figura 15, con respecto al evaluador Uno, donde la calidad obtenida de acuerdo a la evaluación de Modelo en el producto A, la cual está por encima del 90% (sumando todos los valores obtenidos en cada característica en el mismo producto) y en el producto B esta por debajo del 80%. Además se aprecia como la característica Mantenibilidad en el producto A, está por encima del 22% y muy cercano al peso dado por el evaluador (25%) permite que en dicho producto A, sea fácil realizar una corrección, adaptarlo si el entorno cambia o mejorarlo si el cliente desea un cambio en los requisitos, mientras que el producto B está en esas posibilidades de mantenibilidad por debajo del 20%.

Figura 14. Gráfica de Subcaracterísticas Evaluador 1 y 2

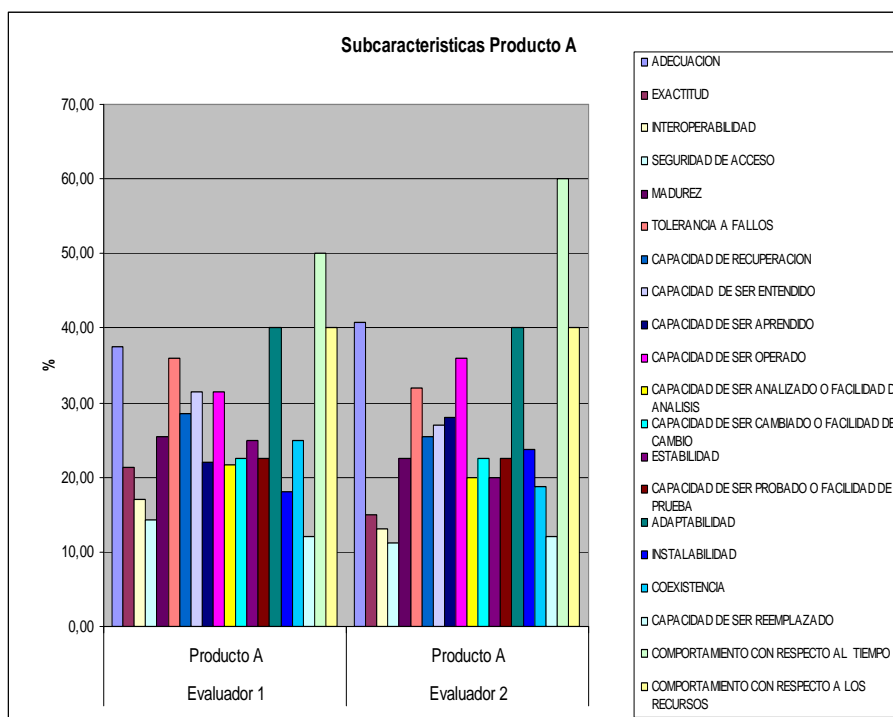
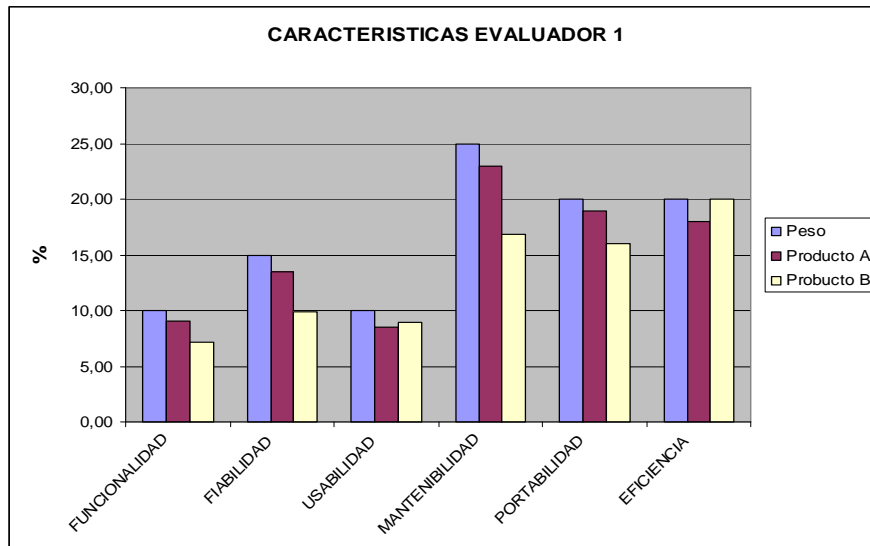


Figura 15. Gráfica de Características Evaluador 1 Producto A y B



Como se puede apreciar el Modelo permite generar una serie de análisis de los productos evaluados con respecto a las características de calidad.

10.6 VALIDACIÓN DEL MODELO

Los datos obtenidos desde el Modelo de Evaluación de Calidad fueron analizados usando el Coeficiente de Correlación^{47 48}, técnica que permite determinar la relación entre dos o más variables, dando un valor cuantitativo. Donde el valor del índice de correlación varía en el intervalo [-1 , +1], donde:

- Si $r = 0$, no existe ninguna correlación. El índice indica, por tanto, una independencia total entre las dos variables, es decir, que la variación de una de ellas no influye en absoluto en el valor que pueda tomar la otra.
- Si $r = 1$, existe una correlación positiva perfecta. El índice indica una dependencia total entre las dos variables denominada relación directa: cuando una de ellas aumenta, la otra también lo hace en idéntica proporción.
- Si $0 < r < 1$, existe una correlación positiva.

⁴⁷ Pita Fernández S. Relación entre variables cuantitativas. Cad Aten Primaria 1997; 4: 141-144.

⁴⁸ Douglas y MARCHAL, William y MASON. Estadística para administración y economía. Alfa omega. Colombia 11ava edición. 2004 Cáp.13 y 14

- Si $r = -1$, existe una correlación negativa perfecta. El índice indica una dependencia total entre las dos variables llamada relación inversa: cuando una de ellas aumenta, la otra disminuye en idéntica proporción.
- Si $-1 < r < 0$, existe una correlación negativa.

Obtenida la Correlación en los dos productos del caso de estudio (tabla 23)

Tabla 23. Validación del Modelo - Coeficiente de Correlación

Producto Software	Coeficiente Correlación r
Producto A	0.95
Producto B	0.94

Los datos generados por el modelo, poseen una correlación positiva, o sea que para este ejemplo de caso de estudio, donde el evaluador asigna pesos diferentes a cada característica de un producto de software que otro evaluador del mismo producto, no lleva a evaluaciones diferentes con respecto a la calidad del producto.

- Además la correlación Positiva indica que los factores (pesos dados por el evaluador o el modelo) de evaluación no llevan a las evaluaciones a resultados diferentes, expresado de otra manera si un evaluador asigna unos pesos a las diferentes características en un producto y otro evaluador le asigna otros pesos al mismo producto los resultados de la evaluación serán proporcionales.

11. CONCLUSIONES

- Se puede determinar que, sí pueden existir modelos que permitan evaluar las diferentes características que afectan la calidad del software de gestión,
- Los modelos de evaluación pueden ser personalizados en actividades de de calidad para distintos producto software y utilizando distintos tipos de proceso para su evaluación, como se puede comprobar en el estado de arte
- Seleccionando los índices de calidad, definiendo niveles de satisfacción, los pesos para el modelo de calidad de las características y subcaracterísticas es posible tener una visión de la calidad del producto software.
- El estándar ISO 9126 define un modelo de calidad del software en el que la calidad se define como la totalidad de características relacionadas con su habilidad para satisfacer necesidades establecidas o implicadas. Además en este estándar se establece que cualquier componente de la calidad del software puede ser descrito en términos de algunos aspectos de una o más de estas seis características.
- La calidad de un producto software, como lo es un software de gestión puede evaluarse considerando una serie de características siguiendo normas ISO.
- El modelo propuesto fue probado a través de un estudio de campo que permitió medir el desempeño de este modelo objetivamente en dos productos software contables.
-

11.1 LOGROS ALCANZADOS

- Realizar una investigación de la cual hicieron participes Ingenieros de sistemas, Contadores y Administradores permitiendo de esta forma realizar un trabajo interdisciplinario que enriqueció el quehacer de los participantes.
- Como resultado del proceso de investigación se ha podido verificar que existe una gran necesidad por la mejora de la calidad del software contable en el entorno del sector público.

- Con la realización de la investigación se aportó a las personas que intervinieron en la formulación, aplicación del modelo y que pertenecen a la Dirección de la empresa y toman decisiones informáticas, en la obtención de conocimientos informáticos de calidad, que de una u otra manera ayudan a la adquisición de un producto de software contable de calidad.
- Demostrar a los estudiantes de ingeniería de sistemas, que la profesión no solo es programación y computadores, sino que existen diversas actividades como la Planeación, la organización, integración, dirección y el control (fases de un proceso administrativo), que para una simple adquisición de un software que requieren de todo ese conjunto de actividades y con el conocimiento de otros profesionales se puede llegar a obtener un producto software de calidad.
- El marco presentado en esta investigación proporciona a las empresas un modelo de evaluación adaptado a las características de alta calidad, para que su producto de software sea competitivo y de calidad.

11.2 APRENDIZAJE OBTENIDO

- La Calidad del software puede medirse después de elaborado el producto. Pero este puede resultar muy costoso si se detectan problemas derivados de imperfecciones en el diseño, por lo que es imprescindible tener en cuenta tanto la obtención de la calidad como su control durante todas las etapas del ciclo de vida del software.
- En la actualidad las empresas se enfrentan con crecientes exigencias en términos de calidad de los clientes y con niveles de competencia mundial sin precedente alguno. Luego, es muy importante tener la capacidad para garantizar a los clientes internos y externos que los productos que se realicen o se adquieran satisfacen y cubren las expectativas de ellos mismos. Por lo que no es sólo la manera en que el proveedor pueda obtener ventaja sobre sus competidores, sino porque la calidad se está transformando rápidamente en un requisito esencial para el mundo.
- Como todo proceso investigativo lleva a descubrir y generar nuevos elementos o necesidades; y que partiendo de una investigación que se realiza, de esta se pueden derivar varias investigaciones, como generar otros modelos de calidad para otros tipos de software u otros tipos de aplicaciones distintos al software; por lo tanto, se puede concluir que la investigación genera más investigación.

11.3 FUTUROS TRABAJOS

- Continuar con la investigación por parte de alumnos de pregrado en la obtención de una herramienta que facilite la funcionalidad del modelo.
- Como futuro trabajo de investigación se puede considerar continuar el desarrollo del modelo, para que cada día sea perfeccionado más y más; adecuar el modelo a las normas derivadas del estándar ISO/IEC 9126-1 a las subsiguientes normas como son las 9126-2, 9126-3 y 9126-4.
- Realizar algunas modificaciones básicas al modelo en cuanto a definición de métricas de calidad de cada una de las subcaracterísticas para la evaluación de otros tipos de software. Por lo que se puede determinar si al que dichas modificaciones pueden generar nuevas investigaciones.

11.3.1 Recomendaciones

- Realizar un análisis académico para que se tenga como electiva tanto en pregrado como en posgrado un espacio académico para el análisis, estudio y discusión sobre la Calidad en el Software desde todas sus perspectivas.
- Promover este modelo para que pueda ser adaptado a cualquier tipo de software y por que no de procesos distintos al software que cumplan con las características de alta calidad presentadas en el estándar.
- El modelo se puede aplicar y/o adecuar para generar una guía para la evaluación de proyectos de grado que impliquen una aplicación software.

REFERENCIAS

- [1] Asociación Colombiana de Ingenieros (ACIS). Calidad del Software. Revista Sistemas Nro. 72. Julio a Septiembre de 1997: Santafé de Bogotá.
- [2] Weyuker, E.J. (1988). Evaluating software complexity measures.
- [3] IEEE Transactions on Software Engineering. 14(9). pp.1357-1365."
- [4] Cruz Ramírez, José, "Historia de la calidad" en EXCELLENTIA, pp. 8-14.
- [5] Jeffrey L. Whitten, Análisis y diseño de sistemas de información. Tercera Edición. Pag. 172.
- [6] Catalogo de Software y Guía de Servicios Informáticos. 15ª Edición 2006. Guía completa de información de los principales productos, servicios y empresas proveedoras en Colombia.
- [7] Jramonet@cca.ictnet.es, accesado Junio 25 de 2005
- [8] Contraloría Departamental del Quindío
- [9] Análisis y diseño de sistemas de información. Tercera Edición Jeffrey L. Whitten.
- [10] DUTSCH S. Maichael. Software Quality Engineering. PrenticeHall 1998
- [11] Quality Assurance Resources Online. <http://w.w.w.quality.org/qc/homcpage/html>
Fecha de Consulta: Junio de 2006.
- [12] Quesada M Gilberto, Presidente Grupo Kaizen S.A., Presidente Grupo Kaizen. <http://www.grupokaizen.com>. Fecha de Consulta: Junio de 2006.
- [13] Gluch, David; Comella-Dorda, Santiago; Hudak John; Lewis, Grace; Walter, John; Weinstock, Charles & Zubrow, David. Model-Based Verification: An Engineering Practice. Software Engineering Institute (SEI). 2002. [Versión electrónica].
- [14] CARDENAS Tangarife José Joaquin. Calidad en Ingeniería del Software. Empresa Nacional de Telecomunicaciones Telecom. Santafé de Bogotá 1998.
- [15] An experiment measuring the effects of personal software process (PSP) training - Software Engineering, IEEE, Vol. 27, No. 5, May 2000.
- [16] Seminario Calidad en el Desarrollo y Contratación. ITC Telecom 1998.

- [17] Journal QUALITY Software Msg. Ron J West IEEE 29-35, 1999.
- [18] Software Engineering Standards Collection, 1991, IEEE. Guide for Software Quality Assurance Planning. Software Engineering Standards Collection”,1991. IEEE. Página 1-46.
- [19] VANESSA Hamar; “Aspectos metodológicos del desarrollo y reutilización de componentes de software” Tesis de Grado, Universidad de los Andes, Mérida-Venezuela 2003.
- [20] GOMEZ, Muñoz Adriana, Adquisición de Software en la Universidad de los Andes guía metodológica”, Tesis de grado, Universidad del los Andes, Santafé de Bogota, 2002.
- [21] Franch, X.; Carvallo, J.P. “Using quality models in software package selection”, Software, IEEE Volume 20, Issue 1, Jan/Feb 2003 Page(s): 34 - 41
- [22] Losavio Francisca. “Quality Models to Design Software Architecture”.Central University of Venezuela, Caracas, Venezuela. Journal of Object Technology, Vol. 1, no. 4, September-October 2002.
- [23] Losavio Francisca, Chirinos Iedy, Levy Nicole. “Quality Characteristics for Software Architecture”. Central University of Venezuela, Caracas, Venezuela. Université de Versailles St.-Quentin France. Journal of Object Technology, Vol. 2, No.2, March-April 2003.Fuente http://www.jot.fm/issues/issue_2003_03/article2.pdf
- [24] Bertoa Manuel F. y Vallecillo Antonio. “Atributos de Calidad para Componentes COTS”. Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga, España. Fecha de Consulta: Octubre de 2007.
www.lcc.uma.es/~av/Publicaciones/02/AtributosCalidadCOTS.pdf
- [25] Franch, X.; Carvallo, J.P. “Un Catálogo de Factores de Calidad para la Definición de Requisitos No- Técnicos en la Selección de Componentes COTS. http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER06/carvallo.pdf
- [26] Franch, X.; Carvallo, J.P. “A Quality-Model-Based Approach for Describing and Evaluating Software Packages”. fuente Base de datos IEEE de la UIS. Universidad Politécnica de Cataluña, Proceedings of the IEEE Joint International Conference on Requirements Engineering 2002 IEEE
- [27] Dromey R.G.“A model for software product quality”, Software Quality Institute.Griffith University. Fuente Base de Datos IEEE UIS.
- [28] Condory – Fernandez Nelly, Belenguer F. Jorge, Albiol Manoli. “Modelo de Agregación Basado en un Sistema Neurodifuso para un proceso de evaluación de calidad de software”. Universidad Nacional de San Agustín, Perú. Universidad

Politécnica de Valencia. Fecha de Visita: Diciembre de 2006
http://oomethod.dsic.upv.es/anonimo/..%5Cfiles%5CInConferenceArticle%5C2002_CLEI.pdf

[29] Castillo V., Juan de Jesús, Pedraza Julio. Conferencias sobre metodología de la Investigación, Universidad del Quindío y UIS.

[30] Dawson W. Christian. El Proyecto Fin de Carrera en Ingeniería Informática Una Guía para el Estudiante. Prentice Hall . 2002.

[31] PRESSMAN S. Roger. Ingeniería del Software. Quinta edición. Editorial Macgraw Hill. España 2003.

[32] Asociación Colombiana de Ingenieros (ACIS). Calidad del Software. Revista Sistemas Nro. 72. Julio a Septiembre de 1997: Santafé de Bogotá.

[33] PINILLA Forero José Dagoberto. Auditoría Informática. Editorial Presencia Ltda. Santafé de Bogotá Colombia.

[34] <http://www.rspa.com>, <http://www.spc.ca/spe/metrvr.htm>, <http://www.pressman5.com>. Fecha de Consulta: Junio de 2005.

[35] CALIDAD en Proyectos Informáticos – <http://lisisu02.usal.es/-moreno/API Tema5.pdf>

[36] PIATTINI G. Mario, Calvo José A. Análisis y Diseño de Aplicaciones Informáticas de Gestión. Alfaomega, 2004, pp 476-508.

[37] Grupo Alarcos ESI – UCLM. Calidad del Producto. Grupo Alarcos ESI – UCLM. Fecha de Consulta: Marzo de 2006, disponible en <http://www.alarcos.inf-uclm.es/doc/Calidad/capitulo05.ppt>

[38] Scalone Fernanda; Estudio Comparativo de los modelos y estándares de Calidad del Software. Tesis Maestría, Universidad Tecnológica Nacional de Buenos Aires. 2006.

[39] International Organization For Standardization (ISO) 2-382/8. “Control, Integridad y Seguridad”

[40] ISO/IEC, ISO/IEC 9126-1:2001 Software Engineering – Product Quality, Part 1: Quality Model, Secretaría General de ISO, Ginebra, 2001

[41] ISO/EIC IS 9126-1 “Information Technology-Software Product Evaluation: quality characteristics and guidelines for their use”,2001

[42] Calidad del producto ISO/IEC, ISO/IEC 9126-1:2001.
<http://kybele.escet.urjc.es/documentos/GCSW/GYCSW%20-%20Tema%204%20%20Calidad%20de%20Producto.pdf>. Enero de 2006.

- [43] SABINO Carlos A. El proceso de la Investigación. Editorial El Cid. Bogotá. 1989
- [44] BRAUDE, E. Ingeniería de Software. Una perspectiva orientada a Objetos. AlfaOmega, 2000
- [45] JEFFREY L. Whitten. Análisis y Diseño de Sistemas de Información. Tercera edición. Editorial Irwin. España 1996. – ALARCON Raúl. Diseño Orientado a Objetos con UML. (Versión electrónica).
- [46] Osorio Rojas Ricardo Arturo, Profesor, La Escala de Likert. Fecha de Consulta: Junio de 2007, Disponible en <http://www.nodo50.org/sindpitagoras/Likert.htm>.
- [47] Pita Fernández S. Relación entre variables cuantitativas. Cad Aten Primaria 1997; 4: 141-144.
- [48] Douglas y MARCHAL, William y MASON. Estadística para administración y economía. Alfa omega. Colombia 11ava edición. 2004 Cáp.13 y 14
- [49] Zuse, H., 1998, "A Framework of Software Measurement", Walter de Gruyter, Berlín-NY.
- [50] Coral – Gudiño. Contabilidad Universitaria. Mc Graw Hill. Cuarta Edición 2001.
- [51] Díaz Hernando. Contabilidad con Aplicaciones Informáticas. Prentice Hall, 2000.
- [52] IEEE, IEEE Std 1061-1992. Standard for a Software Quality Metrics Methodology, IEEE, 1992.
- [53] García Romero Claudia Ivette, El Modelo de Capacidad de Madurez y su Aplicación en Empresas Mexicana de Software. Tesis profesional. http://mailweb.udlap.mx/~tesis/lis/garcia_r_ci/capitulo5.html. 16-05-2005.
- [54] Boeg J. DePanfilis S., Pasquini A. A Method for software Quality Planning, Control and Evaluation, 66-77, IEEE Software, Marzo/ Abril 1999,.
- [55] JOURNAL An Integrated Model of ISO 9001:2000 and CMMI for ISO Registered Organizations Chanwoo Yoo1, Junho Yoon1, Byungjeong Lee2, Chongwon Lee1.
- [56] JOURNAL An Experiment Measuring the Effects of Personal Software Process (PSP) Training Lutz Prechelt, Member, IEEE, and Barbara Unger, Member IEEE transactions on software engineering, Vol. 27, no. 5, february 2000.
- [57] JOURNAL Identifying Extensions Required by RUP (Rational Unified Process) to Comply with CMM (Capability Maturity Model) Levels 2 and 3. IEEE transactions on software engineering, Vol. 29, no. 2, february 2003
- [58] JOURNAL S..L.Pfleeger, N.Fenton y S.Page "Evaluating Software Engineering Standards", COMPUTER pp.71-79, Sept.1994.

[59] KENDALL Kenneth E. y Kendall E. Julie. Análisis y Diseño de Sistemas. Tercera edición. Editorial Prentice Hall. México. 1997.

[60] PFLEEGER, S. Ingeniería de software. Prentice-Hall. 2002

[61] PSP. The Personal Software Process. Software Engineering Institute (SEI). 2003. [Versión electrónica] <http://www.sei.cmu.edu/tsp/psp.html>

[62] QUERUBIN Lóndño Rodrigo. Asociación Colombiana de Ingenieros (ACIS). Guías Generales Sobre Desarrollo de Software. Primera edición. Santafé de Bogotá 1996.

[63] SENN. Análisis y Diseño de Sistemas de Información. Editorial Macgraw Hill. México

[64] Capability Maturity Model. Fecha de Consulta: Febrero de 2006.
http://en.wikipedia.org/wiki/Capability_Maturity_Model.

[65] Características de calidad "Iltities" <http://en.wikipedia.org/wiki/Iltities>. Fecha de Consulta: Marzo de 2007.

[66] Lisandra V. Manzoni and Roberto T. Price, Member, IEEE. IEEE transactions on software engineering, vol. 29, no. 2, february 2003.

[67] Deutsch Michel S., Willis Ronald R.; "Software Quality Engineering, A total Technical and Management Aproach", Prentice Hall, pp 45-98

[68] Ingeniería de Software en la WEB.
http://triton.uniandes.edu.co:5050/dspace/bitstream/1992/344/1/mi_1113.pdf. Marzo 2006.

[58] M. Azuma "SquaRE: the next generation of the ISO/IEC 9126 and 14598 international standards series on software product quality". In ESCOM (European Software Control and Metrics conference), April 2001.

[69] Chung et al., 2000] Chung, L., Nixon, B., Yu, E., and Mylopoulos, J., "Non-Functional Requirements in Software Engineering". Kluwer Academic Publisher, 2000.

[70] Carnegie Mellon. Software Engineering Institute - Capability Maturity Model[®] for Software (SW-CMM[®])

[71] www.sei.cmu.edu/cmm/cmm.html. Visita 04- 28-2005

[72] www.sei.cmu.edu/publications/documents/94.reports/94.tr.012.html.
Visita 04- 28-2005

[73] www.sei.cmu.edu/pub/documents/94.reports/pdf/tr12.94.pdf. Visita 04- 28-2005

[74] www.emprendedor.com/Iso9000/ Visita 14-05-2005

[75] www.iso.ch/iso/en/ISOOnline.frontpage Visita 14-05-2005.

[76] www.aenor.es/iso9000.htm. Visita 14-05-2005

[77] www.isoeasy.org/ Visita 14-05-2005

[78] www.homoqualitas.com/castella/infos/iso90002000/portada.htm Visita 14-05-2005

ANEXOS

ANEXO 1- ORGANIZACIONES

Las organizaciones que han desarrollado los estándares son:

AECL	Atomic Energy of Canada Limited	AIAA	American Institute of Aeronautics and Astronautics
ANS	American Nuclear Society	ANSI	American National Standards Institute
APWA	American Public Works Association	AS	Standards Association of Australia
ASME	American Society of Mechanical Engineers	ASTM	American Society for Testing and Materials
ATA	Air Transport Association of America	BCS	British Computer Society
BSI	British Standards Institution	CCITT	Comité Consultatif International Telegraphique et Telephonique
CEN	Comité Européen de Normalisation	CSA	Canadian Standards Association
DEF	British Defence Standards	DIN	Deutsches Institut für Normung
DOD	Defense Materiel Specifications and Standards Office United States Department of Defense	DTI	Department of Trade and Industry
ECMA	European Computer Manufacturers Associations	EIA	Electronic Industries Association
EPRI	Electric Power Research Institute	ESA	European Space Agency
FAA	Federal Aviation Authority	FDA	Federal Food and Drug Administration
GFMT	German Federal Ministry of Transport	GPA	Gas Processors Association
IAEA	International Atomic Energy Agency	IEC	International Electrotechnical Commission
IEE	British Institution of Electrical Engineers	IEEE	Institute of Electrical and Electronics Engineers
ISA	Instrument Society of America	ISO	International Organization for Standardization
KBSt	Bundesministerium des Innern	NASA	National Aeronautics and Space Administration
NATO	North Atlantic Treaty Organization	NISO	National Information Standards Organization
NIST	National Institute of Standards and Technology	NMTBA	National Machine Tool Builders Association
NRC	Nuclear Regulatory Commission	RTCA	Radio Technical Commission for Aeronautics, Inc.
SAE	SAE International	SEI	Software Engineering Institute
SIS	Standardiseringskommissionen y Sverige	STRI	Icelandic Council for Standardization
UL	Underwriters Laboratories, Inc		

Tabla 9.Organizaciones que han definido estándares

ANEXO 2- CATALOGO DE ESTÁNDARES.

AÑO	ORG	CODIGO	NOMBRE	CAT	TIPO	Req	Dis	Cod	Pro	VV	Cnf	Cal	Otros
1966	ECMA	ECMA-4	FLOW CHARTS	E	prod		*						*
1974	NIST	FIPS PUB.30	SOFTWARE SUMMARY FOR DESCRIBING COMPUTER PROGRAMS AND AUTOMATED DATA SYSTEMS	E	prod								*
1974	NIST	FIPS- PUB.30	SOFTWARE SUMMARY FOR DESCRIBING COMPUTER PROGRAMS AND AUTOMATED DATA SYSTEMS	E	prod								*
1976	NIST	FIPS PUB.38	GUIDELINES FOR DOCUMENTATION OF COMPUTER PROGRAMS AND AUTOMATED DATA SYSTEMS	G	prod								*
1979	NIST	FIPS PUB.64	GUIDELINES FOR DOCUMENTATION OF COMPUTER PROGRAMS AND AUTOMATED DATA SYSTEMS FOR THE INITIATION PHASE	G	prod								*
1979	IEC	643	APPLICATION OF DIGITAL COMPUTERS TO NUCLEAR REACTOR INSTRUMENTATION AND CONTROL	G	prod	*							*
1979	CSA	CAN/CSA-Z243.15.1	BASIC GUIDELINES FOR THE STRUCTURE OF DOCUMENTATION OF COMPUTER BASED SYSTEMS	G	proc prod								*
1980	DOD	NBSIR 79-1940 (R)	INTEGRATED COMPUTER AIDED MANUFACTURING (ICAM) SOFTWARE DOCUMENTATION STANDARDS	E	prod								
1981	EPRI	NSCAC-39	VERIFICATION AND VALIDATION FOR SAFETY PARAMETER DISPLAY SYSTEMS	R	prod					*			
1981	EIA	CMB4-3	COMPUTER SOFTWARE LIBRARIES	G	proc prod						*		
1981	ANSI	ANSI X3.88	AMERICAN NATIONAL STANDARD FOR COMPUTER PROGRAM ABSTRACTS	E	prod								
1981	BSI	BS 6154	METHOD OF DEFINING SYNTACTIC METALENGUAGE	E	prod								
1981	DOD	IDS 1501200001	INTEGRATED COMPUTER AIDED MANUFACTURING (ICAM) DOCUMENTATION STANDARDS	E	prod								
1981	NATO	NATO AQAP-13	SOFTWARE QUALITY CONTROL SYSTEM REQUIREMENTS	E	proc, prod				*		*	*	
1982	BSI	BS 6238	PERFORMANCE MONITORING OF COMPUTER-BASED SYSTEMS	G	proc								
1983/91	IEEE	829	SOFTWARE TEST DOCUMENTATION	E	prod					*			
1983	ISA	ISA S5.3	GRAPHIC SYMBOLS FOR DISTRIBUTED CONTROL/SHARED DISPLAY INSTRUMENTATION, LOGIC AND COMPUTER SYSTEMS	E	prod								*
1983	NIST	FIPS PUB.99	GUIDELINE: A FRAMEWORK FOR THE EVALUATION AND COMPARISON OF SOFTWARE DEVELOPMENT TOOLS	R	proc prod								*
1983	NIST	FIPS PUB.101	GUIDELINE FOR LIFE CYCLE VALIDATION, VERIFICATION AND TESTING OF COMPUTER SOFTWARE	G	proc prod					*			*
1984	DEF	00-16/ISS.1	GUIDE TO THE ACHIEVEMENT OF QUALITY IN SOFTWARE	G	proc		*	*	*		*		
1984	ASTM	E 1029	GUIDE FOR DOCUMENTATION OF CLINICAL LABORATORY COMPUTER SYSTEMS	G	prod								
1984	DOD	MIL-STD-882B	SYSTEM SAFETY PROGRAM REQUIREMENTS	E	proc		*		*				

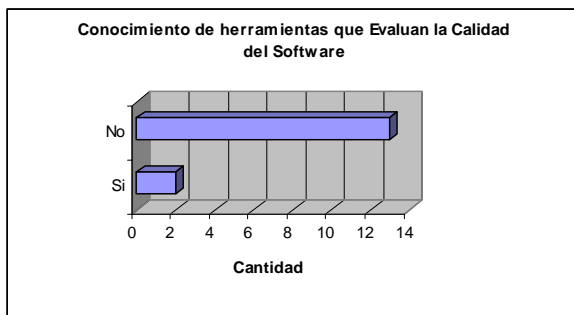
1988	ASTM	E 627	GUIDE FOR DOCUMENTING COMPUTERIZED SYSTEMS	G	proc													
1988	ASTM	E 1246	PRACTICE FOR REPORTING RELIABILITY OF CLINICAL LABORATORY COMPUTER	E	proc													
1988	DOD	MIL-STD-1803	SOFTWARE DEVELOPMENT INTEGRITY PROGRAM (SDIP)	E	proc	*	*	*		*								
1988	IEEE	982.1	DICTIONARY OF MEASURES TO PRODUCE RELIABLE SOFTWARE	E	rec													
1988/93	IEEE	1028	SOFTWARE REVIEWS AND AUDITS	E	proc					*								
1988	FDA	DS-9	REVIEWER GUIDANCE FOR COMPUTER-CONTROLLED MEDICAL DEVICES	G	proc	*	*	*	*	*	*							
1988	DOD	AFSC/AFLCP 800-45	ADQUISITION MANAGEMENT SOFTWARE RISK ABATEMENT	G	proc					*								
1988	IEEE	982.2	GUIDE FOR THE USE OF IEEE STANDARD DICTIONARY OF MEASURES TO PRODUCE RELIABLE SOFTWARE	G	proc													
1988	DOD	STD-7935A	DOD AUTOMATED INFORMATION SYSTEMS (AIS) DOCUMENTATION STANDARDS	E	prod													
1988/93	IEEE	1058.1	SOFTWARE PROJECT MANAGEMENT PLANS	E	prod					*								
1988	ISO	ISO 9127	INFORMATION PROCESSING SYSTEMS - USER DOCUMENTATION AND COVER INFORMATION FOR CONSUMER SOFTWARE PACKAGES	E	prod													*
1988	NIST	S.P. 500-154	GUIDE TO DISTRIBUTED DATABASE MANAGEMENT	R	proc prod	*	*											
1988	NIST	S.P. 500-155	MANAGEMENT GUIDE TO SOFTWARE REUSE	R	proc													*
1989	ASTM	E 623	GUIDE FOR DEVELOPING FUNCTIONAL REQUIREMENTS FOR COMPUTERIZED SYSTEMS	G	proc prod	*	*											
1989	CSA	CAN/CSA-Q396.1.1	SOFTWARE QUALITY ASSURANCE PROGRAM FOR THE DEVELOPMENT OF SOFTWARE USED IN CRITICAL APPLICATIONS	E	proc													
1989	CSA	CAN/CSA-Q396.1.2	SOFTWARE QUALITY ASSURANCE PROGRAM FOR PREVIOUSLY DEVELOPED SOFTWARE USED IN CRITICAL APPLICATIONS	E	proc													
1989	CSA	CAN/CSA-Q396.2.1	SOFTWARE QUALITY ASSURANCE PROGRAM FOR THE DEVELOPMENT OF SOFTWARE USED IN NONCRITICAL APPLICATIONS	E	proc													
1989	CSA	CAN/CSA-Q396.2.2	SOFTWARE QUALITY ASSURANCE PROGRAM FOR PREVIOUSLY DEVELOPED SOFTWARE USED IN NONCRITICAL APPLICATIONS	E	proc													
1989	DOD	TAD STAND E	SOFTWARE DEVELOPMENT, DOCUMENTATION AND TESTING POLICY FOR NAVY MISSION CRITICAL SYSTEMS	E	proc					*								
1989/93	IEEE	1063	SOFTWARE USERS DOCUMENTATION	E	prod													
1989	ASTM	E 1283	GUIDE FOR PROCUREMENT OF COMPUTER-INTEGRATED MANUFACTURING SYSTEMS	G	proc	*	*											
1989	CCITT	Z.100	FUNCTIONAL SPECIFICATION AND DESCRIPTION LANGUAGE (SDL) CRITERIA FOR USING FORMULA DESCRIPTION TECHNIQUES (FDTs)	E	prod		*	*										

1990	ISO	ISO TR 9294	INFORMATION PROCESSING - GUIDELINES FOR THE MANAGEMENT SOFTWARE DOCUMENTATION	R	rec					*						*
1990	NIST	S.P. 500-180	GUIDE TO SOFTWARE ACCEPTANCE	R	proc						*					
1991	AS	AS 3563.1	SOFTWARE QUALITY MANAGEMENT SYSTEM: REQUIREMENT	E	proc											*
1991	AS	AS 3563.2	SOFTWARE QUALITY MANAGEMENT SYSTEM: IMPLEMENTATION GUIDE	G	proc											*
1991	EIA	CMB7-1	ELECTRONIC INTERCHANGE OF CONFIGURATION MANAGEMENT DATA	R	proc								*			
1991	ESA	PSS-05-0 ISS.2	SOFTWARE ENGINEERING STANDARDS	E	proc prod	*	*	*	*	*	*	*	*	*	*	*
1991	EIA	CMB7-2	GUIDELINE FOR TRANSITIONING CONFIGURATION MANAGEMENT TO AN AUTOMATED ENVIRONMENT	R	proc								*			
1991	ASTM	E 1384	GUIDE FOR DESCRIPTION FOR CONTENT AND SCTRURE OF ALL AUTOMATED PRIMARY RECORD OF CARE.	G	prod	*	*									*
1991	ECMA	TR/55	REFERENCE MODEL FOR FRAMEWORKS OF SOFTWARE ENGINEERING ENVIRONMENTS	R	prod											*
1991	ISO	ISO 9000-3	QUALITY MANAGEMENT AND QUALITY ASSURANCE STANDARDS - PART 3, GUIDELINES FOR THE APPLICATION OF ISO 9001 TO THE DEVELOPMENT, SUPPLY, AND MAINTENANCE OF SOFTWARE	G	proc											*
1991	ISO	ISO 9126	INFORMATION TECHNOLOGY - SOFTWARE PRODUCT EVALUATION - QUALITY CHARACTERISTICS AND GUIDELINES FOR THEIR USE	E	proc											*
1991	NASA	NASA STD-2100	SOFTWARE DOCUMENTATION STANDARD SOFTWARE ENGINEERING PROGRAM	E	prod											*
1991	NATO	STANAG 4159	MATERIAL CONFIGURATION MANAGEMENT POLICY AND PROCEDURE FOR MULTINATIONAL JOINT PROJECTS	E	proc								*			
1991	NISO	ANSI/NISO Z39.67	COMPUTER SOFTWARE DESCRIPTION	E	prod											*
1992	AIAA	AIAA R-023	RECOMMENDED PRACTICE FOR HUMAN-COMPUTER INTERFACES FOR SPACE SYSTEM OPERATIONS	G	prod	*	*									*

ANEXO 3 - ENCUESTAS

Los Resultados de la Encuesta a los Ingenieros de Sistemas se analizan a Continuación:

Pregunta 1: Conocimiento de herramientas que evalúen la Calidad del software.



Como se observa 12 de los 15 ingenieros encuestados desconocen herramientas que evalúen la calidad del software. 2 de los 15 conocen herramientas como:

- Programa Junior: verifica Técnicas de programación
- Modulo Bensmarch: Mide rendimiento de software
- Herramientas en el ámbito de Auditoría.

Lo anterior significa que la mayoría de los Ingenieros de Sistemas desconocen herramientas que cumplan con dicha función, .

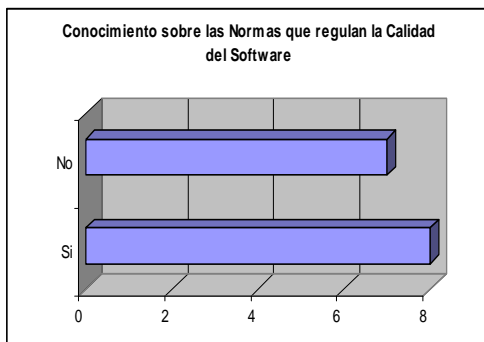
Los Ingenieros de Sistemas que afirman tener conocimiento de dichas herramientas, no están orientados a evaluar la calidad del software de Gestión.

Pregunta 2: Criterios que determinan la calidad del Software.

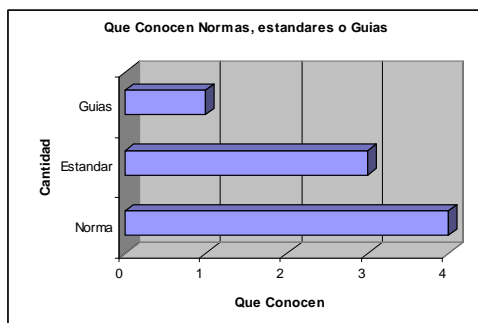
De un total de 15 Ingenieros de Sistemas, todos respondieron expresando varios criterios, así:

<i>Criterios que determinan la Calidad del Software</i>	<i>Cantidad</i>
Interfaz amigable	10
Funcionalidad	12
Seguridad	7
Soporte	8
Tiempo de respuesta	10
Modularidad en el diseño	5
Proceso de desarrollo de software	3
Confiabilidad	7
Flexibilidad	2
Mantenimiento	5
Código fuente	2
Escalable	1
Compatibilidad	1
Portabilidad	1
Recurso de Máquina	5
Integridad	10
Operable	12
Tolerante a errores	15
Recuperable	10
Fácil aprendizaje	9
Capacidad de Ampliación	8
Costos	15
Adecuado	9
Capacidad para ser entendido y aprendido	7
Estabilidad	8
Productivo	9
Fácil Instalación	10

Pregunta 3: Conocimiento sobre las normas, estándares, guías que regulan la calidad del software.



De los 15 Ingenieros de Sistemas, 8 conocen normas que regulan la calidad del software y 7 las desconocen, pero de los 8 que conocen sobre normatividad estos conocen de Normas, estándares y guías así:

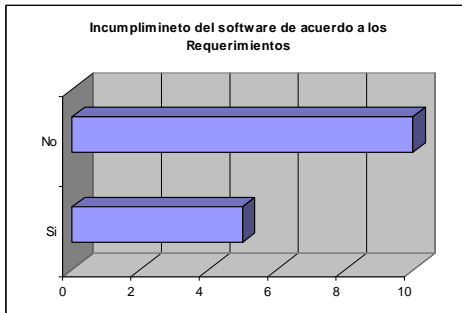


Pregunta 4: Deficiencias del Software de Gestión.

De un total de 15 ingenieros, todos respondieron expresando varios criterios, así:

<i>Deficiencias del Software de Gestión</i>	<i>Cantidad</i>
Soporte	8
Seguridad	5
Sistemas cerrados	6
Modularidad en el diseño	5
Interfaz Funcionalidad	5
Integridad	4
Costos	2
Diseño	2
Defectos	2
Análisis	1
Mantenimiento	1
Portabilidad	9
Corrección	6
Usabilidad	8
Dificultad en Auditoria	7
Escalabilidad	7
Reportes o Informes	3

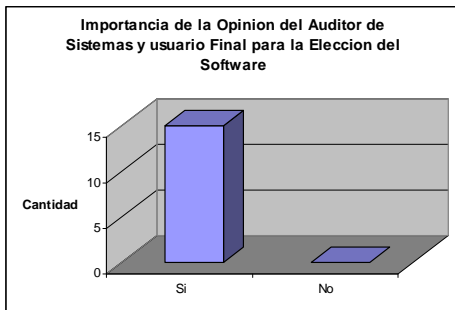
Pregunta 5: Incumplimiento del software de acuerdo a los requerimientos.



De los 15 Ingenieros de sistemas, 5 aceptan que han adquirido software que no cumple con los requerimientos establecidos y 10 opinaron haber adquirido software que cumple con los requerimientos.

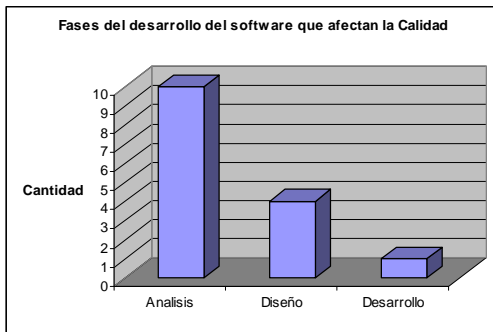
Esto significa que el software de gestión que se desarrolla actualmente no se adapta a las necesidades reales del usuario. Lo cual demuestra la necesidad del usuario de contar con un modelo que le permita evaluar el cumplimiento de requerimientos.

Pregunta 6: Importancia de la opinión del Auditor de Sistemas y el Usuario final, para la elección del software.



De los 15 Ingenieros encuestados, respondieron que era muy importante la opinión del auditor de sistemas y en especial del usuario final para la elección del software. Lo anterior significa que el Ingeniero de Sistemas considera determinante la opinión del auditor de sistemas y usuario final para la elección de un software de gestión.

Pregunta 7: fases del desarrollo del software que afectan la calidad del software.



De los 15 Ingenieros encuestados, 10 consideran que la fase que más afecta la Calidad del Software ya sea de gestión o de otro tipo es la fase de Análisis.

Los Resultados de la Encuesta a los Usuarios Finales, estas determinaron que entre los criterios que consideran más relevantes en una buena calidad del software de gestión están:

- El software debe tener facilidad de uso.
- Facilidad de Corrección
- Facilidad mantenimiento
- Facilidad para los procesos de Auditoria
- Ser flexible
- Ser fiable
- Ser seguro en su información y el programa
- Modular

- Se deben utilizar en lo posible interfaces amigables
- Distribución de la información, de tal manera que esta aparezca en las mismas zonas
- El programa debe entenderse fácilmente.
- Garantizar la fiabilidad de los datos
- Velocidad de respuesta
- Cumplimiento adecuado de los requerimientos de usuario
- Variedad y facilidad de interpretación en la presentación de reportes e informes
- Control en la entrada de datos no validos
- Manuales de usuario sencillos
- No deben existir bloqueos frecuentes en el sistema.

ANEXO 4 – CARACTERÍSTICAS, SUBCARACTERÍSTICAS Y MÉTRICAS

Para un mejor manejo en el Modelo de las Características, Subcaracterísticas y métricas, estas se codificaran de la siguiente Manera:

1. FUNCIONALIDAD

1.1 Adecuación

- 1.1.1 Permite Identificar y definir la empresa (nombre, actividad, Nit, régimen)
- 1.1.2 Define el plan de cuentas campo por campo (código, suma de la cuenta, Nombre, Nivel, Tercero, Base, centro, Presupuesto).
- 1.1.3 El plan de cuentas se adecua a las nuevas necesidades de la empresa o de normatividad legal.
- 1.1.4 Define documentos a Utilizar (comprobantes, Consignaciones, Recibos, Facturas, Notas, provisiones)
- 1.1.5 Consulta inmediata de información contable a cualquier Nivel
- 1.1.6 Capacidad de Búsqueda por Cualquier campo Fecha, NIT, Código, cuenta, concepto
- 1.1.7 Permite Introducir la Información inicial requerida, con base en los saldos del último balance
- 1.1.8 Permite la Preparación de Soportes Contables (notas contables, recibos de caja, comprobantes de egreso, factura de venta, total crédito y debito)
- 1.1.9 Permite Realizar provisiones sociales de acuerdo a la Nomina
- 1.1.10 Permite la creación y asiento de los distintos tipos de ajuste
- 1.1.11 Genera los distintos reportes de Consolidación de la Información Contable
- 1.1.12 Genera los 5 estados financieros de acuerdo a la normatividad vigente
- 1.1.13 Permite en forma automática realizar el análisis financiero ya sea horizontal o vertical, o Análisis con Indicadores Financieros

1.2 Exactitud

- 1.2.1 Cumple con los principios de Contabilidad como entidad, Equidad, Continuidad, periodo Contable, Unidad, valuación
- 1.2.2 Valida los datos de entrada a los diferentes módulos
- 1.2.3 Garantiza la exactitud en los informes contables
- 1.2.4 Estandarización de los formatos y procesos para el manejo de la Información de Consultas y reportes
- 1.2.5 Especifica todas las funciones contables mínimas requeridas

1.3 Interoperabilidad

- 1.3.1 Manejo de múltiples usuarios e impresoras
- 1.3.2 Exporta datos a otros ambientes o entre módulos, con utilidades o con herramientas propias del sistema.
- 1.3.3 Importa datos desde otros ambientes utilizando herramientas del motor de la base de datos o herramientas propias del sistema.
- 1.3.4 Se conecta y opera fácilmente con otros sistemas

1.4 Seguridad

- 1.4.1 Posee Claves de acceso que identifiquen a cada Usuario
- 1.4.2 Manejo de múltiples meses y posibilidad de reabrir meses ya cerrados, para los módulos que lo requieren.
- 1.4.3 Define usuarios con diferentes atributos de acceso así: de solo consulta, de grabación, de modificación, de eliminar y de transmitir
- 1.4.4 Solo permite la imputación de cuentas auxiliares o de último nivel

2. FIABILIDAD

2.1 MADUREZ

- 2.1.1 Presenta en línea reportes de alteración en los datos
- 2.1.2 Ayuda en línea y facilidad de impresión de ésta.
- 2.1.3 Presenta en línea información de datos incompletos al ejecutar un proceso
- 2.1.4 Presenta en línea información de datos incompletos al realizar un proceso

2.2 TOLERANCIA A FALLOS

- 2.2.1 Si sucede una falla en un modulo del sistema, el sistema sigue funcionando
- 2.2.2 Existen procedimientos que permitan la continuidad del trabajo ante eventuales fallas del equipo o del paquete
- 2.2.3 Si sucede una falla esta no afecta a todo el sistema
- 2.2.4 Si sucede una falla esta no daña el estado del sistema o de los datos

2.3 CAPACIDAD DE RECUPERACION

- 2.3.1 El sistema se recupera sin la intervención del operador
- 2.3.2 Es capaz de recuperar datos en caso de fallas
- 2.3.3 Utilización de fortalezas de base de datos para copias de seguridad y rutinas de recuperación.
- 2.3.4 El sistema ofrece mecanismos automáticos para la recuperación

3. USABILIDAD

3.1 CAPACIDAD DE SER ENTENDIDO

- 3.1.1 Presenta guías que permita al usuario entender el uso de sus tareas
- 3.1.2 La Cantidad de veces que el usuario accede durante una sesión, a las funciones de ayuda es:
(5) Nunca (4) 1 - 3 veces (3) 4-6 veces (2) 7-9 veces (1) Mas de 10 veces
- 3.1.3 Número de errores visualizados durante una sesión
(5) Nunca (4) 1 - 2 errores (3) 3-6 errores (2) 7-8 errores (1) Mas de 10 errores.
- 3.1.4 Tiempo que requiere el usuario para pasar de una sesión o proceso a otro
(5) Menos de 0,30seg (4) Entre 30 seg. y 1 min. (3) Entre 1 min. y 2 min.
(2) Entre 2 min. y 3 min. (1) Mas de 3 min.

3.2 CAPACIDAD DE SER APRENDIDO

- 3.2.1 Presenta guías que le permitan al usuario aprender sobre su aplicación
- 3.2.2 Los manuales son claros y ordenados
- 3.2.3 El tiempo que se requiere para ser aprendido es:
(5) 2 días (4) Dos y 4 días (3) 4 días y 5 días (2) Una semana (1) más de una semana

3.3 CAPACIDAD DE SER OPERADO

- 3.3.1 No Requiere personal especializado en computación para el manejo del paquete
- 3.3.2 La operación está conforme al objetivo, contexto y factores ergonómicos como color, forma, despliegue, interfaz gráfica

4. MANTENIBILIDAD

4.1 CAPACIDAD DE SER ANALIZADO O FACILIDAD DE ANALISIS

- 4.1.1 El paquete genera reportes de deficiencias en su proceso por falta de datos o error en ellos
- 4.1.2 El paquete informa sobre datos desactualizados de acuerdo a calendarios inicialmente definidos u otros parámetros

4.1.3 El paquete presenta reporte de posibles causas de los fallos presentados en un proceso determinado

4.2 CAPACIDAD DE SER CAMBIADO O FACILIDAD DE CAMBIO

4.2.1 No Requiere de Personal especializado para realizarle cualquier tipo de cambio o actualización

4.2.2 El paquete permite la variación de las estructuras contables sin necesidad de rediseñar el SIC

4.2.3 Permite la configuración y/o parametrización de los datos

4.2.4 Permite modificarse y adaptarse a cambios requeridos

4.3 ESTABILIDAD

4.3.1 Una modificación al software no altera el funcionamiento del mismo y no presenta efectos inesperados

4.4 CAPACIDAD DE SER PROBADO O FACILIDAD DE PRUEBA

4.4.1 Una vez modificado un elemento del software permite que este sea validado sin alterar su funcionamiento

4.4.2 Permite generar reportes de prueba una vez se han realizado modificaciones

5. PORTABILIDAD

5.1 ADAPTABILIDAD

5.1.1 Se adaptación a todas las exigencias fiscales y tributarias de la legislación colombiana

5.1.2 Define documentos de mayor utilización en la empresa

5.1.3 Se adapta el programa a los requerimientos de la empresa

5.1.4 Edición del plan de cuentas de acuerdo con las normas y necesidades de la empresa

5.2 INSTALABILIDAD

5.2.1 Permite ajustar el software a nuevos entornos de la empresa

5.2.2 El paquete No requiere de personal técnico y especializado para su instalación

5.2.3 El paquete no depende de entornos de software especializados y específicos

5.2.4 El paquete no depende de entornos de hardware especializados y específicos

5.3 COEXISTENCIA

5.3.1 Funciona en forma integrada o modular con cada una de las aplicaciones distintas existentes en la empresa

5.3.2 Es compatible con otras aplicaciones como Inventarios, Presupuesto, Activos Fijos, Nomina, facturación, entre otras

5.3.3 Está en la capacidad de compartir recursos comunes con otras aplicaciones

5.4 CAPACIDAD DE SER REEMPLAZADO

5.4.1 Permite el cambio de formatos fuentes para la presentación de informes

5.4.2 El paquete es fácil de rehusar en otras aplicaciones del mismo entorno

5.4.3 El paquete puede ser usado con el mismo propósito que otro producto de software

6. EFICIENCIA

6.1 COMPORTAMIENTO CON RESPECTO AL TIEMPO

6.1.1 El tiempo necesario para completar exitosamente una transacción:

(5) Menos de tres segundos (4) Entre 3 y 5 Segundos (3) Entre 5 segundos y 8 segundos (2) Entre 8 segundos y 10 segundos (1) Mas de 10 segundos

6.1.2 La Información se procesa siempre en tiempo real o se requiere de un procedimiento adicional

6.1.3 El número de transacciones completas con éxito en periodo de tiempo dado.

(5) mas de 20 (4) Entre 12 y 20 transacciones (3) Entre 12 y 6 transacciones

(2) entre 6 y 2 Transacciones (1) Menos de 2 transacciones

6.2 COMPORTAMIENTO CON RESPECTO A LOS RECURSOS

- 6.2.1 El tiempo de CPU usado por el software para un proceso determinado es (1) Muy Alto (2) Alto (3) Medio (4) Bajo (5)Muy bajo
- 6.2.2 Los requerimientos de Memoria Ram: (5) menos de 64 Megas (4) Entre 64 Megas y 128 Megas (3) Entre 128 Megas y 256 megas (2) Entre 256 megas y 512 Megas (1) Mas de 512 Megas
- 6.2.3 Los recursos de memoria caché requeridos por la aplicación son (1) Altos (3) Medios (5) Bajos

ANEXO 5. CASO DE ESTUDIO, EVALUACIONES

ETAPA 1: Definición de Requerimientos de Calidad: donde se especifican los requerimientos en términos de las características, subcaracterísticas y métricas a usar. En este caso se toma el software contable de la empresa A, B. y un evaluador por perfil.

ETAPA 2: Preparación de la Evaluación:

Paso 1: Se seleccionaron todas las Características, subcaracterísticas y métricas de Calidad, propuestas inicialmente por el modelo para cada perfil.

Paso 2: Definición de los Niveles de Puntaje, donde el Evaluador definió el peso correspondiente a cada característica y subcaracterística así: (tener en cuenta el anexo 4)

Evaluador 1 Producto A y B

CARACATERISTICA	Peso A	Peso B	SUBCARACTERISTICA	Peso Prod A	Peso Prod B
<i>Funcionalidad</i>	10	10	1.1	40	40
			1.2	25	25
			1.3	20	20
			1.4	15	15
<i>Fiabilidad</i>	15	15	2.1	30	30
			2.2	40	40
			2.3	30	30
<i>Usabilidad</i>	10	10	3.1	35	35
			3.2	30	30
			3.3	35	35
<i>Mantenibilidad</i>	25	25	4.1	25	25
			4.2	25	25
			4.3	25	25
			4.4	25	25
<i>Portabilidad</i>	20	20	5.1	40	40
			5.2	20	20
			5.3	25	25
			5.4	15	15
<i>Eficiencia</i>	20	20	6.1	50	50
			6.2	50	50

Evaluador 2 Producto A y B

CARACTERISTICA	Peso A	Peso B	SUBCARACTERISTICA	Peso Prod A	Peso Prod B
<i>Funcionalidad</i>	15	15	1.1	45	45
			1.2	20	20
			1.3	20	20
			1.4	15	15
<i>Fiabilidad</i>	10	10	2.1	30	30
			2.2	40	40
			2.3	30	30
<i>Usabilidad</i>	15	15	3.1	30	30
			3.2	30	30
			3.3	40	40
<i>Mantenibilidad</i>	25	25	4.1	25	25
			4.2	25	25
			4.3	25	25
			4.4	25	25
<i>Portabilidad</i>	25	25	5.1	40	40
			5.2	25	25
			5.3	20	20
			5.4	15	15
<i>Eficiencia</i>	10	10	6.1	50	50
			6.2	50	50

ETAPA 3: Evaluación, el evaluador 1 evalúa cada una de las métricas del producto A y B respectivamente.

CARACTERISTICA	Peso	SUBCARACTERISTICA	Peso	Métricas	Procd A	Prod B
<i>Funcionalidad</i>	10	1.1	40	1.1.1	5	5
				1.1.2	5	5
				1.1.3	4	3
				1.1.4	5	4
				1.1.5	4	3
				1.1.6	5	4
				1.1.7	5	5
				1.1.8	5	4
				1.1.9	5	3
				1.1.10	5	3
				1.1.11	5	4
				1.1.12	4	3
				1.1.13	4	3
1.2	25	1.2.1	5	5		
		1.2.2	4	3		
		1.2.3	4	3		
		1.2.4	4	4		
		1.2.5	4	3		
1.3	20	1.3.1	5	2		
		1.3.2	4	4		
		1.3.3	4	2		
		1.3.4	4	3		
1.4	15	1.4.1	5	5		
		1.4.2	4	2		
		1.4.3	5	5		
		1.4.4	5	4		
<i>Fiabilidad</i>	15	2.1	30	2.1.1	5	3

				2.1.2	4	5
				2.1.3	4	3
				2.1.4	4	4
		2.2	40	2.2.1	4	2
				2.2.2	5	3
				2.2.3	4	3
				2.2.4	5	4
		2.3	30	2.3.1	5	3
				2.3.2	4	3
				2.3.3	5	4
				2.3.4	5	3
<i>Usabilidad</i>	10	3.1	35	3.1.1.	5	5
				3.1.2	4	4
				3.1.3	4	4
				3.1.4	5	5
		3.2	30	3.2.1	4	4
				3.2.2	5	4
				3.2.3	2	5
		3.3	35	3.3.1	5	5
				3.3.2	4	4
<i>Mantenibilidad</i>	25	4.1	25	4.1.1	5	4
				4.1.2	5	5
				4.1.3	3	3
		4.2	25	4.2.1	5	5
				4.2.2	5	3
				4.2.3	4	3
				4.2.4	4	3
		4.3	25	4.3.1	5	3
		4.4	25	4.4.1	5	3
				4.4.2	4	3
<i>Portabilidad</i>	20	5.1	40	5.1.1	5	5
				5.1.2	5	4
				5.1.3	5	4
				5.1.4	5	5
		5.2	20	5.2.1	4	3
				5.2.2	5	5
				5.2.3	5	5
				5.2.4	4	5
		5.3	25	5.3.1	5	3
				5.3.2	5	3
				5.3.3	5	3
		5.4	15	5.4.1	4	3
				5.4.2	3	3
				5.4.3	5	5
<i>Eficiencia</i>	20	6.1	50	6.1.1	5	5
				6.1.2	5	5
				6.1.3	5	5
		6.2	50	6.2.1	5	5
				6.2.2	5	5
				6.2.3	5	5

Paso 3: Valoración. Se aplica el Modelo del Evaluador 1 y se valora para el producto A

CARACTERISTICA	Peso	SUBCARACTERISTICA	Peso	Metricas	Procd A	Calidad Subcar	Calidad Caract	Calidad Total		
<i>Funcionalidad</i>	10	1.1	40	1.1.1	5	37.38	9.0	90.92		
				1.1.2	5					
				1.1.3	4					
				1.1.4	4					
				1.1.5	3					
				1.1.6	4					
				1.1.7	5					
				1.1.8	5					
				1.1.9	4					
				1.1.10	4					
				1.1.11	4					
				1.1.12	4					
				1.1.13	3					
				1.2	25	1.2.1	4	15.00		
						1.2.2	4			
						1.2.3	4			
						1.2.4	4			
						1.2.5	3			
				1.3	20	1.3.1	3	13.00		
		1.3.2	4							
		1.3.3	3							
		1.3.4	3							
		1.4	15	1.4.1	4	11.25				
				1.4.2	3					
				1.4.3	4					
				1.4.4	4					
<i>Fiabilidad</i>	15	2.1	30	2.1.1	3	22.50	8.60			
				2.1.2	5					
				2.1.3	3					
				2.1.4	4					
				2.2	40	2.2.1	4	36.0		
						2.2.2	4			
						2.2.3	4			
						2.2.4	4			
				2.3	30	2.3.1	3	28.50		
						2.3.2	4			
		2.3.3	5							
		2.3.4	5							
<i>Usabilidad</i>	10	3.1	35	3.1.1	5	31.50	8.50			
				3.1.2	4					
				3.1.3	4					
				3.1.4	5					
				3.2	30	3.2.1	4	22.0		
						3.2.2	5			
						3.2.3	5			
		3.3	35	3.3.1	5	31.50				
				3.3.2	4					
<i>Mantenibilidad</i>	25	4.1	25	4.1.1	4	21.67	22.92			
				4.1.2	5					
				4.1.3	3					
				4.2	25	4.2.1	5	22.50		
						4.2.2	5			
						4.2.3	4			
						4.2.4	4			
				4.3	25	4.3.1	4	25.0		
		4.4	25	4.4.1	5	22.50				

<i>Portabilidad</i>	20	5.1	40	4.4.2	4	40.0	19.0
				5.1.1	5		
				5.1.2	5		
				5.1.3	5		
		5.2	20	5.1.4	5		
				5.2.1	4	18.0	
				5.2.2	5		
				5.2.3	5		
		5.3	25	5.2.4	5		
				5.3.1	5	25.0	
				5.3.2	5		
		5.3.3	4				
		5.4	15	5.4.1	4	12.0	
				5.4.2	4		
5.4.3	3						
<i>Eficiencia</i>	20	6.1	50	6.1.1	5	50.0	18.0
				6.1.2	5		
				6.1.3	5		
		6.2	50	6.2.1	5	40.0	
				6.2.2	5		
				6.2.3	5		

Valoración. Se aplica el Modelo para el Evaluador 1 y se valora para el producto B

CARACTERISTICA	Peso	SUBCARACTERISTICA	Peso	Metricas	Procd B	Calidad Subcar	Calidad Caract	Calidad Total
<i>Funcionalidad</i>	10	1.1	40	1.1.1	5	30.15	7.19	78.87
				1.1.2	5			
				1.1.3	3			
				1.1.4	4			
				1.1.5	3			
				1.1.6	4			
				1.1.7	5			
				1.1.8	4			
				1.1.9	3			
				1.1.10	3			
				1.1.11	4			
				1.1.12	3			
				1.1.13	3			
		1.2	25	1.2.1	5	18.75		
				1.2.2	3			
				1.2.3	3			
				1.2.4	4			
				1.2.5	3			
		1.3	20	1.3.1	2	11.00		
1.3.2	4							
1.3.3	2							
1.3.4	3							
1.4	15	1.4.1	5	12.00				
		1.4.2	2					
		1.4.3	5					
		1.4.4	4					
<i>Fiabilidad</i>	15	2.1	30	2.1.1	3	22.50	9.90	
				2.1.2	5			
				2.1.3	3			
				2.1.4	4			
		2.2	40	2.2.1	2	24.0		
				2.2.2	3			
				2.2.3	3			
				2.2.4	4			
		2.3	30	2.3.1	3	19.50		

				2.3.2	3		
				2.3.3	4		
				2.3.4	3		
<i>Usabilidad</i>	10	3.1	35	3.1.1.	5	31.50	8.90
				3.1.2	4		
				3.1.3	4		
				3.1.4	5		
				3.2	30		
				3.2.2	4		
				3.2.3	5		
		3.3	35	3.3.1	5	31.50	
				3.3.2	4		
<i>Mantenibilidad</i>	25	4.1	25	4.1.1	4	20.00	16.88
				4.1.2	5		
				4.1.3	3		
		4.2	25	4.2.1	5	17.50	
				4.2.2	3		
				4.2.3	3		
				4.2.4	3		
		4.3	25	4.3.1	3	15.00	
				4.4	25		
				4.4.2	3		
<i>Portabilidad</i>	20	5.1	40	5.1.1	5	36.00	16.00
				5.1.2	4		
				5.1.3	4		
				5.1.4	5		
		5.2	20	5.2.1	3	18.0	
				5.2.2	5		
				5.2.3	5		
				5.2.4	5		
		5.3	25	5.3.1	3	15.00	
				5.3.2	3		
				5.3.3	3		
		5.4	15	5.4.1	3	11.00	
				5.4.2	3		
				5.4.3	5		
<i>Eficiencia</i>	20	6.1	50	6.1.1	5	50.00	20.00
				6.1.2	5		
				6.1.3	5		
		6.2	50	6.2.1	5	50.00	
				6.2.2	5		
				6.2.3	5		

EVALUADOR 2

ETAPA 3: Evaluación, el evaluador 2 evalúa cada una de las métricas del producto A y B respectivamente

CARACTERÍSTICA	Peso	SUBCARACTERÍSTICA	Peso	Métricas	Procd A	Prod B
<i>Funcionalidad</i>	15	1.1	45	1.1.1	5	5
				1.1.2	5	5
				1.1.3	4	3
				1.1.4	4	4
				1.1.5	4	3
				1.1.6	5	4
				1.1.7	5	5
				1.1.8	5	4
				1.1.9	5	3
				1.1.10	4	3

				1.1.11	5	4
				1.1.12	4	3
				1.1.13	4	3
		1.2	20	1.2.1	4	5
				1.2.2	4	3
				1.2.3	4	3
				1.2.4	4	4
				1.2.5	3	3
		1.3	20	1.3.1	3	2
				1.3.2	4	4
				1.3.3	3	2
				1.3.4	3	3
		1.4	15	1.4.1	4	5
				1.4.2	3	2
				1.4.3	4	5
				1.4.4	4	4
<i>Fiabilidad</i>	10	2.1	30	2.1.1	3	3
				2.1.2	5	5
				2.1.3	3	3
				2.1.4	4	4
		2.2	40	2.2.1	4	2
				2.2.2	4	3
				2.2.3	4	3
				2.2.4	4	4
		2.3	30	2.3.1	3	3
				2.3.2	4	3
				2.3.3	5	4
				2.3.4	5	3
<i>Usabilidad</i>	15	3.1	30	3.1.1	5	5
				3.1.2	4	4
				3.1.3	4	4
				3.1.4	5	5
		3.2	30	3.2.1	4	4
				3.2.2	5	4
				3.2.3	5	5
		3.3	40	3.3.1	5	5
				3.3.2	4	4
<i>Mantenibilidad</i>	20	4.1	25	4.1.1	4	4
				4.1.2	5	5
				4.1.3	3	3
		4.2	25	4.2.1	5	5
				4.2.2	5	3
				4.2.3	4	3
				4.2.4	4	3
		4.3	25	4.3.1	4	3
		4.4	25	4.4.1	5	3
				4.4.2	4	3
<i>Portabilidad</i>	25	5.1	40	5.1.1	5	5
				5.1.2	5	4
				5.1.3	5	4
				5.1.4	5	5
		5.2	25	5.2.1	4	3
				5.2.2	5	5
				5.2.3	5	5
				5.2.4	5	5
		5.3	20	5.3.1	5	3
				5.3.2	5	3
				5.3.3	4	3
		5.4	15	5.4.1	4	3
				5.4.2	3	3
				5.4.3	5	5
<i>Eficiencia</i>	10	6.1	60	6.1.1	5	5

				6.1.2	5	5
				6.1.3	5	5
		6.2	40	6.2.1	5	5
				6.2.2	5	5
				6.2.3	5	5

Paso 3: Valoración. Se aplica el Modelo PARA EL EVALUADOR 2 y se valora para el producto A

CARACTERISTICA	Peso	SUBCARACTERISTICA	Peso	Metricas	Procd A	Calidad Subcar	Calidad Caract	Calidad Total		
<i>Funcionalidad</i>	15	1.1	45	1.1.1	5	40.85	12.01	88.52		
				1.1.2	5					
				1.1.3	4					
				1.1.4	4					
				1.1.5	4					
				1.1.6	5					
				1.1.7	5					
				1.1.8	5					
				1.1.9	5					
				1.1.10	4					
				1.1.11	5					
				1.1.12	4					
				1.1.13	4					
				1.2	20	1.2.1	4	15.00		
		1.2.2	4							
		1.2.3	4							
1.2.4	4									
1.2.5	3									
		1.3	20	1.3.1	3	13.00				
1.3.2	4									
1.3.3	3									
1.3.4	3									
		1.4	15	1.4.1	4	11.25				
1.4.2	3									
1.4.3	4									
1.4.4	4									
<i>Fiabilidad</i>	10	2.1	30	2.1.1	3	22.50	8.00			
				2.1.2	5					
				2.1.3	3					
				2.1.4	4					
				2.2	40	2.2.1	4	32.00		
		2.2.2	4							
		2.2.3	4							
		2.2.4	4							
				2.3	30	2.3.1	3	25.50		
		2.3.2	4							
2.3.3	5									
2.3.4	5									
<i>Usabilidad</i>	15	3.1	30	3.1.1	5	27.00	13.65			
				3.1.2	4					
				3.1.3	4					
				3.1.4	5					
				3.2	30	3.2.1	4	28.00		
		3.2.2	5							
		3.3	40	3.3.1	5	36.00				
3.3.2	4									

<i>Mantenibilidad</i>	25	4.1	25	4.1.1	4	20.00	21.25
				4.1.2	5		
				4.1.3	3		
		4.2	25	4.2.1	5	22.50	
				4.2.2	5		
				4.2.3	4		
				4.2.4	4		
		4.3	25	4.3.1	4	20.00	
		4.4	25	4.4.1	5	22.50	
				4.4.2	4		
<i>Portabilidad</i>	25	5.1	40	5.1.1	5	40.0	23.60
				5.1.2	5		
				5.1.3	5		
				5.1.4	5		
		5.2	25	5.2.1	4	23.75	
				5.2.2	5		
				5.2.3	5		
				5.2.4	5		
		5.3	20	5.3.1	5	18.67	
				5.3.2	5		
				5.3.3	4		
		5.4	15	5.4.1	4	12.0	
				5.4.2	3		
				5.4.3	5		
<i>Eficiencia</i>	10	6.1	60	6.1.1	5	60.00	10.00
				6.1.2	5		
				6.1.3	5		
		6.2	40	6.2.1	5	40.00	
				6.2.2	5		
				6.2.3	5		

Valoración. Se aplica el Modelo para el Evaluador 2 y se valora para el producto B

CARACTERISTICA	Peso	SUBCARACTERISTICA	Peso	Metricas	Procd B	Calidad Subcar	Calidad Caract	Calidad Total
<i>Funcionalidad</i>	15	1.1	45	1.1.1	5	30.15	7.19	78.87
				1.1.2	5			
				1.1.3	3			
				1.1.4	4			
				1.1.5	3			
				1.1.6	4			
				1.1.7	5			
				1.1.8	4			
				1.1.9	3			
				1.1.10	3			
				1.1.11	4			
				1.1.12	3			
				1.1.13	3			
		1.2	20	1.2.1	5	18.75		
				1.2.2	3			
				1.2.3	3			
				1.2.4	4			
		1.3	20	1.3.1	2	11.00		
				1.3.2	4			
				1.3.3	2			
1.3.4	3							

		1.4	15	1.4.1	5	12.00			
				1.4.2	2				
				1.4.3	5				
				1.4.4	4				
<i>Fiabilidad</i>	10	2.1	30	2.1.1	3	22.50		9.90	
				2.1.2	5				
				2.1.3	3				
				2.1.4	4				
		2.2	40	2.2.1	2	24.0			
				2.2.2	3				
				2.2.3	3				
				2.2.4	4				
		2.3	30	2.3.1	3	19.50			
				2.3.2	3				
2.3.3	4								
2.3.4	3								
<i>Usabilidad</i>	15	3.1	30	3.1.1.	5	31.50		8.90	
				3.1.2	4				
				3.1.3	4				
				3.1.4	5				
		3.2	30	3.2.1	4	26.00			
				3.2.2	4				
				3.2.3	5				
		3.3	40	3.3.1	5	31.50			
				3.3.2	4				
		<i>Mantenibilidad</i>	25	4.1	25	4.1.1	4		20.00
4.1.2	5								
4.1.3	3								
4.2	25			4.2.1	5	17.50			
				4.2.2	3				
				4.2.3	3				
				4.2.4	3				
4.3	25			4.3.1	3	15.00			
4.4	25			4.4.1	3	15.00			
				4.4.2	3				
<i>Portabilidad</i>	25	5.1	40	5.1.1	5	36.00		16.00	
				5.1.2	4				
				5.1.3	4				
				5.1.4	5				
		5.2	25	5.2.1	3	18.0			
				5.2.2	5				
				5.2.3	5				
				5.2.4	5				
		5.3	20	5.3.1	3	15.00			
				5.3.2	3				
				5.3.3	3				
		5.4	15	5.4.1	3	11.00			
				5.4.2	3				
				5.4.3	5				
<i>Eficiencia</i>	10	6.1	60	6.1.1	5	50.00		20.00	
				6.1.2	5				
				6.1.3	5				
		6.2	40	6.2.1	5	50.00			
				6.2.2	5				
				6.2.3	5				

ENCUESTA SOBRE LA CALIDAD DEL SOFTWARE

El objetivo de esta encuesta es recolectar datos que se utilizarán en la investigación "Modelo de evaluación de calidad del software de gestión"

Esta encuesta es realizada por el Investigador y dirigida a Ingenieros de Sistemas

INSTRUCCIONES

Esta encuesta es de carácter individual. No escriba su nombre ni firma. Le sugerimos diligenciar este cuestionario de la forma más objetiva, con el fin de obtener un resultado eficiente en la investigación que se realiza.

1. ¿Conoce Usted herramientas que evalúen la calidad del software?

Si No

Si su respuesta fue afirmativa, decir cuales herramientas _____

2. ¿Qué factores considera más importantes para medir la calidad del software?

3. ¿Tiene conocimiento de las normas que regulan la calidad del software?

Si No

4. ¿Cuales son las deficiencias más comunes que usted ha identificado en el software?

5. ¿Ha adquirido software que no cumpla con algunos de los requerimientos establecidos para la adquisición?

Si No

6. ¿Considera importante la opinión del auditor de sistemas y usuario final, para la elección del software?

Si No

7. ¿En cual de las siguientes fases del desarrollo de sistemas, considera Usted que se afecta más la calidad del software?

Análisis Diseño Análisis