
EVALUACION A LA CALIDAD DE PRODUCTO SOFTWARE BASADO EN UN ESTUDIO DE CASO

Cesar A. Moreno Gómez

Estudiante del Programa de Ingeniería de Sistemas

Universidad del Magdalena

cesar.ingsistemas@gmail.com

RESUMEN:

La calidad, tanto en la industria como en el comercio y en las organizaciones de servicios y nuevas tecnologías, es hoy en día un claro factor diferencial competitivo. Y, evidentemente, la construcción de software no es una excepción. El problema radica en que a menudo los profesionales no saben cómo aumentar la calidad de sus desarrollos de software de una forma eficaz, sin embargo, la respuesta a esta problemática puede ser sencilla.

Una herramienta clave para conseguir una aplicación de alta calidad y libre de errores es contar con un proceso de pruebas efectivo, lamentablemente, y aunque es una de las técnicas fundamentales del aseguramiento de la calidad, al proceso de pruebas no se le suele dar la importancia que merece, lo que significa que muchos proyectos no cuentan con una buena metodología y/o con las herramientas adecuadas para diseñar e implementar un proceso de pruebas adecuado a sus necesidades.

Debido a la complejidad de probar una aplicación, la etapa de pruebas puede llegar a ser de las más lentas del proceso de desarrollo de software, pero, si un proceso de pruebas se desarrolla siguiendo una planificación, una metodología y unas herramientas adecuadas se pueden conseguir importantes disminuciones en los costos de desarrollo y mantenimiento del software, así como una reducción en el número de errores.

En este artículo, se pretende entregar una visión de cómo es posible medir la calidad en los proyectos informáticos, en especial un producto software, enmarcándose en el estudio de un caso, aplicándole una serie de evaluaciones y pruebas al producto final, teniendo como criterio los conocimientos y herramienta adquiridos en el diplomado de Gestión de Calidad del Software, y basándonos en algunos de los estándares de la ISO, IEEE, CMMI entre otros para la evaluación de calidad de procesos y productos software.

PALABRAS CLAVE:

Estándar, Calidad, Software, SINAI, Validación, Prueba.

ABSTRAC:

The quality, as much in the industry as in the commerce and the organizations of services and new technologies, is nowadays a clear factor competitive differential. And, evidently, the construction of software is not an exception. The problem is in which often the professionals do not know how to increase the quality of its developments of software of an effective form, nevertheless, the answer to this problematic one can be simple.

A key tool to obtain an application of high free quality and of errors is to count on an effective process of tests, lamentably, and although she is one of the fundamental techniques of the securing of the quality, to the process of tests usually it does not occur the importance to him that deserves, which means that many projects do not count on a good methodology and/or the suitable tools to design and to implement a process of tests adapted to their necessities.

Due to the complexity to prove an application, the stage of tests can get to be from slowest from the process of development of software, but, if a process of tests are developed following a planning, a methodology and suitable tools to important diminutions in the development costs and maintenance of software can be obtained, as well as a reduction in the number of errors.

In this article, it is tried to give a vision of how reaching a high quality in the computer science projects, in special a software product, framing itself in the study of a case, applying to him a series of evaluations and tests to the end item, having like criterion the acquired knowledge and tool in the graduate of Management of Quality of Software, and basing to us on some of the standards of the ISO, IEEE, CMMI among others after the evaluation of quality of processes and software products.

KEYWORDS

Standard, Quality, Software, SINAI, Validation, Test.

1 INTRODUCCIÓN

A pesar de la creciente participación del software en el mundo actual y de los avances producidos, su proceso de construcción aun no es el adecuado.

Actualmente, la satisfacción hacia el uso de un producto de software puede marcar una gran diferencia en un mercado que muestra una gama de productos similares. Es así como el desarrollo de artículos que integren las expectativas de los clientes y usuarios, y además posean un alto índice de calidad pueden generar una gran diferencia entre la amplia gama que compiten en el mercado.

Pero ante esta afirmación surge un interrogante, ¿Cuándo un producto software es de calidad?, Ahora bien, la preocupación por ofrecer productos acompañados de altos niveles de calidad no es una actividad nueva, a lo largo de este siglo han surgido distintas interpretaciones de como brindar calidad a estos, por ende el desarrollo de productos software no está ausente de dicha intranquilidad.

Siendo esto así, entonces se podrá afirmar, ¿Qué la calidad de un producto software se basa en el desarrollo de este?

Antes de empezar hablar acerca de en qué podría consistir la calidad de los productos software, se debería definir qué se entiende por calidad, a que es aplicable y de qué forma puede ser relacionada con productos software.

Según el diccionario de la Real Academia de la Lengua Española, calidad se puede definir como: *“Propiedad o conjunto de propiedades inherentes a una cosa que permiten apreciarla como igual, mejor o peor que las restantes de su especie”.*

Por lo cual, es posible suponer que la calidad de un producto es medible en términos tales como, si sus características o atributos son comparables con una serie de características que posee otro software, sin embargo, la palabra *calidad* tiene múltiples significados, por tanto, debe definirse en el contexto que se esté considerando.

Teniendo claro lo anterior centremos dicho contexto en el desarrollo de productos software, por lo cual, la calidad de éste depende de cómo responda a las preferencias y a las necesidades de los clientes, por lo que se dice que la calidad es adecuación al uso de sí mismo en la actualización de los roles presentados a un consumidor.

Por ello la calidad puede ser vista como un estilo o filosofía de vida en un mundo que está cambiando y evolucionando para desarrollar un lugar mejor donde vivir.

Pero cerrando un poco más el marco hacia la calidad teniendo una perspectiva más hacia producto como tal, se puede afirmar que está viene dada por la apreciación del cliente objetivo de ese producto, en función del conjunto de características que ese cliente evalúa para el producto, y del nivel significativo que cada una de ellas tiene para ese cliente, es decir, si un producto software cumple una serie de características instauradas por dicho cliente, se valido decir que es un producto software de calidad.

Sin embargo, según Pressman, la calidad es *“la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados, y con las características implícitas que se espera de todo software desarrollado profesionalmente”*.

Esto lleva a pensar qué no solo la visión del usuario o cliente final es lo único a tener en cuenta, también hay que preocuparse por los estándares de desarrollo; con esto la calidad de un producto de software entra en un campo más amplio en donde la calidad de un producto de software, va ligada a la calidad del proceso de desarrollo que se gestionó para la creación del artefacto final.

Callaos y Callaos proponen un concepto de calidad del software en el cual están involucrados tanto características internas como el contexto organizacional, lo que genera un enfoque sistémico del concepto de Calidad del Software.

En todos los enfoques de calidad, incluyendo el de la calidad del software, la organización necesita ser considerada como organización orgánica más bien que mecánica. Usa sus recursos, las ideas y los esfuerzos para transformar las entradas en mercancías y servicios de la calidad. Así pues, las características y las tecnologías necesarias para lograr este tipo de acercamiento deben ser determinadas. Esta es la razón por la cual el proyecto del desarrollo del software se considera el

elemento de la organización a través del cual se maneja el desarrollo del software.

Basándose en esto, el proyecto se puede considerar como el sistema que define el contexto que modela la calidad del proceso y calidad del producto respectivamente. Tal como indica Humphrey, *“La calidad del proceso garantiza la calidad del producto y consecuentemente no se pueden desligar estas dos calidades”*.

Todo esto se resume en decir que no solo es la calidad del individual del producto, sino la calidad de este y la calidad del proceso de desarrollo para la construcción del mismo, esto es conocido como Calidad Sistémica del Producto.

Manteniendo las ideas expuestas con anterioridad, en este artículo, se desarrolló una evaluación a un producto de software, un caso de estudio por lo cual este documento se encuentra subdividido en varias secciones que mostraran el trabajo realizado.

En la sección 2 se encontrará todo lo concerniente al desarrollo del trabajo realizado sobre el estudio de caso, dividiéndose esta sección en tres sub secciones donde se mostraran una información general sobre sistemas web, información general y concreta sobre el caso de estudio y en lleno la evaluación a este, buscando entender porque la calidad de un producto de software radica en sus dos partes, producto y proceso de desarrollo.

Adicionalmente, dicha evaluación se subdividió en tres fases principales.

En la primera fase, se evaluó la calidad del producto de software como tal, es decir, se verificó si el sistema cumple ciertas características donde la base fue el estándar ISO/IEC 9126 para evaluación de productos de software, luego se realizaron las diversas mediciones que se requieran y se obtuvo el nivel de calidad del producto de software.

La segunda fase, se presentará lo concerniente a la calidad del proceso de desarrollo de software, se determinó el grado de aplicación y el nivel de calidad de este realizando una comparación con el modelo de calidad de proceso, desarrollado por el Instituto de Ingeniería del Software de la

Universidad Carnegie Mellon, el modelo CMMI Capability Maturity Model Integration (Integración del Modelo de Capacidad y Madurez).

Y en la última fase, se realizó una integración de las calidades parciales de los sub-modelos, lo cual es hacer una integración de las fases anteriores y establecer el nivel de calidad que tiene el proyecto de software evaluado.

Ya en la tercera sección principal, de este artículo, se presentan una serie de conclusiones, validas a tener en cuenta para futuros trabajos.

Por último se hará una serie de anotaciones y agradecimientos a todos los que intervinieron de alguna forma en la realización de este trabajo y la formación profesional del autor.

2 DESARROLLO DEL TEMA

Antes de comenzar a realizar es bueno tener ciertos datos iniciales, que nos ayudarán a comprender mucho más fácil el trabajo realizado.

2.1 Calidad en Sistemas de Información en Internet

Debido a la importancia que la calidad de software en Internet ha despertado en los últimos años, la Conferencia Internacional de la Ingeniería de Software del año 2002 (ICSE 2002) se centró en los aspectos de calidad para los Sistemas en Internet. En esta conferencia se concluyó que los criterios de calidad más importantes son los siguientes:

- ♣ *Fiabilidad*
- ♣ *Usabilidad*
- ♣ *Funcionalidad*
- ♣ *Escalabilidad*
- ♣ *Seguridad*
- ♣ *Mantenibilidad*

Con lo descrito anteriormente, es posible darse cuenta que el estudio de los atributos de calidad para sistemas de software puede ser muy extenso. Así mismo, la Internet fue diseñada para presentar información, mediante una apariencia sencilla, con sitios (o portales) sin una complejidad significativa, ya que su contenido era primordialmente de texto e hipervínculos. Sin embargo, a medida que fue creciendo la Internet, las aplicaciones se volvieron

más complejas y con mayores demandas de *Calidad*. Algunos ejemplos de estas aplicaciones son el comercio electrónico, las aplicaciones distribuidas, las Intranets, etc.

Mucha de la complejidad asociada al desarrollo de aplicaciones en Internet, tiene que ver con el proceso de desarrollo elegido y también con la integración de los elementos que las conforman. Cada día, un mayor número de organizaciones manejan su información por medio de sistemas en Internet, por lo cual, se demanda una mayor calidad en todos los aspectos, para este tipo de sistemas de software.

Luego de entender esta pequeña información se va a trabajar de lleno en el estudio de caso.

2.2 Estudio de Caso SINAI



Figura. 1 Logo de SINAI

El aplicativo SINAI, abreviatura para *Sistema de Información para el Control y Seguimiento de los Proyectos Financiados por el Fondo Patrimonial para la Investigación*, siendo este un aplicativo web desarrollado para la Vice-rectoría de Investigación de la Universidad del Magdalena creado por los Ingenieros *Dana Caballero Navarro* y *Carlos Mario Lizarazo Terraza*, con el propósito de que esta herramienta realizara la gestión de la información de los proyectos de investigación avalados por la Vice-rectoría de Investigación de la Universidad del Magdalena.

Esta aplicación se ha desarrollado bajo sistemas distribuidos. Estos sistemas permiten que exista una comunicación y coordinación entre unos elementos de cómputo conectados a una red, por medio del intercambio de mensajes.

En la funcionalidad de un programa distribuido se pueden definir tres niveles que son: el manejador de la base de datos (nivel de almacenamiento), el

procesador de aplicaciones o reglas del negocio y la interfaz de usuario (nivel de presentación). Esta distribución se conoce como el modelo o arquitectura de tres capas (ver figura. 2).

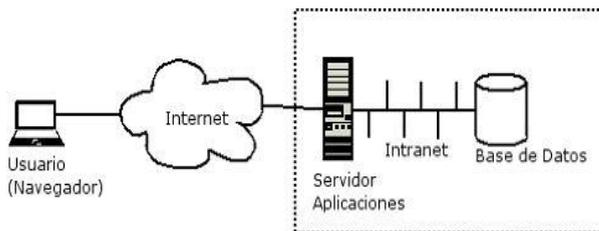


Figura. 2 Arquitectura de Tres Capas

La arquitectura de tres capas tiene como objetivo primordial la separación de la lógica de negocios de la lógica de diseño, un ejemplo básico de esto es separar la capa de datos de la capa de presentación al usuario.

Para tener un poco más de información se presentan algunos datos técnicos del sistema como tal (Ver Tabla 1).

Tabla. 1 Información Técnica de SINAI

Funciona bajo Computación Distribuida
Motor de Base de datos Oracle 9i
Servidor de Aplicaciones: Sun Java Application Server
Enterprise Java Beans.

2.2.1 Computación Distribuida

Los sistemas distribuidos permiten redistribuir el peso del procesamiento desarrollando partes del mismo en diferentes puntos, permitiendo aliviar la carga al servidor e intercambiar datos. Este sistema se basa en el modelo de cliente servidor, como se explicaba anteriormente.

2.2.2 Servidor de Aplicaciones: Sun Java Application Server

El Sun Java System Application Server es un servidor de aplicaciones sobre la plataforma J2EE y provee las bases para construir aplicaciones

confiables, escalables y manejables. Contiene un conjunto de novedades y soporte para desarrollo basado en componentes, el Java System Application Server provee funcionalidades subyacentes básicas para el desarrollo y el despliegue de aplicaciones orientadas a negocio.

La figura 3 muestra en detalle el anterior enunciado.

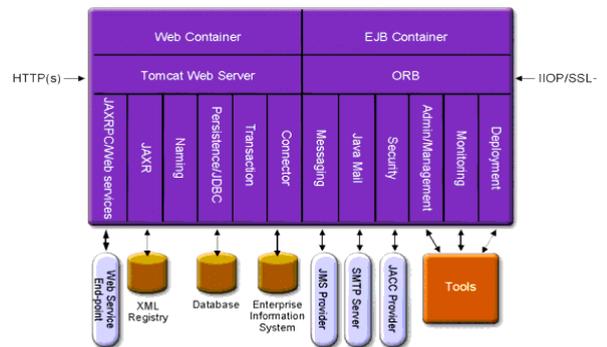


Figura 3. Arquitectura del servidor de aplicaciones.

2.2.3 Enterprise Java Beans

Un componente Enterprise JavaBeans, o un bean enterprise, es un cuerpo de código que contiene campos y métodos para implementar módulos de lógica de negocios. Se puede imaginar un bean enterprise como un bloque de construcción que puede ser usado solo con otros bean enterprise para ejecutar la lógica de negocios en el servidor Java EE.

2.2.4 Motor de Base de Datos Oracle 9i

En este se almacenan las bases de datos, tablas, índices. Es uno de los servidores que más carga tiene. El servidor de base de datos utilizado fue ORACLE 9i, ya que es uno de los más robustos y confiables a nivel de integridad de datos y seguridad, además, la Universidad cuenta con licencia Oracle. Esta tecnología permite que las bases de datos sean escalables respecto a las amplias posibilidades que abren las nuevas tecnologías de la conectividad, Oracle ha apostado por desarrollar una herramienta totalmente integrada en Internet y con unas características de fiabilidad, escalabilidad y seguridad de acuerdo con las necesidades de la nueva economía.

2.2.5 Arquitectura del Sistema

Este sistema está dividido en dos grandes áreas que son, la parte pública y la parte de administración.

La sección pública del sistema, principalmente es un área de carácter informativo, que publicará noticias y eventos de la Vice-rectoría de Investigación, relacionadas con convocatorias, convenios, grupos y proyectos.

En la parte de administración, se adicionará la información que sea publicada en este entorno web. Esta sección la integrarán los siguientes módulos: administración general, Centro, Grupo, Investigador, Proyecto de Investigación.

Cada módulo se caracteriza de la siguiente manera:

- ♣ Módulo de Administración general: en este módulo se manipula la información de las tablas auxiliares, para proveer consultas básicas.
- ♣ Módulo de Centro: Desde aquí se maneja la información relacionada con los Centros de Investigación de la Universidad del Magdalena y cualquier tipo de búsqueda en la que se solicite información de los mismos. Además, desde aquí se pueden administrar las áreas de investigación que desde el Plan de Ciencia y Tecnología se estipulen, para su investigación desde los Centros.
- ♣ Módulo de Grupo: En esta sección se gestiona la información de los grupos de investigación y su interacción con los Centros de Investigación, los Investigadores y los Proyectos de Investigación. A esto se suma que se pueden realizar consultas basadas en el Plan de Ciencia y Tecnología (capital, eje, programa, subprograma, línea de investigación).
- ♣ Módulo de Investigador: este módulo gestiona los datos personales del administrador, sumado a ciertos datos especiales que permitirán realizar consultas específicas.

- ♣ Módulo de Proyecto: se manejan los parámetros necesarios para el registro, control y seguimiento de los proyectos de investigación, financiados por Fonciencias, además de un menú variado de consultas bajo distintos parámetros, para la eficiente administración de este módulo.

2.2.6 Metodología utilizada para el desarrollo y/o la construcción del Producto Software

En cuanto a la metodología de desarrollo elegida para la elaboración de la solución informática en el marco del caso de estudio, se optó por el Proceso Unificado (Rational Unified Process), con la idea de construir la solución sistemática (software), a través de las fases que desde aquí se plantean.

Esta metodología organiza las actividades de desarrollo siguiendo dos criterios ortogonales.

En el eje vertical, se describen las actividades fundamentales y que en términos de RUP se denominan *componentes*, los cuales establecen cómo avanzar en la conceptualización y construcción del sistema.

En el eje horizontal, se describen los criterios para la planeación y control en el tiempo, corresponden a la dinámica del proceso de desarrollo pues establecen cuándo se deben realizar las acciones definidas por los componentes, como se aprecia en la figura 4.



Figura 4. Esquema de la metodología RUP

2.3 Desarrollo de la Evaluación

Como se había mencionado anteriormente para tratar de mostrar en que consiste la calidad de un producto software se tomó un caso real, en esta oportunidad SINAI, al cual se le realizaron una serie de pruebas y test para verificar que tanta calidad posee el software.

Ahora bien para llevar a cabo esta evaluación se decidió crear un plan de evaluación, el cual entregó las pautas para realizar dicha evaluación de una manera ordenada y segura, entregando resultados cuantitativos y de fácil verificación.

Adicional a esto se tomó la decisión de dividir la prueba en dos partes para poder facilitar el trabajo de evaluación, estas dos partes son calidad en el producto como tal, y revisión de la calidad del proceso de construcción y desarrollo de este producto, cabe resaltar que para ambos casos se utilizo el mismo plan y no se presentó ningún inconveniente a la hora de aplicarlo.

Ahora bien como se trata, realizar evaluaciones y como la forma mas común y conocida de evaluar cualquier cosa es realizando ciertas comparación y pruebas, en este trabajo se optó por tener esta forma de evaluación, gracias a ser relativamente fácil de desarrollar y los resultados son sencillo de analizar, para esto se tomaron como puntos de referencias de calidad dos estándares que proporcionan la información pertinente y precisa para entregar un alto grado de confiabilidad a la evaluación desarrollada.

Estos estándares son, la norma internacional ISO/IEC 9126, la cuál proporciona las guías de un modelo de calidad para un producto de software. Esta norma sigue el esquema propuesto por los modelos anteriores (McCall, Boehm, etc), además, retoma algunas de las características planteadas por dichos modelos, pero puntualiza seis características que considera fundamentales en la evaluación de calidad de un producto de software, a saber: la funcionalidad, la confiabilidad, la usabilidad, la eficiencia, la mantenibilidad y la portabilidad. Las características propuestas por la Norma ISO/IEC 9126 indican los aspectos que se

deben observar al producto de software que se desea evaluar.

En la figura 5 muestran las seis características planteadas por la Norma con sus correspondientes subcaracterísticas.

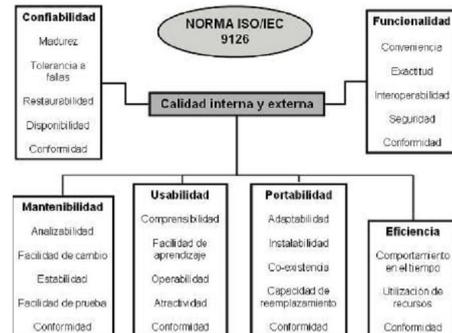


Figura 5. Modelo de calidad interna y externa propuesto por la Norma ISO/IEC 9126.

Como se había mencionado antes, esta norma será la guía que se utilizará para la evaluación del producto, de aquí se sustraerán 3 características por medio de las cuales y basado en unas métricas propias del modelo, se pudieron realizar mediciones más concretas que arrojaron resultados claros y qué llevaron a conclusiones lógicas, más adelante en este trabajo se denotara esta forma de evaluar.

Y para evaluar el proceso de producción y/o desarrollo de dicho producto, se hizo uso de otro modelo muy conocido, el modelo CMMI (Capability Maturity Model Integration) de SEI (Instituto de Ingeniería del Software de la Universidad Carnegie Mellon).

En términos del SEI, el Software Process (Proceso Software) es el conjunto de actividades, métodos, prácticas y transformaciones que la gente usa para desarrollar y mantener sus productos asociados a planes de proyecto, modelos, documentación, etc.

El SEI desarrolló un modelo que de alguna manera permite clasificar la madurez del proceso de software en cinco niveles.

El CMMI es un modelo de calidad del software que clasifica las empresas en niveles de madurez. Estos niveles sirven para conocer la madurez de los procesos que se realizan para producir software.

♣ **Definir las Características, con cada una de sus Subcaracterísticas y las métricas a evaluar.**

Se trata de establecer qué se va evaluar, es decir, definir qué o cuáles características se van a evaluar.

♣ **Establecer las métricas a aplicar en la evaluación.**

Se refiere a establecer las métricas, que se utilizarán para medir la calidad del proyecto o producto a evaluar.

♣ **Determinar el porcentaje de peso que tendrá cada una de las características, subcaracterísticas y atributos.**

Es simplemente definir con criterios validos, que peso tendrá para nuestra evaluación cada característica, subcaracterísticas y atributos.

♣ **Definir Escala de Medición para el nivel de la Calidad**

Se refiere a establecer una escala de niveles de calidad en donde se pueda determinar el nivel del software después de realizada la evaluación.

♣ **Establecer las herramientas a utilizar en la evaluación**

Es definir o establecer qué herramientas se van a utilizar tanto para la recolección de la información como para el análisis de dicha información.

✓ **Recopilar la información de la evaluación.**

Es realizar la evaluación concretamente como tal, es decir, recolectar todos los datos posibles para poder llegar a una conclusión.

✓ **Analizar los resultados obtenidos.**

Se refiere a analizar los datos obtenidos y llegar a una conclusión.

✓ **Concluir el nivel de calidad en que se encuentra el proyecto o Producto.**

Dar un nivel de calidad al producto dependiendo de los resultados obtenidos a partir de los objetivos propuestos.

✓ **Documentar la evaluación realizada en el proyecto.**

Hacer una documentación de las conclusiones que se obtuvieron de la evaluación de calidad realizada en el proyecto.

Este plan de evaluación fue el que se aplicó para evaluar, valga la redundancia, la calidad del producto y la calidad del proceso de desarrollo de este.

2.3.2 Evaluación del Producto Software

Como se había mencionado con anterioridad en este artículo, para proceder a realizar la evaluación del producto se basó en la norma ISO/IEC 9126, pero al entender que un trabajo de evaluación es muy extenso se decidió, teniendo en cuenta lo concluido por la Conferencia Internacional de la Ingeniería de Software del año 2002 (ICSE 2002), atacar a tres (3) de las características principales para realizar nuestra evaluación.

De las tablas 2 a la 4 se definen las características a tener en cuenta al realizar la evaluación y sus subcaracterísticas.

Tabla 2. Definición de Funcionalidad

Funcionalidad
La capacidad del producto de software para proveer las funciones que satisfacen las necesidades explícitas e implícitas cuando el software se utiliza bajo condiciones específicas.

Tabla 2.1 Sub Características de Funcionalidad

Adecuación	La capacidad del producto de software para proveer un adecuado conjunto de funciones para las tareas y objetivos especificados por el usuario.
Exactitud	La capacidad del producto de software para proveer los resultados o efectos acordados con un grado necesario de precisión.
Conformidad con la Funcionalidad	La capacidad del producto de software de adherirse a los estándares, convenciones o regulaciones legales y prescripciones similares referentes a la funcionalidad.

Tabla 3. Definición de Fiabilidad

Fiabilidad
La capacidad del producto de software para mantener un nivel específico de funcionamiento cuando se está utilizando bajo condiciones especificadas.

Tabla 3.1 Sub Características de Fiabilidad

Madurez	La capacidad del producto de software para evitar fallas como resultado de errores en el software.
Tolerancia a fallos	La capacidad del producto de software para mantener un nivel especificado de funcionamiento en caso de errores del software o de incumplimiento de su interfaz especificada
Recuperabilidad	La capacidad del producto de software para restablecer un nivel especificado de funcionamiento y recuperar los datos afectados directamente en el caso de una falla.
Conformidad con la fiabilidad	La capacidad del producto de software para adherirse a las normas, convenciones o regulaciones relativas a la fiabilidad.

Tabla 4. Definición de Usabilidad

Usabilidad
La capacidad del producto de software de ser entendido, aprendido, usado y atractivo al usuario, cuando es utilizado bajo las condiciones especificadas.

Tabla 4.1 Sub Características de Usabilidad

Apropiabilidad	La capacidad del producto de software para permitir al usuario entender si el software es adecuado, y cómo puede ser utilizado para las tareas y las condiciones particulares de la aplicación.
Facilidad de Aprendizaje	La capacidad del producto de software para permitir al usuario aprender su aplicación.
Operabilidad	La capacidad del producto de software para permitir al usuario operarlo y controlarlo.
Atractibilidad	La capacidad del producto de software para gustarle al usuario
Conformidad del uso	La capacidad del producto de software para adherirse a los estándares, convenciones, guías de estilo o regulaciones relacionadas a la usabilidad.

Definidas las características y subcaracterísticas se continúa estableciendo una ponderación, o un valor porcentual, es decir, qué tanto peso tendrá una característica en esta evaluación, luego se le entregará de forma muy similar a cada subcaracterística, esto se muestra en la tabla 5:

Tabla 5. Ponderación cada Característica

Características	Ponderación
Funcionalidad	30 %
Usabilidad	40 %
Fiabilidad	30 %
Total (%)	100 %

Y las subcaracterísticas se ponderan de la siguiente manera (Ver tablas de la 6 a la 8):

Tabla 6. Sub Características de Funcionalidad (Ponderación)

Subcaracterística	Ponderación
<i>Adecuación</i>	30 %
<i>Exactitud</i>	40 %
<i>Conformidad con la Funcionalidad</i>	30 %
Total	100 %

Tabla 7. Sub Características de Fiabilidad (Ponderación)

Subcaracterística	Ponderación
Madurez	30 %
Tolerancia a Fallos	20 %
Recuperabilidad	30 %
Conformidad con la Fiabilidad	20 %
Total	100 %

Tabla 8. Sub Características de Usabilidad (Ponderación)

Subcaracterística	Ponderación
Aprobiabilidad	20 %
Facilidad de Aprendizaje	30 %
Operabilidad	20 %
Atractibilidad	20 %
Conformidad del Uso	10 %
Total	100 %

A partir de estas ponderaciones se procedió a encontrar una serie de atributos y métricas, estas métricas de software son medidas usadas para obtener valores exactos que indiquen la tendencia de un producto de software con respecto a determinada característica que se quiera conocer. Una métrica de software es “una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado”. Es decir, una métrica relaciona de alguna forma las medidas individuales sobre algún aspecto. (Ver figura 7).

CARACTERÍSTICAS	SUBCARACTERÍSTICAS	ATRIBUTOS	MÉTRICAS	Justificación Evaluación / Calificación	
				Valor	Unidad
FUNCIONALIDAD	ADECUACIÓN	¿Están de acuerdo con la especificación funcional?	Func. Correctas = 10 Func. Faltantes = 10	0,50	1,00
		¿Se han detectado errores de implementación o implementación incorrecta?	Func. Correctas = 10 Func. Faltantes = 10	0,50	1,00
		¿Se han detectado errores de implementación o implementación incorrecta durante el desarrollo de ciclo de vida?	Func. Correctas = 10 Func. Faltantes = 10	0,50	1,00
		¿Se han detectado errores de implementación o implementación incorrecta durante el desarrollo de ciclo de vida?	Func. Correctas = 10 Func. Faltantes = 10	0,50	1,00
		¿Se han detectado errores de implementación o implementación incorrecta durante el desarrollo de ciclo de vida?	Func. Correctas = 10 Func. Faltantes = 10	0,50	1,00
		¿Se han detectado errores de implementación o implementación incorrecta durante el desarrollo de ciclo de vida?	Func. Correctas = 10 Func. Faltantes = 10	0,50	1,00
	EXACTITUD	¿Se han detectado errores de implementación o implementación incorrecta durante el desarrollo de ciclo de vida?	Func. Correctas = 10 Func. Faltantes = 10	0,50	1,00
		¿Se han detectado errores de implementación o implementación incorrecta durante el desarrollo de ciclo de vida?	Func. Correctas = 10 Func. Faltantes = 10	0,50	1,00
		¿Se han detectado errores de implementación o implementación incorrecta durante el desarrollo de ciclo de vida?	Func. Correctas = 10 Func. Faltantes = 10	0,50	1,00
	CONFORMIDAD CON LA FUNCIONALIDAD	¿Se han detectado errores de implementación o implementación incorrecta durante el desarrollo de ciclo de vida?	Func. Correctas = 10 Func. Faltantes = 10	0,50	1,00
		¿Se han detectado errores de implementación o implementación incorrecta durante el desarrollo de ciclo de vida?	Func. Correctas = 10 Func. Faltantes = 10	0,50	1,00
		¿Se han detectado errores de implementación o implementación incorrecta durante el desarrollo de ciclo de vida?	Func. Correctas = 10 Func. Faltantes = 10	0,50	1,00

Figura 7. Esquema de Atributos y Métricas

Si siguiendo estos patrones en esta evaluación se utilizaron una serie de métricas, mediante las cuales se obtuvieron una serie de resultados los cuales se exponen a continuación (Ver Tablas 9 y 9.1)

Tabla 9. Calificación para la Funcionalidad

Características	Calificación Total	Ponderaciones Total	Calificación Ponderada Total
Funcionalidad	82 %	30 %	24,6 %

Tabla 9.1 Calificación para subcaracterísticas de la Funcionalidad

Subcaracterísticas	Calificación Total	Ponderaciones Total	Calificación Ponderada Total
Adecuación	90 %	30 %	27 %
Exactitud	100 %	40 %	40 %
Conformidad con la Funcionalidad	50 %	30 %	15 %

La figura 8 muestra el comportamiento de cada subcaracterística contra la ponderación que se le dio al iniciar la evaluación, la suma de estas ponderaciones entrega una calificación de la característica, sin embargo esta calificación se debe ponderar con lo especificado en el cuadro de ponderaciones de las características (Tabla 5.)

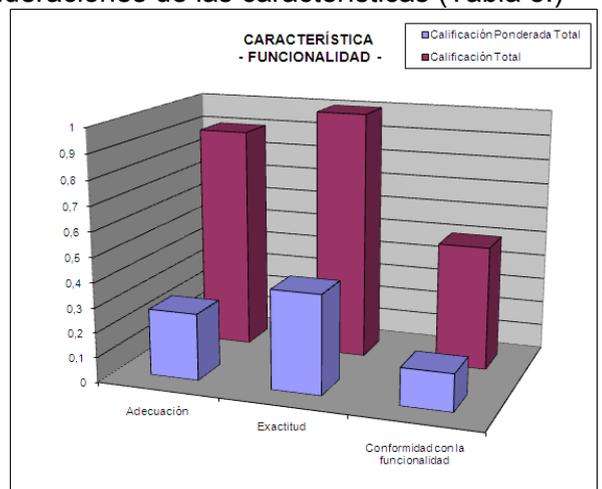


Figura 8. Calificación para Funcionalidad

Esto se cumple para las 3 características escogidas de la siguiente manera (ver tablas de 10 a la 11):

Tabla 10. Calificación para la Fiabilidad

Características	Calificación Total	Ponderaciones Total	Calificación Ponderada Total
Fiabilidad	52,1 %	30 %	15,6 %

Tabla 10.1 Calificación para subcaracterísticas de la Fiabilidad

Subcaracterísticas	Calificación Total	Ponderaciones Total	Calificación Ponderada Total
Madurez	35 %	30 %	10,5 %
Tolerancia a fallos	60 %	20 %	12 %
Recuperabilidad	70 %	30 %	21 %
Conformidad con la fiabilidad	43 %	20 %	8,6 %

La figura 9 muestra el comportamiento de los datos con lo concerniente a la característica de la Fiabilidad.

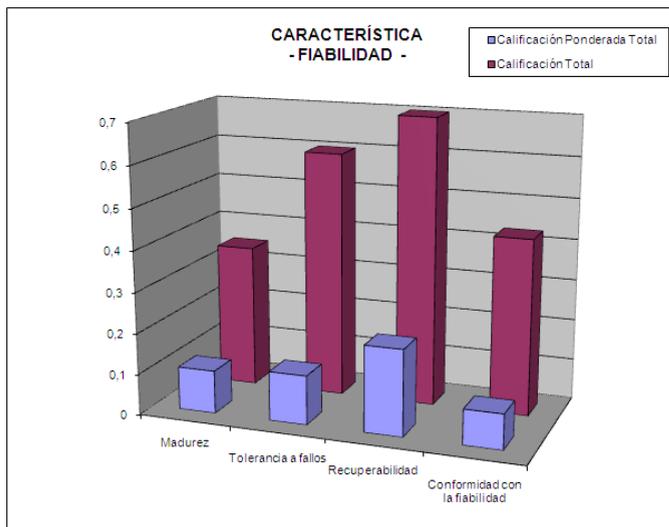


Figura 9. Calificación para Fiabilidad

Tabla 11. Calificación para la Usabilidad

Características	Calificación Total	Ponderaciones Total	Calificación Ponderada Total
Usabilidad	73,3 %	40 %	28,5 %

Tabla 11.1 Calificación para subcaracterísticas de la Usabilidad

Subcaracterísticas	Calificación Total	Ponderaciones Total	Calificación Ponderada Total
Apropiabilidad	76	20 %	15,2 %
Facilidad de Aprendizaje	77	30 %	23,1 %
Operabilidad	70	20 %	14 %
Atractibilidad	62	20 %	12,4 %
Conformidad del uso	66	10 %	6,6 %

Y en la figura 10 se presentan los datos correspondientes a la característica de la usabilidad.

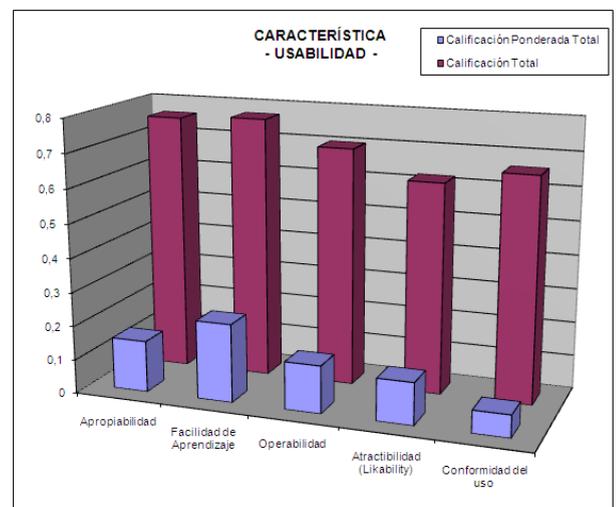


Figura 10. Calificación para Usabilidad

Ahora bien, lo mismo que se ha hecho para calificar las características se hará para la calificación del producto, es decir se sumarán las calificaciones de las características y se encontrará el dato concreto de calidad.

Pero antes de esto analizan ciertos aspectos a tener en cuenta, primero que todo se creará una matriz en donde se mostraran los diferentes niveles de calidad, esto se muestra en la figura 12:

Tabla 12. Matriz de Niveles de Calidad

Intervalo de Calidad	Nivel de Calidad
85 % - 100 %	Excelente
70 % - 84 %	Bueno
50 % - 69 %	Aceptable
0 % - 49 %	Deficiente

Esta matriz servirá para valorar el producto de software y cada una de las características, por lo cual se puede decir, qué SINAI con respecto a la Funcionalidad y a la Usabilidad se encuentra en un nivel bueno, pero en lo relacionado a la Fiabilidad su nivel es aceptable.

Avanzando en el tema, se observara en qué nivel de calidad se encuentra el producto de software SINAI, en donde como se mencionó, se tomarán las características, las cuales se sumarán y permitirán ver el nivel de calidad como se aprecia en la tabla 13.

Tabla 13. Calificación global del Producto Software

Características	Calificación Total	Ponderación Total	Calificación Ponderada Total
Funcionalidad	82 %	30 %	24,6 %
Fiabilidad	52,1 %	30 %	15,6 %
Usabilidad	71,3 %	40 %	28,5 %

En la figura 11, así como en las anteriores graficas se muestra la calificación que obtuvo la característica contra la calificación ponderada total teniendo en cuenta la ponderación ya definida en la tabla de ponderación de características (Tabla. 5).

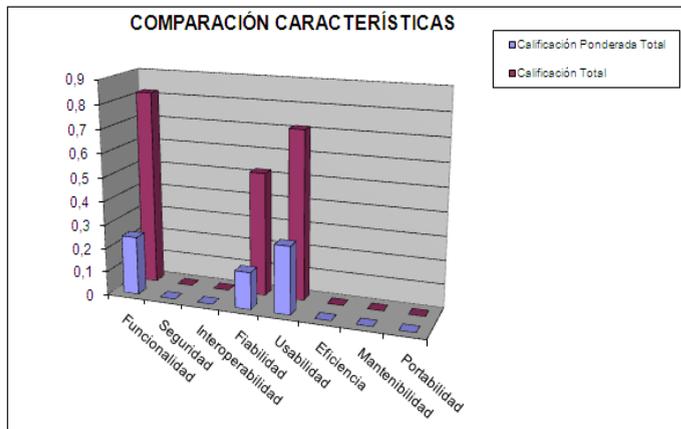


Figura 11. Calificación Total del Producto

Ahora bien si se suman estos datos de calificación ponderada, se obtendrá que nuestro software posee una calificación de 68,7 % de calidad, lo cual lo colocaría en un nivel de calidad aceptable, teniendo en cuenta que la Funcionalidad y la Usabilidad son

sus mejores características y la Fiabilidad es algo baja con respecto a lo esperado.

De esta manera se califica a SINAI con nivel aceptable a lo que corresponde al producto, pero no se debe olvidar que la calidad se descompone en dos partes y esta es solo una por lo cual ahora se verá cómo le va con el proceso de desarrollo.

2.3.3 Evaluación de Proceso de Desarrollo

Para esta parte se hará algo muy sencillo, puesto a la complejidad del modelo CMMI y el alcanzar sus cinco niveles es algo muy costoso y de largo tiempo solo se enfocará en los tres primeros niveles. Al alcanzar un nivel tres de madurez, según CMMI se puede hablar de calidad en un proceso de desarrollo.

Para esto entiéndase que para cumplir ciertos niveles de madurez según CMMI, se deben cumplir una serie de áreas de procesos los cuales se muestran a continuación:

Como el nivel 1, es el nivel inicial y es en el que se encuentra cualquier proceso de desarrollo, por lo cual no se tendrá en cuenta.

Para el nivel 2 se encuentran las siguientes:

- Gestión de la configuración
- Medición y análisis
- Monitorización y control de proyecto
- Planificación de proyecto
- Gestión calidad procesos y productos
- Gestión de requisitos
- Gestión y acuerdo con proveedores

Y para el nivel 3 se establecen las siguientes áreas de proceso:

- Análisis y resolución de decisiones
- Gestión integral de proyecto
- Gestión integral de proveedores
- Gestión de equipos
- Entorno organizativo para integración
- Definición de procesos
- Procesos orientados a la organización
- Formación

- Integración de producto
- Desarrollo de requisitos
- Gestión de riesgos
- Solución técnica
- Validación
- Verificación

Debido a que son muchas áreas y el trabajo se haría muy largo entonces solo se tomarán un área de proceso de cada nivel y se verificó a través de una pequeña prueba de acoplamiento entre estas áreas de proceso, el modelo de construcción y/o desarrollo que se utilizó para el producto software y el alcance logrado por los desarrolladores en utilización del modelo, esto se realizó de la siguiente manera:

Primero se escogieron las dos siguientes áreas de proceso, del nivel dos se utilizó la Gestión de Requisitos y para el nivel tres la Verificación.

Ahora bien cada área de proceso posee una serie de objetivos y tareas específicas por esto y al igual de cómo se hizo con el modelo general solo se tomó un objetivo para evaluar y de este una tarea, con esto se verá que tanto acoplamiento o sinergia existe entre el modelo CMMI y el modelo utilizado para SINAI.

Para realizar esta evaluación se verificó el grado de sinergia que existe entre las prácticas específicas de CMMI y algunas actividades de las realizadas en RUP, para entregar un grado de sinergia se definió lo siguiente:

ALTA: Se encontró un alto grado de sinergia entre y las prácticas CMMI – RUP, es decir, se cumple con total especificación cada practica CMMI en RUP.

MEDIA: Se puede encontrar cierto apoyo a la práctica CMMI con cierta actividad de RUP.

BAJA: No existen mecanismos de apoyo a la práctica, o la práctica estaba fuera del alcance de las RUP.

Definido se puede definir una valorización porcentual (ver tabla 13):

Tabla 13. Valorización Porcentual de Calidad

Valorización Porcentual	Valorización Cualitativa
0 % - 49 %	Baja
50 % - 79 %	Media
80 % - 100 %	Alta

Luego de esto se muestra la aplicabilidad de dicha tarea específica en el desarrollo de SINAI.

Para la parte de Gestión de Requisitos se tomaron como medidas de Sinergia lo siguiente:

- ✦ RUP proporciona el apoyo adecuado o los mecanismos necesarios para cumplir la tarea específica.

Y la tarea específica será:

- ✦ Obtener el compromiso de los Requisitos.

Ahora para comprender los detalles de la evaluación ver la tabla 14.

Tabla 14. Detalles de la Evaluación 1

Detalle de Evaluación
Elementos RUP: Flujo de trabajo: Requisitos Detalle de Flujo de trabajo: Administrar el alcance del sistema Actividad: Priorizar casos de uso.
Comentarios: RUP emplea casos de uso y un plan de trabajo conjunto para lo que queda por hacer. El arquitecto siguiendo una lista de prioridades para los casos de uso, solicita un primer artefacto. El equipo del proyecto está involucrado en la revisión de la lista de prioridades basado en los riesgos del proyecto, la disponibilidad de recursos, y las necesidades de los interesados.
Grado de Sinergia: 95%

Para la parte de la Verificación se tomó como medidas de Sinergia lo siguiente:

- ✦ La Verificación se produce con cada iteración.

2.3.4 Conclusiones del proceso de Evaluación

Luego de realizado el proceso de evaluación se pudo obtener una serie de datos que permiten entregar un concepto final de calidad sistémica del producto software, se encontró que la calidad del producto como tal fue de un 68,7 % lo que la ubica en un nivel de calidad aceptable y que el proceso de desarrollo se ubico en un intervalo entre el 80 % y 100 % lo que le entrega un nivel de calidad alto, lo que lleva a concluir que el producto sistémico es de mejora continua, es decir, se tenía claro lo que se iba a construir, pero al momento de construcción se dejaron ciertos detalles que llevaron a la caída de la calidad en el producto final, sin embargo y teniendo conocimiento que este tipo de proyectos no son desarrollados, con total profesionalismo, y que en adición a esto 2 de las características críticas de este se ubicaron en un rango bueno, es posible aceptar este producto como una primera versión del mismo.

3 CONCLUSIONES FINALES

Es claro que un buen análisis de calidad es una de las bases más importantes para la toma de decisiones dentro de una organización que desarrolla software de calidad. Por tal razón, la evaluación de un producto de software permitirá mejorar el control de la calidad de un producto de software.

En este trabajo se han aplicado algunos estándares para predecir el comportamiento de los atributos de calidad de un sistema de software.

Dentro del desarrollo y el campo de aplicación de la Evaluación de Software, un aspecto muy importante tener en cuenta es la planeación de la evaluación dado a que está entrega un proceder claro y centrado del desarrollo de las actividades.

El desarrollo de cualquier software tiene como único objetivo satisfacer una necesidad planteada por un cliente. Pero ¿cómo se puede saber si el producto construido corresponde exactamente con lo que el cliente pidió? Y ¿cómo puede un desarrollador estar seguro, que el producto que ha construido va a funcionar correctamente? para esto es recomendable, que el producto de software sea

evaluado a medida que se va construyendo. Por lo tanto, se hace necesario llevar a cabo, en paralelo al proceso de desarrollo, un proceso de evaluación o comprobación de los distintos productos o modelos que se van generando, en el que participarán desarrolladores y clientes, lo quedará como resultado un mejor producto, aumentando de manera significativa la satisfacción del usuario final.

Evaluación significa medir el estado actual de los procesos de desarrollos de la organización (ya sea algunos proyectos o toda la organización), de manera de conocer las fortalezas, riesgos y debilidades. Los resultados del diagnóstico harán posible la generación de un plan de mejoramiento adecuado. Midiendo y localizando los problemas reales permitirá asignar los recursos a aquellas áreas de mejoras más urgentes, o donde la inversión será más efectiva.

Todo producto de Software requiere una serie de evaluaciones para medir su calidad sin embargo, el realizar con total exactitud estas evaluaciones para entregar conclusiones totalmente objetivas no es muy fácil, se trata de una tarea ardua y dura y es necesario poseer una serie de implementos y herramientas para facilitar esta labor.

4 AGRADECIMIENTOS

Luego de realizado este trabajo, el autor del mismo presenta una serie de agradecimientos a las siguientes personas:

Al Dios Todopoderoso, por siempre encontrarse para el apoyo y sobretodo como guía en el proyecto de vida del autor.

A sus padres el señor Cesar Moreno Pumarejo y la señora Rosa Gómez Torres, por el conocimiento y valores morales entregados.

A la señorita Bibiana Chimá Martínez, novia del autor, por todo el ánimo y apoyo brindado, en la etapa de formación profesional de este.

A la Ingeniera Inés Meriño, por su sencillez y determinación, como Directora del Programa de Ingeniería de Sistemas.

A los profesores, María del Pilar Sales, Johan Robles Solano, Alexander Barón, Eirá Madera Omar Rodríguez y Ernesto Galvis quienes contribuyeron significativamente a la formación profesional del autor.

A mis amigas de toda la vida Kelyn Alcalá, Belkis Chamorro y Evianys Gutiérrez, quienes siempre mantuvieron el buen humor y las ganas de seguir delante del autor.

Y a todos los compañeros del proceso de formación, por su apoyo incondicional y desinteresado durante toda el campo de formación profesional del autor.

5 BIBLIOGRAFIA

I. Sommerville. Software Engineering, Sixth Edition. Pearson Education Limited, Harlow England, 2001.

J. Offutt. Quality Attributes of Web Software Applications. IEEE Software, 0740-7459/02. March/April 2002.

Pressman, 2005. Ingeniería del Software: Un Enfoque Práctico. 6ª Edición. McGraw-Hill, 2005.

Pfleeger, 2002. Ingeniería del Software. Teoría y Práctica. PrenticeHall, 2002.

Sommerville, 2005. Ingeniería del Software. 7ª Edición, Addison-Wesley. Julio 2005.

IEEE Computer Society, 2004: Guide to the Software Engineering Body of Knowledge (SWEBOK), 2004 version. <http://www.swebok.org>

ISO/IEC 9126 –1. Software engineering –Product quality. Part 1: QualityModel. Firstedition. 2001

Capability Maturity Model Integration, <http://www.sei.cmu.edu/cmmi/>

SEI, 2006, Software Engineering Institute, What is CMMI? <http://www.sei.cmu.edu/cmmi/general/general.html>

CURRÍCULO



Cesar Moreno Gómez, estudiante de grado del programa de Ingeniería de Sistemas de la Universidad del Magdalena, actualmente contratista de la Comercializadora Internacional Carcas Limitada (C.I. CARCAS LTDA).