



**UNIVERSIDAD AUTÓNOMA
DE AGUASCALIENTES**

CENTRO DE CIENCIAS BÁSICAS

MAESTRÍA EN CIENCIAS EXACTAS, SISTEMAS Y DE LA INFORMACIÓN

ÁREA: INGENIERÍA DE SOFTWARE

TESIS

**“DESARROLLO DE UN MÉTODO DE EVALUACIÓN DE SOFTWARE QUE
MEJORE LA CALIDAD DEL PRODUCTO FINAL”**

PRESENTA

I.S.C. Abigail Castañeda Santos

DIRECTOR DE TESIS

Dr. Juan Manuel Gómez Reynoso

SINODALES

M.I.T.C. Estela Lizbeth Muñoz Andrade

M.I.T.C. Francisco Guillermo Gutiérrez Nájera

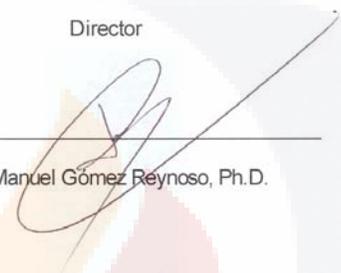
AGUASCALIENTES, AGS. , Julio de 2009

Por este conducto autorizamos al tesista:

Ing. Abigail Castañeda Santos

La impresión de su documento final de Tesis, ya que cumple con los requisitos del contenido y forma exigidos en la Universidad Autónoma de Aguascalientes.

Director



Juan Manuel Gómez Reynoso, Ph.D.

Sinodal



M.I.T.C. Estela Lizbeth Muñoz Andrade

Sinodal



M.I.T.C. Francisco Guillermo Gutiérrez Nájera

AGRADECIMIENTOS

Gracias a mi familia por la confianza depositada y la motivación constante, durante la realización de todo mi posgrado, aún en las situaciones difíciles que nos tocó vivir. De manera particular a mis padres por enseñarme a no darme por vencida y siempre luchar por lo que quiero.

Agradezco a mi director de tesis el Dr. Juan Manuel Gómez Reynoso, el apoyo brindado de manera técnica pero sobre todo moral, para lograr el término de esta investigación de manera exitosa, ya que con su dedicación y paciencia me motivó siempre para el logro, de los objetivos establecidos.

Agradezco también a los maestros Estela Lizbeth Muñoz Andrade y Carlos Arévalo por su apoyo en el avance y revisión de la presente tesis.

Doy gracias también a la Universidad Autónoma de Aguascalientes por la oportunidad brindada para la realización de este posgrado, así como a todas las personas que intervinieron para que esto fuera posible. A la Dirección General de Planeación y al Área de Sistemas en la cual laboro, por las facilidades brindadas para la aplicación de la investigación realizada tanto en tiempo como en recursos.

Y finalmente a todas esas personas que estuvieron siempre apoyándome durante todo el trayecto de todo mi posgrado y proceso de titulación.

RESUMEN

Actualmente el uso de aplicaciones de software dentro de las organizaciones, se ha convertido en una herramienta básica para el buen funcionamiento y desarrollo de las mismas, pero frecuentemente dichas organizaciones se enfrentan a la liberación de aplicaciones de software que no cumplen totalmente con sus necesidades, ya que por ejemplo no satisfacen los requerimientos establecidos inicialmente, tienen baja facilidad de uso o un pobre diseño estético, por lo que se vuelve necesario el establecimiento de un mecanismo de evaluación detallada de los productos de software, por medio del cual se pueda detectar este tipo de problemática antes de su liberación a los usuarios finales, buscando de esta manera lograr mejorar la calidad del producto.

Con la presente investigación se busco enfocar la evaluación de la calidad del producto de software en relación a elementos que los usuarios pueden identificar y que impactan en su percepción de calidad del producto. Para lograr lo anterior se recurrió al análisis e integración de dos de los modelos de calidad, más utilizados dentro de la rama de la Ingeniería de Software, determinando cuáles de los aspectos podían evaluarse de manera tangible y cualitativamente por parte del usuario. Después de lo anterior se diseño un instrumento de evaluación que aplicado iterativamente en base a versiones modificadas de la aplicación, permitiera una mejora incremental de la calidad, al realizar las modificaciones y correcciones de la aplicación en base a los resultados de cada iteración.

De esta forma se logró el desarrollo final de un método de evaluación iterativo de la calidad del producto de software en base a factores de calidad identificables por el usuario e integrando la participación del usuario final en dicha evaluación, buscando garantizar que los productos de software que se liberen de manera formal cuenten con un nivel de calidad óptimo.

Para la validación del instrumento realizado, se utilizó como caso de estudio una aplicación de software utilizada dentro de la Universidad Autónoma de Aguascalientes, con la participación de usuarios reales. Se realizaron 2 iteraciones de evaluación y se utilizó estadística descriptiva para el análisis de los datos recolectados.

INDICE

1.INTRODUCCIÓN 1

2.REVISIÓN DE LA LITERATURA 2

 2.1 Descripción de Teorías Base 2

 2.1.1 ¿Qué es Calidad?..... 2

 2.1.2 Gestión de la Calidad. 3

 2.1.3 Calidad del proceso y del producto. 4

 2.1.4 Aseguramiento y estándares de calidad. 5

 2.1.4.1 Factores de Calidad de McCall 6

 2.1.4.2 Factores de Calidad del Estándar ISO 9126 8

 2.1.5 Planificación de la Calidad. 11

 2.1.6 Control de la Calidad. 12

 2.1.7 Medición y Métricas de Software..... 13

 2.2 Importancia de la Calidad del Software..... 16

3.FORMULACIÓN DEL PROBLEMA DE INVESTIGACIÓN 20

 3.1 Descripción del Problema de Investigación y su Relevancia 20

 3.2 Pregunta de Investigación..... 22

 3.3 Objetivos de la Investigación..... 22

 3.3.1 Objetivo General 22

 3.3.2 Objetivos Específicos 23

 3.4 Hipótesis..... 23

4.DESARROLLO DE LA INVESTIGACIÓN 26

 4.1 Materiales y Métodos 26

 4.2 Estudio Piloto 30

 4.3 Estudio Final..... 31

 4.3.1 Población Objetivo 31

 4.3.2 Selección de la Muestra 31

 4.3.3 Recopilación de la Información. 32

 4.3.3.1 Primera Iteración 33

 4.3.3.2 Segunda Iteración 36

5.ANÁLISIS DE DATOS..... 38

 5.1 Primera Iteración 38

 5.2 Segunda Iteración 45

6.CONCLUSIONES 58

 6.1 Conclusiones Generales 58

 6.2 Limitaciones 61

 6.3 Trabajos Futuros 63

BIBLIOGRAFIA 65

LISTA TABLAS

Tabla 1. Atributos y sub-atributos de Calidad del Estándar ISO 9126 (1991)..... 10

Tabla 2. Clasificación de métricas de software (Adaptada de Fenton, 1991) 16

Tabla 3. Equivalencias McCall – ISO 9126 29

Tabla 4. Factores para evaluar la calidad de una aplicación de software desde el punto de vista del usuario 29

Tabla 5. Tabla de Frecuencia de Edad - Primera iteración 33

Tabla 6. Tabla de Frecuencia de Género - Primera iteración 34

Tabla 7. Tabla de Frecuencia de Nivel de Estudios - Primera iteración..... 34

Tabla 8. Tabla de Frecuencia de Tipo de Puesto - Primera iteración 35

Tabla 9. Tabla de Frecuencia de Área de Estudios - Primera iteración 35

Tabla 10. Tabla de Frecuencia de Edad - Segunda iteración 36

Tabla 11. Tabla de Frecuencia de Género - Segunda iteración..... 36

Tabla 12. Tabla de Frecuencia de Nivel de Estudios - Segunda iteración..... 37

Tabla 13. Tabla de Frecuencia de Tipo de Puesto - Segunda iteración 37

Tabla 14. Tabla de Frecuencia de Área de Estudios - Segunda iteración 37

Tabla 15. Estadísticas e Histograma del Factor Conformidad - Primera iteración..... 38

Tabla 16. Estadísticas e Histograma del Factor Exactitud - Primera iteración..... 39

Tabla 17. Estadísticas e Histograma del Factor Seguridad- Primera iteración 40

Tabla 18. Estadísticas e Histograma del Factor Madurez - Primera Iteración 40

Tabla 19. Estadísticas e Histograma del Factor Tolerancia a Fallas - Primera Iteración41

Tabla 20. Estadísticas e Histograma del Factor Recuperabilidad - Primera Iteración ... 42

Tabla 21. Estadísticas e Histograma del Factor Comprensibilidad - Primera Iteración . 42

Tabla 22. Estadísticas e Histograma del Factor Facilidad de Aprendizaje - Primera Iteración 43

Tabla 23. Estadísticas e Histograma del Factor Operabilidad - Primera Iteración..... 44

Tabla 24. Estadísticas e Histograma del Factor Tiempo de Respuesta - Primera Iteración 44

Tabla 25. Estadísticas e Histograma del Factor Adaptabilidad - Primera Iteración 45

Tabla 26. Estadísticas comparativas entre iteraciones del Factor Conformidad..... 46

Tabla 27. Prueba de Muestras Independientes del Factor Conformidad 46

Tabla 28. Estadísticas comparativas entre iteraciones del Factor Exactitud 47

Tabla 29. Prueba de Muestras Independientes del Factor Exactitud..... 47

Tabla 30. Estadísticas comparativas entre iteraciones del Factor Seguridad..... 48

Tabla 31. Prueba de Muestras Independientes del Factor Seguridad 48

Tabla 32. Estadísticas comparativas entre iteraciones del Factor Madurez 49

Tabla 33. Prueba de Muestras Independientes del Factor Madurez..... 49

Tabla 34. Estadísticas comparativas entre iteraciones del Factor Tolerancia a Fallas.. 50

Tabla 35. Prueba de Muestras Independientes del Factor Tolerancia a Fallas 50

Tabla 36. Estadísticas comparativas entre iteraciones del Factor Recuperabilidad 51

Tabla 37. Prueba de Muestras Independientes del Factor Recuperabilidad..... 51

Tabla 38. Estadísticas comparativas entre iteraciones del Factor Comprensibilidad 52

Tabla 39. Prueba de Muestras Independientes del Factor Comprensibilidad..... 52

Tabla 40. Estadísticas comparativas entre iteraciones del Factor Facilidad de Aprendizaje 53

Tabla 41. Prueba de Muestras Independientes del Factor de Facilidad de Aprendizaje54

Tabla 42. Estadísticas comparativas entre iteraciones del Factor Operabilidad..... 54

Tabla 43. Prueba de Muestras Independientes del Factor de Operabilidad 55

Tabla 44. Estadísticas comparativas entre iteraciones del Factor Tiempo de Respuesta 55

Tabla 45. Prueba de Muestras Independientes del Factor Tiempo de Respuesta 56

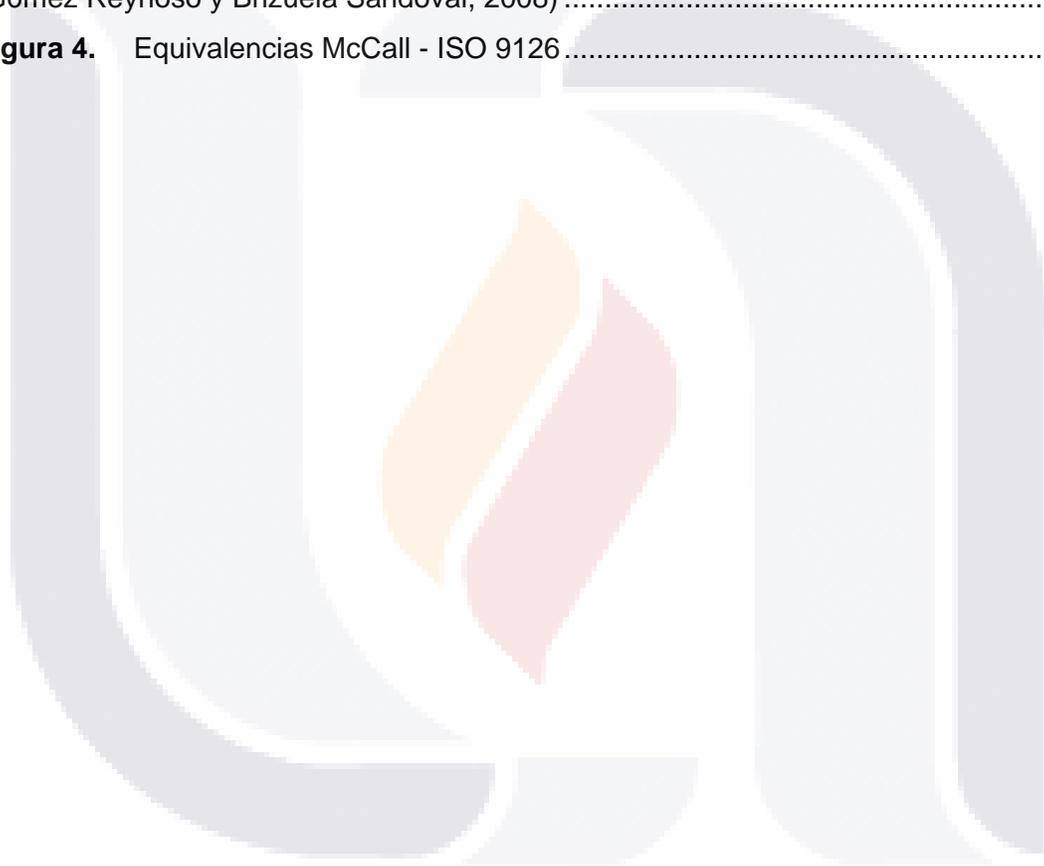
Tabla 46. Estadísticas comparativas entre iteraciones del Factor Adaptabilidad 56

Tabla 47. Prueba de Muestras Independientes del Factor Adaptabilidad..... 57

Tabla 48. Resultados concentrados de la Prueba de Hipótesis..... 57

LISTA DE FIGURAS

Figura 1. Factores que afectan la calidad del producto (adaptado de Sommerville, 2002)	4
Figura 2. Factores de calidad del software de McCall (adaptado de Sommerville, 2002)	7
Figura 3. Modelo de desarrollo base del método de evaluación de calidad del producto (Gómez Reynoso y Brizuela Sandoval, 2008)	27
Figura 4. Equivalencias McCall - ISO 9126	28



1. INTRODUCCIÓN

La finalidad tangible de toda aplicación de software es cubrir una necesidad del cliente. Por lo que al desarrollar un producto de software, se debe buscar que este realmente cumpla con tal objetivo. Esto puede impactar en la aceptación y satisfacción total del cliente.

Por otro lado, el uso de aplicaciones de software dentro de las organizaciones se ha convertido en una herramienta básica para el buen funcionamiento y desarrollo de las mismas. Muchas veces estas organizaciones se enfrentan a la liberación de productos de software que no cumplen totalmente con sus necesidades y esto puede ir desde factores como no cumplir con los requerimientos establecidos hasta factores como su facilidad de uso o diseño estético. Debido a esto surge la necesidad de que los productos de software pasen por una evaluación detallada, en la cual se pueden detectar problemáticas antes de su liberación a los usuarios finales. De esta manera, se puede mejorar la calidad del producto final y cumplir con la finalidad básica de las aplicaciones de software: satisfacer las necesidades del usuario final.

El proceso de medición de la calidad del software debe implementarse en todo el ciclo de vida del mismo, pero es importante destacar la evaluación del producto terminado, ya que el tener un alto nivel de calidad en el proceso de desarrollo, no garantiza productos finales de calidad que cumplan realmente con las expectativas y necesidades del cliente.

Actualmente, existen diferentes modelos de calidad que indican cuales son las cualidades deseables para determinar la calidad de un producto de software. Sin embargo, tales modelos no establecen mecanismos definidos para su evaluación, por lo que la presente investigación propone integrar y definir en base a estos modelos, los principales factores que se pueden analizar y la forma de evaluarlos, considerando principalmente la integración de los usuarios finales en dicha evaluación, todo esto para contribuir a lograr aplicaciones de software de alta calidad.

2. REVISIÓN DE LA LITERATURA

2.1 Descripción de Teorías Base

2.1.1 ¿Qué es Calidad?

El ISO 9000:2000 define calidad como: “Grado en el que un conjunto de características inherentes cumple con los requisitos”. Donde una característica es un rasgo diferenciador y un requisito es una necesidad o expectativa establecida, generalmente implícita u obligatoria. Juran (1990) define calidad como la adecuación para el uso, satisfaciendo las necesidades del cliente.

Por otro lado Eigenbaum (1986) argumenta que la calidad es un sistema eficaz que permite integrar los esfuerzos de mejora de la calidad de los distintos grupos de una organización, esto con el fin de proporcionar productos y servicios a niveles que permitan la satisfacción del cliente. Por otro lado, Dolado y Fernández (2000) definen de manera genérica la calidad como la valoración por los consumidores de la superioridad o excelencia de un bien o servicio.

Adicionalmente, Garvin (1988) distingue tres clases de calidad: subjetiva, objetiva y rentable. La calidad subjetiva o externa es aquella relacionada con los atributos de los bienes y servicios, así como su valor asignado por el cliente. La calidad objetiva o interna alude al desarrollo del bien o servicio y a su grado de cumplimiento respecto a las especificaciones previamente establecidas. Por último, la calidad rentable hace referencia al efecto global de la calidad sobre los costos de la empresa, esto es, evalúa si las características de diseño (calidad objetiva) son adecuadas a las necesidades del cliente y si ambas condiciones se cumplen de manera eficiente.

Dentro de la rama de la Ingeniería de Software, la calidad del software se define como “la concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente” (Pressman 2005, p.772). De igual manera, este autor sugiere que la calidad del software debe

integrarse en todo el ciclo de vida del mismo y que para la implantación del control de calidad en el desarrollo de software se deben realizar las siguientes actividades:

- Aplicación de metodología y técnicas de desarrollo
- Reutilización de procesos de revisión formales
- Prueba del software
- Ajustes a los estándares de desarrollo,
- Control de cambios, mediciones y recopilación de información
- Gestión de informes sobre el control de calidad.

Pero, para poder implementar la calidad dentro de una organización o actividad determinada, se necesita llevar a cabo la Gestión de la Calidad, aspecto que se detalla en el apartado siguiente.

2.1.2 Gestión de la Calidad.

La Gestión de la Calidad consiste en la realización de actividades coordinadas que permiten dirigir y controlar una organización en lo relativo a la calidad (ISO 9000:2000). Esta se basa en la determinación y aplicación de las políticas de calidad de la empresa (objetivos y directrices generales). La Gestión de la Calidad se aplica normalmente a nivel empresa, pero también puede haber una gestión de la calidad dentro del desarrollo de cada proyecto de software.

Sommerville (2002) indica que la gestión de la calidad del software se estructura en tres actividades principales, las cuales son:

1. *Aseguramiento de la Calidad.* Esta actividad consiste en el establecimiento de un marco de trabajo de procedimientos y estándares organizacionales que conducen a software de alta calidad.
2. *Planeación de la Calidad.* Esta actividad consiste en la selección de procedimientos y estándares adecuados a partir de este marco de trabajo y la adaptación de estos para un proyecto de software específico.

3. *Control de la Calidad.* Esta actividad consiste en la definición y promulgación de los procesos que aseguran que los procedimientos y estándares para la calidad del proyecto son seguidos por el equipo de desarrollo de software.

De manera general, para lograr la calidad del software existen dos componentes imprescindibles, los cuales deben intersectarse para que se logren aplicaciones de software con calidad suficiente. Estos componentes son: 1) calidad del proceso y 2) calidad del producto. Un software que solo cuide una de las dos tiene el enorme riesgo de fallar miserablemente. A continuación se describen ampliamente cada una de ellas.

2.1.3 Calidad del proceso y del producto.

Pressman (2005) argumenta que una suposición subyacente de la administración de la calidad, es que la calidad del proceso de desarrollo afecta directamente a la calidad de los productos a entregar.

Sommerville (2002) indica que el mejoramiento de la calidad se centra en identificar buenos productos de calidad, examinar el proceso utilizado para desarrollar estos productos y después generalizar estos procesos para que se apliquen a varios proyectos. Sin embargo, la relación entre el proceso de software y la calidad del producto es compleja. Cambiar el proceso no siempre conduce a mejorar la calidad del producto. Adicionalmente, él menciona que existen cuatro factores que afectan la calidad del producto y estos se muestran en la Figura 1.



Figura 1. Factores que afectan la calidad del producto (adaptado de Sommerville, 2002)

Este mismo autor, indica que la administración de la calidad del proceso comprende:

1. Definir estándares de proceso como las revisiones a realizar, cuándo llevarlas a cabo, etc.
2. Supervisar el proceso de desarrollo para asegurar que se sigan los estándares
3. Hacer informes del proceso para la administración del proyecto y para el comprador del software.

2.1.4 Aseguramiento y estándares de calidad.

Según la Norma ISO 9000:2000, el aseguramiento de la calidad es la parte de la gestión de la calidad orientada a proporcionar confianza en que se cumplirán los requisitos de calidad.

Pressman (2005) menciona que el aseguramiento de la calidad del software (ACS) es el conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza que el software satisfará los requisitos establecidos de calidad. Para lograr el ACS es necesaria la utilización tanto de modelos como de estándares establecidos, o bien, es necesario el establecimiento de un nuevo modelo en base a la combinación de los existentes.

Los modelos de calidad son aquellos documentos que integran la mayor parte de las mejores prácticas, proponen temas de administración en los que cada organización debe hacer énfasis, integran diferentes prácticas dirigidas a los procesos clave y permiten medir los avances en calidad (Piattini, 2003).

Los estándares de calidad son aquellos que permiten definir un conjunto de criterios de desarrollo que guían la forma en que se aplica la Ingeniería del Software. Los estándares suministran los medios para que todos los procesos se realicen de la misma forma y son una guía para lograr la productividad y la calidad (Piattini, 2003)

Sommerville (2002) menciona que existen 2 tipos de estándares que se establecen como parte del proceso de aseguramiento de la calidad, los cuales son:

- Estándares del producto. Son aquellos que aplican al producto de software a desarrollar. Incluyen estándares de documentos tales como la estructura del documento de requerimientos a producir, estándares de documentación como encabezados estándar de comentarios para una definición de clases de objetos, y estándares de codificación, que definen como utilizar un lenguaje de programación.
- Estándares del proceso. Son aquellos que definen los procesos a seguir durante el desarrollo de software. Incluyen definiciones de procesos de especificación, de diseño, de validación y una descripción de los documentos a generar en el transcurso de estos procesos.

De acuerdo a Pressman (2005) la calidad del software es una compleja combinación de factores que variarán entre las diferentes aplicaciones y los clientes que las solicitan. Dichos factores que afectan a la calidad del software se dividen en 2 grandes grupos, los cuales son:

1. Los que se miden directamente (por ejemplo: número de defectos descubiertos durante las pruebas).
2. Los que solo se miden indirectamente (por ejemplo: facilidad de uso o facilidad de mantenimiento).

Es importante que en ambos tipos de factores se lleve a cabo una medición, la cual básicamente consistirá en comparar el software con algún conjunto de datos (por ejemplo, estándares relacionados) y obtener así algún indicio sobre la calidad.

2.1.4.1 Factores de Calidad de McCall

McCall, Richards y Walters (1977) propusieron una clasificación de los factores que afectan la calidad del software. Dichos factores (ver Figura 2) se concentran en tres

aspectos importantes de un producto de software: sus características operativas, su capacidad para experimentar cambios y su capacidad para adaptarse a nuevos entornos.



Figura 2. Factores de calidad del software de McCall (adaptado de Sommerville, 2002)

Sommerville (2002) describe cada uno de los componentes del triángulo de McCall de la siguiente manera:

1. Revisión del producto. El cual se divide en:

- *Facilidad de Mantenimiento.* Es el esfuerzo necesario para localizar y corregir un error en un programa.
- *Flexibilidad.* Es el esfuerzo necesario para modificar un programa en operación.
- *Facilidad de Prueba.* Es el esfuerzo que demanda probar un programa con el fin de asegurar que realiza su función.

2. Transición del Producto. El cual se divide en:

- *Portabilidad.* Es el esfuerzo necesario para transferir el programa de un entorno de hardware o software a otro.

- *Facilidad de Reutilización.* Es el grado en que un programa (o partes de él) puede reutilizarse en otras aplicaciones (en relación con el empaquetamiento y alcance de las funciones que realiza el programa).
- *Interoperabilidad.* Es el esfuerzo necesario para acoplar un sistema con otro.

3. Operación del Producto. El cual se divide en:

- *Corrección.* Es el grado en que un programa cumple con su especificación y satisface los objetivos que propuso el cliente.
- *Confiabilidad.* Es el grado en que se esperaría que un programa desempeñe su función con la precisión requerida.
- *Eficiencia.* Es la cantidad de código y de recursos de cómputo necesarios para que un programa realice su función.
- *Integridad.* Es el grado de control sobre el acceso al software o los datos por partes de las personas no autorizadas.
- *Facilidad de Uso.* Es el esfuerzo necesario para aprender, operar y preparar los datos de entrada de un programa e interpretar la salida.

Todos estos factores son muy importantes e imprescindibles y medibles de formas diferentes.

2.1.4.2 Factores de Calidad del Estándar ISO 9126

El ISO 9126 (1991) es un estándar que se desarrolló como un intento por identificar los atributos de calidad para el software. Dicho estándar identifica seis atributos claves de la calidad, los cuales se describen a continuación:

1. *Funcionalidad.* Es el grado en que el software satisface las necesidades que indican los siguientes sub-atributos: idoneidad, exactitud, interoperabilidad, cumplimiento y seguridad.

2. *Confiabilidad*. Es la cantidad de tiempo en que el software está disponible para usar según los siguientes sub-atributos: madurez, tolerancia a fallas y facilidad de recuperación.
3. *Facilidad de uso*. Es la facilidad con que se usa el software de acuerdo con los siguientes sub-atributos: facilidad de comprensión, facilidad de aprendizaje y operabilidad.
4. *Eficiencia*. Es el grado en que el software emplea en forma óptima los recursos del sistema, como lo indican los siguientes sub-atributos: comportamiento en el tiempo, comportamiento de los recursos.
5. *Facilidad de mantenimiento*. Es la facilidad con que se repara el software de acuerdo con los siguientes sub-atributos: facilidad de análisis, facilidad de cambio, estabilidad y facilidad de prueba.
6. *Portabilidad*. Es la facilidad con que se lleva el software de un entorno a otro según los siguientes sub-atributos: adaptabilidad, facilidad para instalarse, cumplimiento, facilidad para reemplazarse.

En la Tabla 1, se describe brevemente los sub-atributos que componen a cada uno de los atributos de calidad mencionados.

Atributo	Sub-atributos	Definición
Funcionalidad	Conformidad	Es la capacidad del software de proporcionar un conjunto de funciones adecuado para tareas específicas
	Exactitud	Es la capacidad del software para proporcionar resultados correctos o efectos convenidos con el grado necesario de precisión.
	Interoperabilidad	Es la capacidad del software para interactuar con uno o más sistemas específicos
	Cumplimiento	Es la capacidad del software para adherirse a estándares, normas, convenciones o regulaciones en legislaciones y prescripciones similares relacionadas con la funcionalidad
	Seguridad	Es la capacidad del software para proteger la información y los datos, de accesos no autorizados ya sea accidentales o deliberados.
Confiabilidad	Madurez	Es la capacidad del software para evitar fallas como resultado de fallas en el software.
	Tolerancia a fallos	Es la capacidad del software para mantener un nivel de desempeño específico en caso de fallas o de que se infrinjan sus interfaces especificadas.
	Recuperabilidad	Es la capacidad del software para restablecer su nivel de desempeño y recuperar los datos directamente afectados en caso de falla, así como con el tiempo y el esfuerzo necesario para ello.
Facilidad de uso	Comprensibilidad	Es la capacidad del software para permitirle al usuario entender si el software es conveniente y cómo puede ser utilizado
	Facilidad de aprendizaje	Es la capacidad del software para permitirle al usuario aprender su aplicación
	Operabilidad	Es la capacidad del software para permitirle al usuario su operación y control.
Eficiencia	Comportamiento en el tiempo	Es la capacidad del software para proporcionar tiempos adecuados de respuesta, de procesamiento y de tasas de eficiencia en el desempeño de su función
	Comportamiento de los recursos	Es la capacidad del software para utilizar una cantidad apropiada de recursos y su duración en el desempeño de su función
Facilidad de mantenimiento	Facilidad de análisis	Es la capacidad del software para ser diagnosticado por deficiencias o causas de fallas, o para identificar las partes que se deben modificar
	Facilidad de cambio	Es la capacidad del software para permitir modificaciones, para eliminar fallas o para cambios de entorno.
	Estabilidad	Es la capacidad del software para evitar efectos no esperados debido a modificaciones.
	Facilidad de prueba	Es la capacidad del software para permitir validar las modificaciones realizadas
Portabilidad	Adaptabilidad	Es la capacidad del software para adaptarse a diferentes ambientes especificados sin aplicar acciones o medios distintos a los ofrecidos para este propósito considerados por el software.
	Capacidad de instalación	Es la capacidad del software para ser instalado en un entorno especificado
	Conformidad	Es la capacidad del software para adherirse a estándares o convenciones relativas a la portabilidad.
	Capacidad de reemplazamiento	Es la capacidad del software para ser utilizado en lugar de otro producto de software especificado para el mismo propósito en el mismo ambiente.

Tabla 1. Atributos y sub-atributos de Calidad del Estándar ISO 9126 (1991)

2.1.5 Planificación de la Calidad.

Según la Norma ISO 9000:2000, la planificación de la calidad es la parte de la gestión de la calidad enfocada al establecimiento de los objetivos de la calidad y a la especificación de los procesos operativos necesarios y de los recursos relacionados para cumplir los objetivos de calidad.

El estándar ISO/IEC 90003:2004 describe la Planificación de la Calidad del Software (PCS) como la parte de la Gestión de la Calidad encargada de realizar el proceso administrativo de desarrollar y mantener una relación entre los objetivos y recursos de la organización; y las oportunidades cambiantes del mercado.

La PCS inicia en las primeras etapas del proceso de software. Un plan de calidad define la calidad del producto deseado, define como valorar ésta calidad y por lo tanto lo que significa software de “alta calidad”. El resultado de un proceso de planeación de la calidad es un plan de calidad del proyecto (Sommerville, 2002).

Humphrey (1989) sugiere la siguiente estructura para un plan de calidad:

1. *Introducción al Producto.* Contiene la descripción del producto, el mercado al que se dirige y las expectativas de calidad del producto.
2. *Planes del producto.* Contiene las fechas de terminación del producto y las responsabilidades importantes juntos con los planes de su distribución y servicio
3. *Descripciones del proceso.* Contiene los procesos de desarrollo y servicios a utilizar para el desarrollo y administración del producto.
4. *Objetivos de Calidad.* Contiene las metas y planes de calidad para el producto, incluye una identificación y justificación de los atributos de calidad importantes del producto.
5. *Riesgo y administración de riesgos:* Contiene los riesgos clave que podrían afectar la calidad del producto y las acciones para abordar estos riesgos.

El plan de calidad define los atributos de calidad más importantes del producto a ser desarrollado y define el proceso de evaluación de la calidad (Sommerville, 2002).

2.1.6 Control de la Calidad.

Según la Norma ISO 9000:2000, el control de la calidad es la parte de la gestión de la calidad orientada al cumplimiento de los requisitos de la calidad.

El Control de la Calidad del Software (CCS) son las técnicas y actividades de carácter operativo, utilizadas para satisfacer los requisitos relativos a la calidad, centradas en 2 objetivos fundamentales: 1) mantener bajo control un proceso y 2) eliminar las causas de los defectos en las diferentes fases del ciclo de vida (Pressman, 2005). El proceso de CCS tiene su propio conjunto de procedimientos e informes a utilizar durante el desarrollo de software.

De acuerdo a Sommerville (2002), el CCS implica vigilar el proceso de desarrollo de software para asegurar que siga los procedimientos de aseguramiento y estándares de calidad. Los productos resultantes de un proceso de software, se comprueban contra los estándares definidos del proyecto en el proceso de CCS.

Sommerville (2002) menciona también que existen 2 enfoques complementarios para el CCS, los cuales son:

1. Revisiones de la calidad en las que el software, su documentación y los procesos utilizados para producir ese software son revisados por un grupo de personas. Son responsables de comprobar que se han seguido los estándares del proyecto y que el software y los documentos estén acordes a esos estándares. Toman las desviaciones de los estándares y las ponen a consideración de la administración del proyecto.
2. Valoración automática del software en la que el software y los documentos producidos se procesan por algún programa y se comparan con los estándares que aplican a ese proyecto de desarrollo en particular. Esta valoración automática comprende una medida cuantitativa de algunos atributos del software.

2.1.7 Medición y Métricas de Software

La medición del software se refiere a derivar un valor numérico para algún atributo de un producto de software o un proceso del software (Sommerville, 2002).

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

La medición es esencial, si es que se desea realmente conseguir la calidad en software. Es por eso que existen distintos tipos de métricas para poder evaluar, mejorar y clasificar al software final, las cuales serán manejadas dependiendo del entorno de desarrollo del software al cual pretendan orientarse (González, 2001).

Pressman (2005) explica que el proceso de medición se debe llevar a cabo mediante cinco actividades:

1. *Formulación*: Es la derivación de medidas y métricas del software apropiadas para la representación de software en cuestión.
2. *Recolección*: Es el mecanismo con el que se acumulan datos necesarios para obtener las métricas formuladas.
3. *Análisis*: Es el cálculo de las métricas y la aplicación de herramientas matemáticas.
4. *Interpretación*: La evaluación de los resultados de las métricas en un esfuerzo por conocer mejor la calidad de la representación.
5. *Realimentación*: Son las recomendaciones obtenidas de la interpretación de métricas técnicas transmitidas al equipo de software.

A continuación se proporcionan algunas definiciones de métrica de software:

1. Una métrica de software es una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado (IEEE, 1994).
2. Una métrica de software es cualquier tipo de medida relacionada con un sistema, proceso o documentación de software (Sommerville, 2002).

Ejiogu (1991) define un conjunto de atributos para obtener métricas efectivas del software, por lo tanto la métrica obtenida y las medidas que conducen a ello deben cumplir con las siguientes características fundamentales:

- *Simples y Calculables.* Debe ser relativamente fácil de aprender a derivar la métrica, y su cálculo no debe exigir cantidades anormales de tiempo o esfuerzo
- *Empírica o intuitivamente persuasivas.* La métrica debe satisfacer las nociones intuitivas del ingeniero acerca del atributo del producto que se está construyendo
- *Consistentes y objetivas.* La métrica siempre debe arrojar resultados que no permitan ambigüedad alguna
- *Consistentes en el uso de unidades y dimensiones.* El cálculo matemático de la métrica debe emplear medidas que no lleven combinaciones extrañas de unidades
- *Independientes del lenguaje de programación.* Las métricas deben basarse en el modelo de análisis o diseño, o en la estructura del propio programa
- *Mecanismos efectivos para la retroalimentación de alta calidad.* La métrica debe llevar a un producto final de la más alta calidad

Además, literatura previa (Pressman 2005) explica que la recolección y análisis de las métricas debe cumplir los siguientes principios:

1. Siempre que sea posible, la recolección de datos y el análisis de los mismos debe ser automatizado.
2. Deben aplicarse técnicas estadísticas válidas para establecer la relación entre los atributos internos del producto y las características externas de calidad.
3. Deben utilizarse directrices interpretativas y recomendaciones para establecer cada métrica.

De acuerdo a Dolado y Fernández (2000) las entidades cuyos atributos se pueden interesar medir en el software, se clasifican en 3 grandes grupos de los cuales se derivan sus respectivas métricas:

- *Procesos.* Son un conjunto de actividades relacionadas con el software que habitualmente incluyen el factor tiempo. Se trata de ejemplos de procesos de

actividades bien definidas, como crear un documento de especificación o integrar el sistema. Las métricas de proceso cuantifican atributos como tiempo, esfuerzo, número de incidencias, etc. Por ejemplo: esfuerzo y tiempo promedio requeridos para reparar los errores reportados.

- *Productos*. Son el resultado (programas, documentación y otras entregas) de la actuación de los procesos. Las métricas del producto son medidas del producto de software e incluyen el tamaño del producto (por ejemplo número de líneas de código), la complejidad de la estructura de datos y el tipo de software (comercial, científico, de sistemas, etc.).
- *Recursos*. Se trata del conjunto de elementos que son entradas para la producción del software. Ejemplos de recursos son el personal, las herramientas y los métodos. Ejemplo de métricas de los recursos para los niveles de experiencia del personal, son los años que un grupo ha estado usando un lenguaje de programación, años de experiencia de construcción de software comercial.

Por otra parte, Fenton (1991) argumente que al medir el software, se suele hacer la siguiente clasificación de atributos:

- Atributos internos. Son aquellos que se pueden medir en términos de la propia entidad (producto, proceso o recurso).
- Atributos externos. Son los atributos que solo se pueden medir con respecto al entorno de la entidad.

Tomando en cuenta todo lo anterior, él realiza la clasificación de métricas mostradas en la Tabla 2.

Entidad	Atributo interno	Atributo externo
<i>Producto</i>		
Especificaciones	Tamaño, reutilización, modularidad, redundancia, funcionalidad, sintaxis, corrección	Fácil de comprender, fácil mantenimiento
Diseño	Tamaño, reutilización, modularidad, cohesión, acoplamiento, funcionalidad	Calidad, complejidad fácil mantenimiento
Código	Tamaño, reutilización, modularidad, cohesión, acoplamiento, funcionalidad, complejidad Algorítmica, etc.	Calidad, complejidad fácil mantenimiento Fiabilidad fácil uso
Test	Tamaño cobertura	Calidad
<i>Procesos y Recursos</i>		
Construcción de Especificaciones	Tº de esfuerzo, nº de Cambios	Calidad, Costo, Estabilidad
Diseño	Tº de esfuerzo, nº de errores en las especificaciones	Efectividad-Costos
Test	Tº, Esfuerzo, Nº de errores Encontrados	Efectividad-Costos, Fiabilidad, Estabilidad
Personal	Edad, Costo	Productividad, Experiencia
Equipos	Tamaño, Nivel de Comunicación	Productividad, Calidad
Software	Precio, tamaño	Usabilidad, Fiabilidad.
Hardware	Precio, Memoria., Rapidez	Fiabilidad

Tabla 2. Clasificación de métricas de software (Adaptada de Fenton, 1991)

2.2 Importancia de la Calidad del Software

Pressman (2005) indica que en el desarrollo de software, la calidad del diseño incluye requisitos, especificaciones y el diseño del sistema. La calidad de concordancia es un tema enfocado principalmente en el diseño de la implementación. Si ésta sigue el diseño y el sistema resultante satisface sus requisitos y metas de desempeño, la calidad de concordancia es alta.

Pero, ¿la calidad del diseño y la calidad de concordancia son los únicos temas que deben considerar los ingenieros de software? Glass (1998) argumenta que es conveniente una relación más intuitiva dando la siguiente fórmula:

Satisfacción del usuario = producto manejable + buena calidad + entrega dentro presupuesto y tiempo

Dado lo anterior, Glass (1998) afirma que la calidad es importante, pero que si el usuario no está satisfecho minimiza dicha importancia. De Marco (1999) también refuerza ésta idea al afirmar que la calidad de un producto es una función de cuanto cambia el mundo para mejorar.

Dentro de la Ingeniería de Software, la Calidad del Software es una de las ramas de mayor importancia. Sobre todo por el acelerado crecimiento que ha tenido la industria del software en los últimos años, lo que ha llevado a una búsqueda constante de técnicas que permitan obtener cada vez mejores productos de software.

La Calidad del Software puede enfocarse tanto al proceso como al producto, por lo que su determinación puede darse como una combinación de ambos enfoques.

Dolado y Fernández (2000) argumentan lo siguiente:

- La preocupación por la calidad en el ámbito del software se ha centrado en asegurarlo a través de los procesos de acuerdo con los estándares tradicionales adoptados en otras ramas de la ingeniería. Este enfoque puede proporcionar buenos resultados en las industrias en las que existe un paralelismo evidente entre los criterios de valoración del usuario y los factores técnicos que lo determinan, pero no parece adecuado para aquellas en las que los criterios determinantes de la elección del cliente difieren bastante de los factores técnicos, no son fácilmente identificables o resultan de una combinación aún desconocida de atributos técnicos que solo considerados en conjunto adquieren un significado funcional para el cliente.
- El empleo de factores en la industria del software como la ingeniería humana, la transición y la explotación del producto o la facilidad de prueba, que proponen algunos de los modelos más aceptados y conocidos, no parecen ser los criterios de evaluación más apropiados para valorar la calidad desde la perspectiva del cliente, ya que desde este punto de vista, se asume que el alcance de la excelencia, en lo que a calidad se refiere, es la consecuencia de un proceso de desarrollo enfocado a su consecución. Sin embargo, la calidad percibida por el cliente remite a la confirmación de sus expectativas sobre el producto desarrollado, mientras que la calidad técnica por

definición, ignora cualquier factor no técnico, siendo que el cliente es quien determina el nivel de excelencia de referencia.

- Disponer de un instrumento que permita medir la satisfacción del cliente puede resultar una buena solución al problema de los productos de mala calidad técnica y un adecuado instrumento para orientar a los clientes en su proceso de selección de los bienes y servicios de software.

Por otro lado, Amasaki et al. (2005) afirman que un software con calidad ayuda a reducir los costos de mantenimiento así como a satisfacer los requerimientos de los usuarios.

Todo lo anterior reafirma la importancia de la calidad del software, pero uno de los retos más importantes dentro de la ingeniería de software, es cómo lograr y evaluar dicha calidad.

Sommerville (2002) menciona que existe un amplio rango de atributos de calidad potenciales del software que se deben considerar durante el proceso de planeación de la calidad. Como en general no es posible optimizar todos estos atributos para un sistema, una parte importante de la planeación de la calidad es seleccionar los atributos de calidad importantes y planear como alcanzarlos.

A este respecto Zayaraz et al. (2005) argumentan también que la calidad del software tiene diversos aspectos pero la calidad general puede expresarse en términos de la combinación de varios de ellos.

Por otro lado, además de definirse qué aspectos o factores pueden determinar la calidad del software, se debe determinar también cómo van a ser evaluados tales factores para determinar el nivel de calidad del software y de igual manera cómo lograr mejorar ese nivel de calidad.

Ramani (2007) propone dos técnicas de evaluación del software desde el punto de vista de calidad en el uso: 1) la Evaluación basada en cuestionarios y 2) la Evaluación basada en métricas. De igual forma, este autor menciona que incrementando la calidad en el uso

se pueden obtener diversos beneficios adicionales tales como: incremento de la eficiencia, mejora de la productividad, reducción de errores, reducción en la capacitación y mejora de la aceptación.

Otras investigaciones pasadas (Amasaki, Yoshitomi et al. 2005; Zayaraz, Thambidurai et al. 2005) indican que la calidad del software se mejora reduciendo el número de fallas mediante pruebas exhaustivas.

De igual forma, Sommerville (2002) explica que en investigaciones anteriores (principalmente las realizadas por Deming en 1989) han tratado de reducir la incertidumbre en el control de calidad mediante el control estadístico de la misma, la cual se basa en medir el número de defectos en los productos y relacionar estos defectos con el proceso. El proceso se mejora con el propósito de reducir el número de defectos en el producto. Además, el proceso se mejora hasta que sea repetible, es decir, hasta que los resultados del proceso sean predecibles y el número de defectos se reduzca. Posteriormente, se estandariza e inicia un ciclo de mejora adicional.

Con todo lo anterior, se enfatiza el impacto de la calidad del software y el porqué esta área se ha convertido en una de las ramas relevantes dentro de la Ingeniería de Software, sobre todo por el acelerado crecimiento que ha tenido la industria del software en los últimos años, lo que ha llevado a una búsqueda constante de técnicas que permitan obtener cada vez mejores productos de software y es claro que uno de los retos más importantes es como lograr y evaluar dicha calidad.

3. FORMULACIÓN DEL PROBLEMA DE INVESTIGACIÓN

3.1 Descripción del Problema de Investigación y su Relevancia

Actualmente, en la mayoría de los proyectos de software se realizan acciones para la implementación de planes de calidad en el proyecto, lo cual proporciona una manera ordenada y controlada para llevar a cabo el proceso de desarrollo. Pero esto no garantiza que el producto final será de calidad. Sobre todo porque de manera tangible está evaluación depende en gran parte de la visión del cliente o usuario final.

Aunado a lo anterior, el éxito de una aplicación o producto de software, más que depender de un nivel de calidad evaluado de manera interna en el proyecto de desarrollo, va a estar relacionado con el grado de aceptación y satisfacción del cliente, en base a si el producto cumple o no tanto con sus necesidades funcionales como con sus necesidades operativas. Por ejemplo, los usuarios asignan una buena calidad al software si hace lo que ellos desean, en una forma que sea fácil de aprender y fácil de utilizar (Pfleeger, 2002).

No obstante, muchos de los principales riesgos en el desarrollo de software se dan precisamente porque quienes prueban los productos son los desarrolladores y éstos no siempre están familiarizados con las áreas de negocio de las aplicaciones que se están desarrollando. Por otro lado, los usuarios finales son quienes realmente saben cuáles son sus necesidades y los resultados que esperan obtener.

Este aspecto humano de la Ingeniería de Software puede ser el más crítico, ya que un sistema puede tener un desempeño excelente realizando una función, pero si los usuarios no pueden entender cómo usarlo, el sistema es un fracaso (Pfleeger, 2002).

Con lo anterior, se puede ver la importancia de que lo usuarios sean integrados de una manera formal dentro del ciclo de desarrollo de software, tanto en la definición de planes de prueba como en la evaluación del producto.

Por otro lado, dentro del área de la Ingeniería de Software se han definido diferentes modelos de calidad que indican cuáles son las cualidades deseables para determinar la

calidad de un producto de software y aunque de alguna forma se definen criterios para medir dicha calidad, es necesario establecer métodos de evaluación cuantitativa que permitan obtener resultados de manera más objetiva, integrando tanto los elementos técnicos a evaluar como el involucramiento del usuario final en la determinación de la calidad del software.

Considerando también, que la liberación de un producto de software sin realizar las pruebas adecuadas, lleva implícito un alto grado de vulnerabilidad. Entonces, es importante contemplar que el método de evaluación del producto de software se integre de manera formal dentro del proyecto de desarrollo, buscando que el producto que se entregue al usuario final cuente con un nivel de calidad óptimo, resultado de una adecuada evaluación y corrección.

Zayaraz et al. (2005) proponen una técnica de aseguramiento de calidad en base a la técnica de puntos de función COSMIC FFP y del estándar de calidad ISO 9126 para implementar la calidad a nivel de arquitectura del software. La técnica propuesta mide atributos como el acoplamiento, mantenibilidad, cohesión, complejidad, confiabilidad y usabilidad.

Amasaki et al. (2005) hablan del uso de Software Reliability Growth Model (SRGM) aplicado en cada actividad de prueba para determinar el tiempo de término de la prueba y explican como su aplicación no siempre da resultados correctos, lo cual puede provocar productos con una calidad pobre. Para solucionar esto, proponen el uso de tendencias en base a la relación de éstas con los datos de series de tiempo y la calidad del software. Para lo anterior, utilizan regresión logística y muestran estadísticamente que dos revisiones métricas en las fases de diseño y codificación pueden determinar la tendencia.

Por otro lado, Ramani (2007) se enfoca a la medición de la satisfacción del cliente en base a la calidad de uso de un software, haciendo énfasis en que además de que los productos deben estar creados con una técnica excelente, deben ser fáciles de usar. Hace referencia también a los conceptos de calidad establecidos en el estándar ISO/IEC 8402 y menciona las siguientes características de calidad: funcionabilidad, fiabilidad, eficiencia, mantenibilidad y portabilidad. De la misma manera, él referencia el modelo de calidad ISO 9126 que describe las tres partes de un modelo de software de calidad: 1)

calidad interna (a nivel código), 2) calidad externa (propiedades del código cuando se ejecuta como por ejemplo tiempo de respuesta) y 3) calidad en el uso (como por ejemplo la productividad). Finalmente, enfatiza en el uso de métricas de calidad.

Con el análisis de las investigaciones mencionadas, se encuentra que el mayor énfasis se hace en la calidad durante el proceso de desarrollo, centrándose principalmente en la arquitectura del software y únicamente Ramani (2007) se enfoca más hacia el concepto de la calidad orientada al uso y a la satisfacción del cliente, que es justamente la línea que se sigue en la presente investigación.

3.2 Pregunta de Investigación

Dado todo lo anterior, se identifica la gran necesidad de investigar el efecto que la participación de los usuarios tiene en la calidad final de un producto de software. En consecuencia se plantea la siguiente pregunta:

¿Qué resultados se obtienen en la calidad final de un producto de software al aplicarle un método de evaluación por parte del usuario final en base a factores de calidad determinados en los modelos de calidad ISO-9126 y McCall?

3.3 Objetivos de la Investigación

La presente investigación plantea una serie de objetivos, los cuales se describen a continuación.

3.3.1 Objetivo General

Diseñar un método para la evaluación de un producto de software donde los usuarios finales participen, en base a la definición de factores de calidad de los modelos existentes, que permita mediante su aplicación de manera iterada y con la utilización de técnicas estadísticas, la obtención de un producto con calidad hasta lograr el nivel ideal para su liberación a los usuarios finales.

3.3.2 Objetivos Específicos

- Determinar los principales requerimientos de calidad orientada al producto de una aplicación de software en términos de factores de calidad identificados por el usuario.
- Lograr la integración adecuada de los requerimientos generales con los requerimientos de calidad deseables en la aplicación de software.
- Establecer los criterios de evaluación de los factores de calidad que se aplicaran en cada iteración planeada.
- Establecer los criterios en base a los cuales se considerará que la aplicación está lista para su liberación al usuario final.
- Evaluar la calidad por parte de los usuarios de forma tal que se identifiquen desviaciones en el producto que deben ser corregidas por el equipo de desarrollo.

3.4 Hipótesis

Debido al tipo de investigación, es necesario el planteamiento de una serie de hipótesis, las cuales permiten identificar si la participación de los usuarios finales en la evaluación de la calidad del software a través de los modelos de calidad tienen o no efecto en el producto final. La aplicación de un método iterativo de evaluación de la calidad del software orientada al producto, mediante técnicas estadísticas y en base a la identificación de factores de calidad clave desde el punto de vista del usuario, permite lograr aplicaciones con un alto nivel de calidad, reflejado en una mayor satisfacción del cliente. Las hipótesis de la presente investigación son las siguientes:

H1. La evaluación de la Funcionalidad de un producto de software por parte del usuario final mejora significativamente su calidad.

H1a. La evaluación de la Conformidad de un producto de software por parte del usuario final mejora significativamente su calidad

H1b. La evaluación de la Exactitud de un producto de software por parte del usuario final, mejora significativamente su calidad

H1c. La evaluación de la Seguridad de un producto de software por parte del usuario final mejora significativamente su calidad

H2. La evaluación de la Confiabilidad de un producto de software por parte del usuario final mejora significativamente su calidad

H2a. La evaluación de la Madurez de un producto de software por parte del usuario final mejora significativamente su calidad

H2b. La evaluación de la Tolerancia a Fallas de un producto de software por parte del usuario final mejora significativamente su calidad

H2c. La evaluación de la Recuperabilidad de un producto de software por parte del usuario final mejora significativamente su calidad

H3. La evaluación de la Facilidad de Uso de un producto de software por parte del usuario final mejora significativamente su calidad.

H3a. La evaluación de la Facilidad de Compresión (Comprensibilidad) de un producto de software por parte del usuario final mejora significativamente su calidad

H3b. La evaluación de la Facilidad de Aprendizaje de un producto de software por parte del usuario final mejora significativamente su calidad

H3c. La evaluación de la Operabilidad de un producto de software por parte del usuario final mejora significativamente su calidad

H4. La evaluación de la Eficiencia de un producto de Software por parte del usuario final mejora significativamente su calidad.

H4a. La evaluación del Comportamiento del tiempo de respuesta de un producto de software por parte del usuario final mejora significativamente su calidad

H5. La evaluación de la Portabilidad de un producto de software mejora significativamente su calidad.

H5a. La evaluación de la Adaptabilidad de un producto de software por parte del usuario final mejora significativamente su calidad



4. DESARROLLO DE LA INVESTIGACIÓN

El enfoque general de la investigación es aplicada, ya que en este tipo de investigación el objetivo es aplicar los conocimientos científicos en la solución de problemas de producción de bienes y servicios (Ortiz y García, 2003). En este caso, el servicio o producto es lograr aplicaciones de software con un alto nivel de calidad.

El tipo de Investigación es experimental ya que se pretende en base a la definición de factores definidos como deseables en la calidad del producto de software, determinar un método de evaluación de una aplicación terminada para que medio de pruebas que realizaran grupos de usuarios muestra, determinar su nivel de calidad y en base a este resultado realizar los ajustes necesarios a la aplicación y volver a evaluar hasta lograr una aplicación final con un nivel óptimo de calidad.

La investigación se considera también explicativa, ya que su interés principal es explicar por qué ocurre un fenómeno y en qué condiciones se da y por qué dos o más variables están relacionadas (Hernández, Fernández et al. 1998). En este casos se analizan si con una evaluación adecuada de determinados factores de calidad en una aplicación de software, se puede lograr una aplicación final que garantice una mayor calidad. Traduciendo todo esto en una mayor aceptación y satisfacción del software resultante por parte del cliente.

4.1 Materiales y Métodos

Para la presente investigación se desarrolló un método que consistió en establecer en base a los modelos y estándares de calidad analizados (McCall e ISO 9126), qué factores de calidad de un producto de software deben evaluarse desde el punto de vista del usuario final, así como también cómo deben medirse y en qué términos. Una vez obtenidos los criterios definidos y diseñado el instrumento, se procedió a realizar la evaluación de una aplicación de software en un determinado número de iteraciones mediante grupos de usuarios específicos, con el objetivo de lograr en cada iteración, una evaluación adecuada del producto, que permitiera en lo subsecuente mejorar su calidad hasta lograr un nivel óptimo de la misma para su liberación final.

Para la evaluación de los factores a analizar se utilizaron técnicas de estadística descriptiva, las cuales permitieron obtener un resultado medible y comparable de tal forma que se aprecie de manera concreta la calidad asignada por el usuario al producto de software evaluado.

En la Figura 3 se muestra el proceso de desarrollo de software base para la realización del método desarrollado, dentro del cual el énfasis se realiza en el paso *Pruebas de calidad (usuarios)* con el objetivo de que al ser parte integral del proceso de desarrollo, la evaluación de los usuarios en forma iterada permitirá llegar a la liberación de un producto final óptimo, garantizando de forma inherente la satisfacción del cliente.

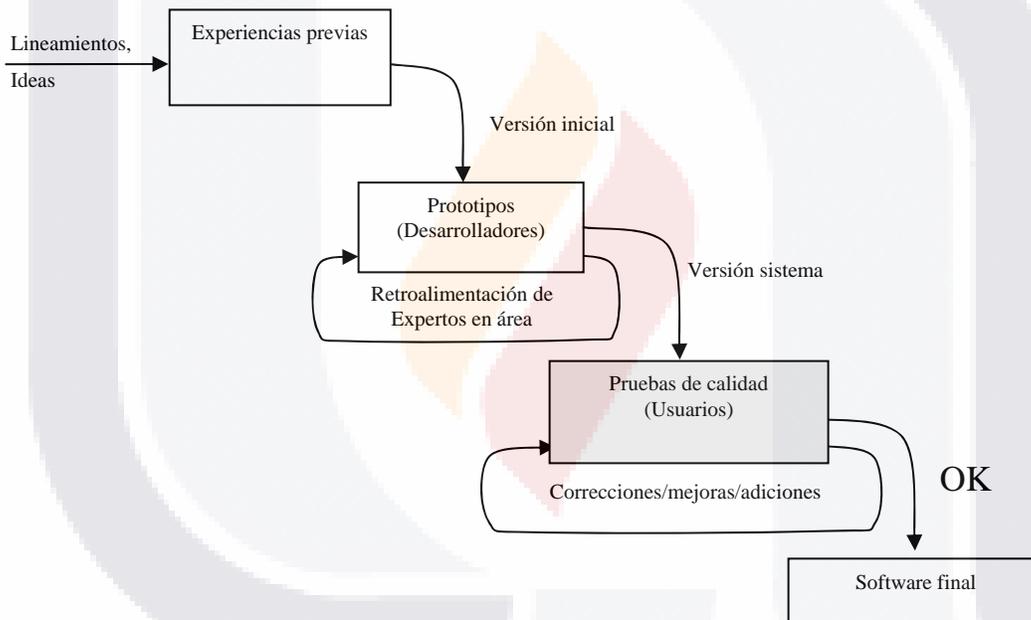


Figura 3. Modelo de desarrollo base del método de evaluación de calidad del producto (Gómez Reynoso y Brizuela Sandoval, 2008)

Para la presente investigación se analizaron los modelos de calidad de McCall e ISO 9126, para lo que se realizó el cruce e integración de cada uno de sus atributos, identificando cuáles de ellos podían evaluarse desde el punto de vista del usuario final (Ver Figura 4 y Tabla 3).

Después de tener identificados los factores (ver Tabla 4) y sus hipótesis relacionadas, se procedió a la formulación de las preguntas necesarias para evaluar cada factor.

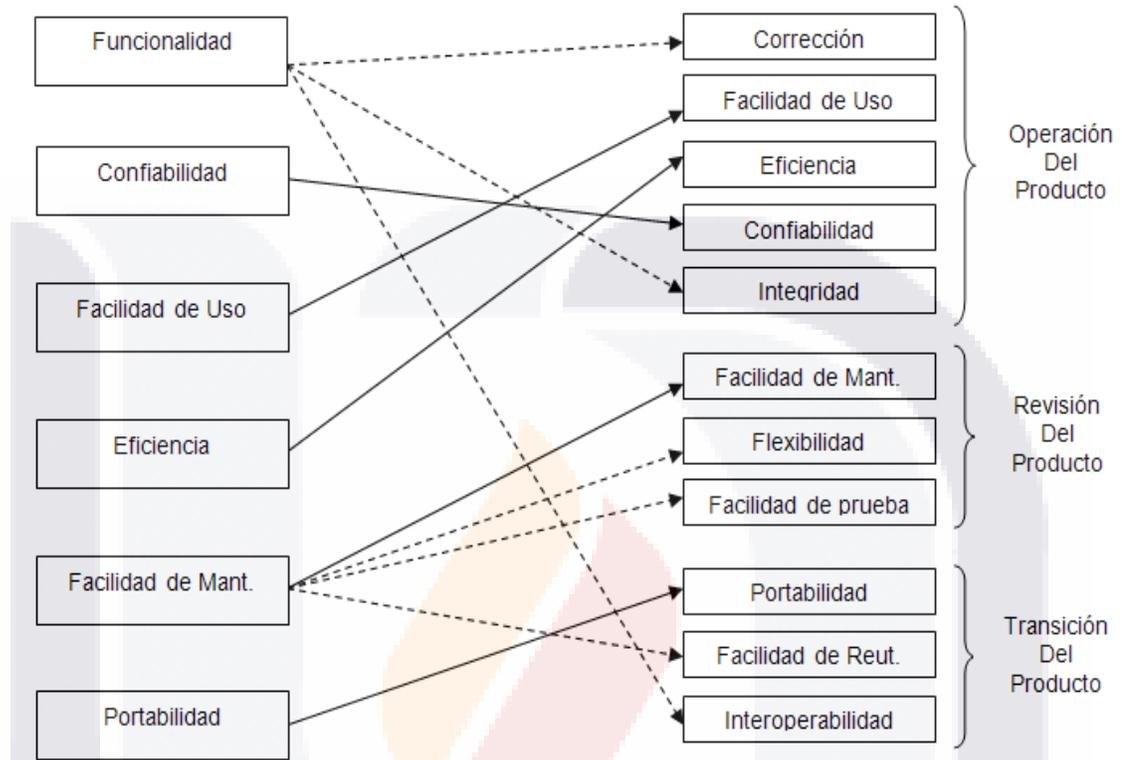


Figura 4. Equivalencias McCall - ISO 9126

Subatributos ISO 9126	McCall
Funcionalidad.Conformidad	OperaciónDelProducto.Corrección
Funcionalidad.Exactitud	OperaciónDelProducto.Corrección
Funcionalidad.Seguridad	OperaciónDelProducto.Integridad
Funcionalidad.Cumplimiento	
Funcionalidad.Interoperabilidad	TransicionDelProducto.Interoperabilidad
Confiabilidad.Madurez	OperaciónDelProducto.Confiabilidad
Confiabilidad.ToleranciaAFallos	
Confiabilidad.Recuperabilidad	
FacilidadDeUso.Comprendibilidad	OperaciónDelProducto.FacilidadDeUso
FacilidadDeUso.FacilidadDeAprendizaje	
FacilidadDeUso.Operabilidad	
Eficiencia.ComportamientoEnElTiempo	OperaciónDelProducto.Eficiencia
Eficiencia.ComportamientoEnLosRecursos	
FacilidadDeMantenimiento.FacilidadDe Análisis	RevisionDelProducto.FacilidadDeMantenimiento
FacilidadDeMantenimiento.FacilidadDe cambio	RevisionDelProducto.Flexibilidad
FacilidadDeMantenimiento.Estabilidad	RevisionDelProducto.FacilidadDeMantenimiento
FacilidadDeMantenimiento.FacilidadDePrueba	RevisionDelProducto.FacilidadDePrueba
Portabilidad.Adaptabilidad	TransicionDelProducto.Portabilidad
Portabilidad.CapacidadDeInstalacion	TransicionDelProducto.Portabilidad
Portabilidad.Conformidad	TransicionDelProducto.Portabilidad
Portabilidad.CapacidadDeReemplazamiento	

Tabla 3. Equivalencias McCall – ISO 9126

Característica de Calidad	Factor	Descripción
Funcionalidad	Conformidad	El producto de software tiene capacidad de proporcionar funciones adecuadas para tareas específicas
	Exactitud	El producto de software proporciona resultados correctos, precisos y satisface los objetivos de cliente
	Seguridad	El producto de software cuenta con mecanismos de seguridad y control de accesos no autorizado a los datos
Confiabilidad	Madurez	El producto de software tiene la capacidad de evitar errores
	Tolerancia a Fallas	El producto de software está preparado para mantener un desempeño adecuado en caso de fallas o errores
	Recuperabilidad	El producto de software que considera las acciones necesarias para recuperarse en caso de fallas de manera fácil y rápida
Facilidad de Uso	Comprendibilidad	El producto de software es fácil de entender en su conveniencia y en cómo puede ser utilizado
	Facilidad de Aprendizaje	El producto de software permite al usuario aprender su utilización de manera fácil y rápida
	Operabilidad	El producto de software facilita al usuario su operación y control
Eficiencia	Tiempo de Respuesta	El producto de software cuenta con tiempos adecuados de respuesta al realizar sus diferentes funciones
Portabilidad	Adaptabilidad	El producto de software puede adaptarse a diferentes entornos sin la necesidad de aplicar acciones específicas diferentes a las establecida

Tabla 4. Factores para evaluar la calidad de una aplicación de software desde el punto de vista del usuario

4.2 Estudio Piloto

En base a la revisión de la literatura y las teorías base sobre calidad del software, modelos y estándares de calidad analizados (McCall e ISO 9126), se identificaron de manera preliminar los elementos de calidad de un producto de software que pueden ser percibidos por los usuarios.

A partir de los elementos identificados como factores en la evaluación de la calidad del producto, se crearon las preguntas necesarias para su valoración, así como la escala de respuestas posible para sus respuestas.

En base a lo anterior, se integró un cuestionario inicial, para su aplicación como parte del estudio piloto, a 14 alumnos de la asignatura de Ingeniería de Software en el 9º semestre de la carrera de Ingeniería en Sistemas Computacionales de la Universidad Autónoma de Aguascalientes. Para tal efecto, los estudiantes evaluaron con el instrumento creado, la calidad de aplicaciones de software que ellos mismos elaboraron como parte de su asignatura.

La aplicación del estudio piloto permitió la retroalimentación de los alumnos involucrados, en cuanto a la utilidad del instrumento creado para la evaluación de la calidad de sus aplicaciones de software. De esta forma, se pudieron identificar problemas potenciales en el diseño, mismos que fueron corregidos, dando como resultado el instrumento final de evaluación de calidad de los productos de software.

Al analizar los resultados del estudio piloto y corregir los aspectos identificados por los participantes en la redacción así como la evaluación del mismo se generó el cuestionario final, el cual sirvió como instrumento de medición para la presente de investigación.

El instrumento final quedó conformado por 25 preguntas integradas en 11 factores. Para contestar cada pregunta, se utilizó una escala de 1 a 7 de Likert. Donde los valores van desde *Totalmente de acuerdo* hasta *Totalmente en desacuerdo*. Una vez obtenido el instrumento final se procedió a su utilización para la evaluación de la aplicación seleccionada como caso de estudio.

4.3 Estudio Final

4.3.1 Población Objetivo

El presente estudio identifica como Usuarios a aquellos que hacen uso de las aplicaciones de software como parte de su actividad diaria o que estén en proceso de su implementación.

4.3.2 Selección de la Muestra

Para la recolección de datos a través del instrumento diseñado, se utilizó como caso de estudio la aplicación de “Petición al Área de Sistemas” perteneciente al Sistema Integral de Información y Modernización Administrativa (SIIMA) de la Universidad Autónoma de Aguascalientes. El SIIMA integra el manejo de las 3 principales áreas funcionales de la institución: Control Escolar, Finanzas y Recursos Humanos y fue desarrollado por el Área de Sistemas interna de la institución.

De manera general, el funcionamiento de la aplicación seleccionada es el siguiente:

- Si una área de la institución requiere alguna modificación o mejora de SIIMA, debe solicitarla al área de Sistemas por medio de la aplicación de Petición al Área de Sistemas incluida dentro de SIIMA, dicha aplicación lleva el seguimiento de una petición desde que el usuario la solicita hasta que ésta es resuelta, pasando por todos los pasos necesarios para su correcta atención y seguimiento.

La aplicación de Petición está integrada por 5 apartados principales ordenados en pestañas, los cuales estarán visibles en base al tipo de usuario que esté utilizando la aplicación. A continuación se describe brevemente cada una de ellas.

- Petición. En esta pestaña se tiene el listado de peticiones de acuerdo a los filtros seleccionados y es la pestaña principal.

- Detalles/Captura. Por medio de esta pestaña se realiza la manipulación general de peticiones: insertar, modificar, etc. así como los comentarios relacionados y en su caso la calificación otorgada a su atención por medio del usuario
- Objetivos. En esta pestaña se realizan las operaciones relacionadas al manejo de objetivos de una petición, tales como insertar (Asignación), modificar, Seguimiento etc.
- Reporte Semanal. En esta pestaña se generan los reportes de peticiones y objetivos asignados a un usuario Analista Desarrollador para su atención.
- Consulta de Objetivos. En esta pestaña se realiza una consulta global de los objetivos registrados en las diferentes peticiones.

Esta aplicación fue elegida como caso de estudio ya que es utilizada por usuarios de diferentes perfiles: usuarios administrativos y directivos que solicitan peticiones, administradores de módulo para su atención y canalización, así como finalmente los desarrolladores para su realización y retroalimentación, por lo que se consideró importante la diversidad de respuestas que se podría obtener en base a los diferentes perfiles involucrados. Además de que esta aplicación es de uso cotidiano para los tipos de usuarios mencionados.

De manera general, la evaluación se aplicó a todos los usuarios del Área de Sistemas (Jefe de Área, Administradores de Módulo, Desarrolladores Analistas y Asistente Técnico) así como a usuarios de áreas administrativas tales como Redes y Telecomunicaciones, Control Escolar, Recursos Humanos, etc. completando un total de 29 personas.

4.3.3 Recopilación de la Información.

La aplicación del instrumento de evaluación se realizó por invitación enviando el cuestionario correspondiente vía correo electrónico a los usuarios seleccionados.

La primera iteración se realizó en base a la versión actual de la aplicación que se encuentra de manera oficial dentro del menú de la aplicación de SIIMA y se obtuvieron un total de 29 cuestionarios contestados.

La segunda iteración se realizó en base a la versión corregida que se hizo de la versión original en base a los resultados de la primera iteración, pero esta versión no fue publicada dentro del menú oficial de SIIMA debido a que no era una aplicación 100% probada para su liberación, de acuerdo a la metodología actual del Área de Sistemas. De esta iteración se obtuvieron 26 cuestionarios contestados teniendo una mortandad de 3 usuarios con respecto a la primera iteración.

A continuación se detalla la descripción de los datos demográficos de los usuarios que intervinieron en ambas iteraciones.

4.3.3.1 Primera Iteración

De los 29 usuarios que intervinieron en la primera iteración, se contó con al menos un usuario de cada rango de edad establecido, como puede verse en la Tabla 5. La mayor concentración se dio en los rangos de 21 a 25 y 36 a 40 años, lo que permitió obtener una retroalimentación variada, ya que se pudieron identificar 2 escenarios principales: a) usuarios con poco tiempo dentro de la institución y por lo tanto menor tiempo de utilización de la aplicación y b) usuarios con al menos 10 años de diferencia que en su mayoría tienen ya varios años laborando en la institución y que están más acostumbrados a la utilización de la aplicación evaluada y en general a diferentes aplicaciones de software.

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje Acumulado
Válidos 21 a 25 años	9	31.0	31.0	31.0
26 a 30 años	1	3.4	3.4	34.5
31 a 35 años	4	13.8	13.8	48.3
36 a 40 años	9	31.0	31.0	79.3
41 años o mas	6	20.7	20.7	100.0
Total	29	100.0	100.0	

Tabla 5. Tabla de Frecuencia de Edad - Primera iteración

Del total de usuarios encuestados, el mayor porcentaje fueron del género *masculino* (ver Tabla 6) con un 69%, esto por la naturaleza de las áreas de trabajo que fueron seleccionadas para la aplicación del instrumento. Cabe mencionar que en relación a la cantidad de usuarios totales de la aplicación evaluada, el porcentaje debe tender a ser más equilibrado.

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje Acumulado
Válidos Femenino	9	31.0	31.0	31.0
Masculino	20	69.0	69.0	100.0
Total	29	100.0	100.0	

Tabla 6. Tabla de Frecuencia de Género - Primera iteración

Respecto al aspecto del nivel de estudios de los usuarios involucrados, se puede observar en la Tabla 7, que la mayoría cuenta con un nivel licenciatura, y que de manera general, la mayor proporción de la población cuenta con un nivel licenciatura o superior. En consecuencia, esto puede provocar que los participantes tiendan a ser más críticos de la aplicación. Entonces, pueden demandar una mejoría notable de la aplicación inicial, lo cual es muy bueno para la presente investigación.

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje Acumulado
Válidos Secundaria-Bachillerato	5	17.2	17.2	17.2
Licenciatura	15	51.7	51.7	69.0
Especialidad	2	6.9	6.9	75.9
Maestría	7	24.1	24.1	100.0
Total	29	100.0	100.0	

Tabla 7. Tabla de Frecuencia de Nivel de Estudios - Primera iteración

En cuanto al Tipo de Puesto de los usuarios seleccionados (ver Tabla 8), la mayoría son de confianza, contando también con un Administrativo Directivo.

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje Acumulado
Válidos Administrativo de confianza	17	58.6	58.6	58.6
Administrativo por honorarios	11	37.9	37.9	96.6
Administrativo directivo	1	3.4	3.4	100.0
Total	29	100.0	100.0	

Tabla 8. Tabla de Frecuencia de Tipo de Puesto - Primera iteración

Finalmente, en relación al área de estudios de los usuarios involucrados, como se puede ver en la Tabla 9, la mayor parte se concentró en el área de *Sistemas de Información*, esto debido a que se incluyó en la muestra a todos los usuarios del Área de Sistemas de la institución. Este tipo de usuarios por el hecho de ser expertos omitieron observaciones cualitativas más específicas y más sugerencias de mejora. En general el contar con usuarios de diferentes áreas de estudio ayudó a tener mejores resultados ya que fue una de las variables demográficas que más impacto tuvieron en la evaluación de la aplicación.

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje Acumulado
Válidos Sistemas de Información	22	75.9	75.9	75.9
Administración	5	17.2	17.2	93.1
Otros	2	6.9	6.9	100.0
Total	29	100.0	100.0	

Tabla 9. Tabla de Frecuencia de Área de Estudios - Primera iteración

4.3.3.2 Segunda Iteración

La edad de los usuarios de la segunda iteración (Ver Tabla 10) siguió la misma tendencia porque precisamente 2 de los usuarios faltantes eran pertenecientes a los rangos de edad más predominantes.

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje Acumulado
Válidos 21 a 25 años	8	30.8	30.8	30.8
26 a 30 años	1	3.8	3.8	34.6
31 a 35 años	2	7.7	7.7	42.3
36 a 40 años	8	30.8	30.8	73.1
41 años o mas	7	26.9	26.9	100.0
Total	26	100.0	100.0	

Tabla 10. Tabla de Frecuencia de Edad - Segunda iteración

En cuanto al género de los usuarios (Ver Tabla 11) aunque siguió predominando el género *Masculino*, la proporción disminuyó porque los casos faltantes fueron precisamente hombres.

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje Acumulado
Válidos Femenino	9	34.6	34.6	34.6
Masculino	17	65.4	65.4	100.0
Total	26	100.0	100.0	

Tabla 11. Tabla de Frecuencia de Género - Segunda iteración

En relación al Nivel de Estudios de los usuarios, los casos faltantes fueron 2 de *Especialidad* y 1 de *Maestría*, por lo que la proporción principal de *Licenciatura* se mantuvo (Ver Tabla 12).

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje Acumulado
Válidos Secundaria-Bachillerato	5	19.2	19.2	19.2
Licenciatura	15	57.7	57.7	76.9
Maestría	6	23.1	23.1	100.0
Total	26	100.0	100.0	

Tabla 12. Tabla de Frecuencia de Nivel de Estudios - Segunda iteración

En cuanto al Tipo de Puesto de los usuarios, uno de los casos faltantes fue del tipo *Directivo* por lo que esta categoría se quedó vacía. Los otros 2 casos fueron de 1 de cada uno de los tipos restantes: *Administrativos de confianza* y *Administrativos por honorarios* (Ver Tabla 13).

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje Acumulado
Válidos Administrativo de confianza	16	61.5	61.5	61.5
Administrativo por honorarios	10	38.5	38.5	100.0
Total	26	100.0	100.0	

Tabla 13. Tabla de Frecuencia de Tipo de Puesto - Segunda iteración

Finalmente en el Área de Estudios de los usuarios de esta iteración (Ver Tabla 14), los usuarios faltantes fueron 1 del área de Sistemas de Información y 1 tanto de *Administración* como de *Otros*, por lo que la proporción de usuarios expertos no cambió.

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje Acumulado
Válidos Sistemas de Información	21	80.8	80.8	80.8
Administración	4	15.4	15.4	96.2
Otros	1	3.8	3.8	100.0
Total	26	100.0	100.0	

Tabla 14. Tabla de Frecuencia de Área de Estudios - Segunda iteración

5. ANÁLISIS DE DATOS

El presente capítulo se muestra el análisis de datos de las iteraciones realizadas, con lo cual se pueden identificar los efectos que tiene la participación de los usuarios en la evaluación de la calidad de las aplicaciones de software. Las siguientes secciones describen los hallazgos de la presente investigación.

5.1 Primera Iteración

A continuación se describen los resultados obtenidos en la primera iteración, respecto a la calidad inicial de la aplicación de software evaluada.

Con respecto al factor *Conformidad*, que indica sí la aplicación cuenta con funciones adecuadas para tareas específicas, se encontró que la calidad promedio fue de 2.2 (ver Tabla 15), lo cual indica que los usuarios opinaron que la versión evaluada de la aplicación tenía algo de calidad en este aspecto, pero que aún le faltaba mucho por mejorar.

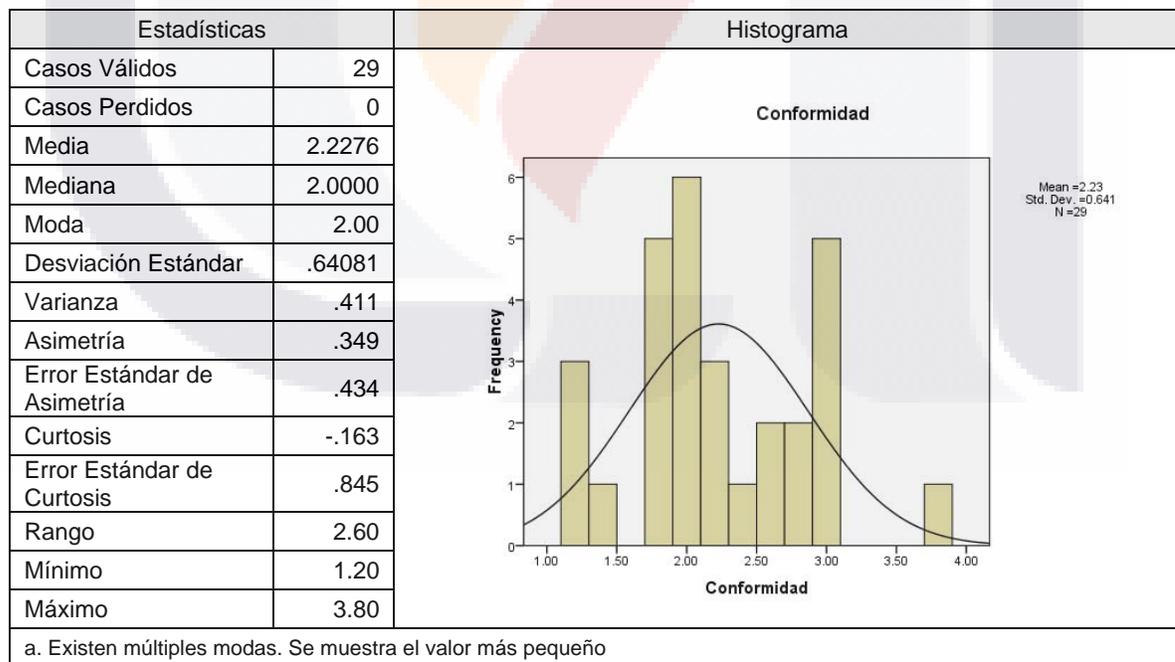


Tabla 15. Estadísticas e Histograma del Factor Conformidad - Primera iteración

En relación al factor *Exactitud* (ver Tabla 16), se encontró que la calidad promedio fue de 2.2, lo cual indica que los usuarios opinaron que la versión evaluada de la aplicación tenía algo de calidad pero no la suficiente. Esto en cuanto a qué tanto la aplicación cumplía con sus objetivos y les proporcionaba resultados exactos y precisos.

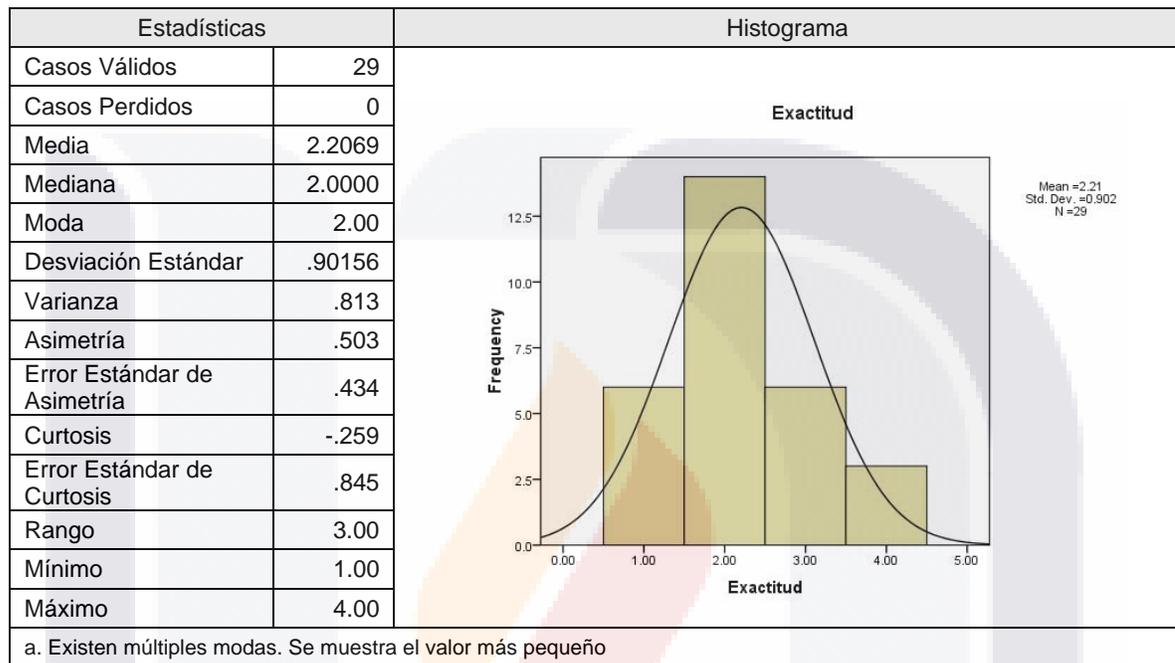


Tabla 16. Estadísticas e Histograma del Factor Exactitud - Primera iteración

En el factor *Seguridad* se puede observar (ver Tabla 17) que la calidad promedio fue de 1.87, lo cual indica que los usuarios opinaron que la versión de la aplicación evaluada, tenía una buena calidad en cuanto a sus mecanismos de seguridad y control de acceso, pero se consideró recomendable revisar qué aspectos estaban afectando, para que la percepción se aproximara a una calidad totalmente satisfactoria.

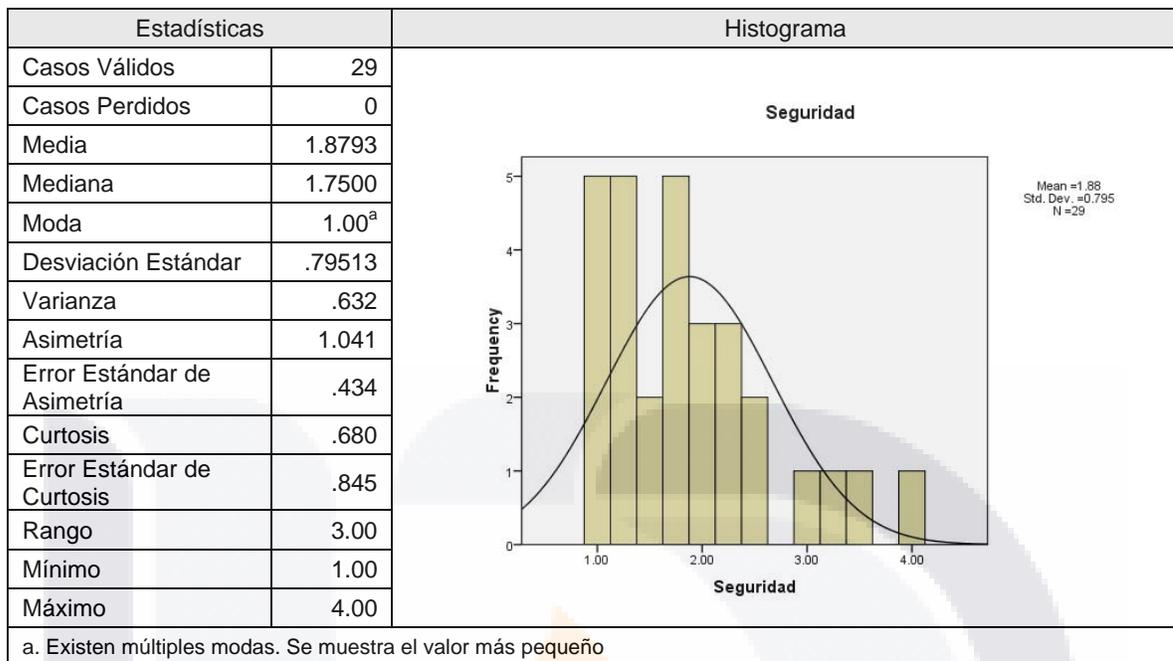


Tabla 17. Estadísticas e Histograma del Factor Seguridad- Primera iteración

Respecto al factor *Madurez* (ver Tabla 18) la calidad promedio fue de 3.17, lo cual indica que los usuarios tuvieron una opinión de una calidad mediocre de la versión evaluada de la aplicación, en cuanto a la capacidad de la aplicación de evitar errores. Esto generó un punto de análisis importante de mejora como parte del Control de Errores de la aplicación.

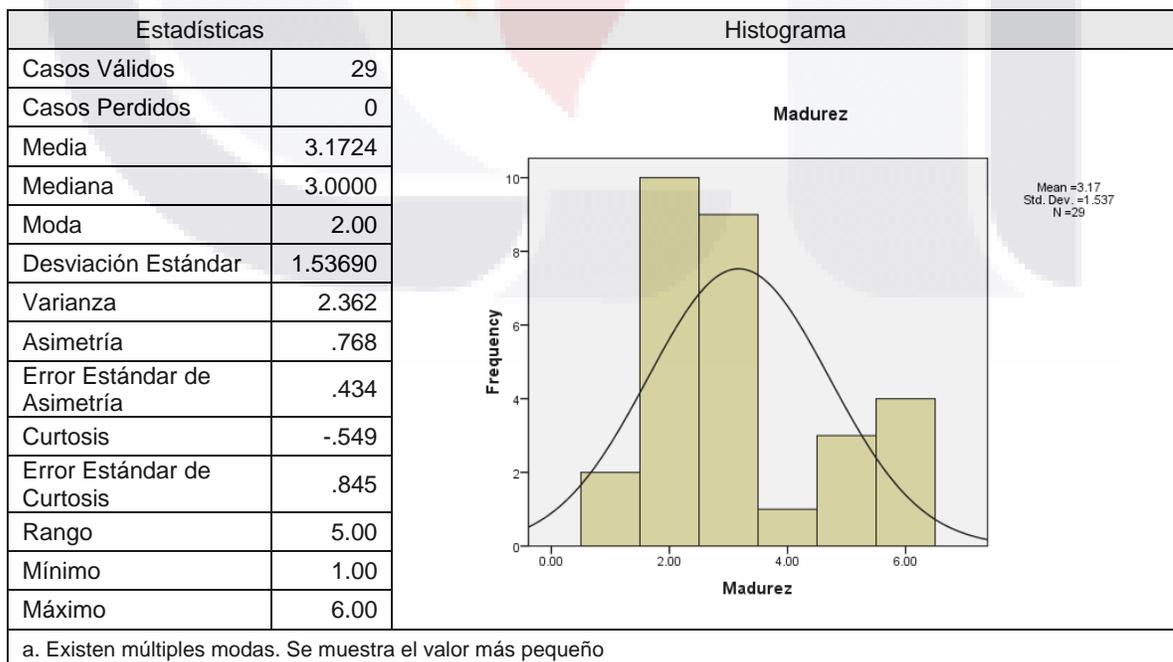


Tabla 18. Estadísticas e Histograma del Factor Madurez - Primera Iteración

En el factor *Tolerancia a Fallas* se encontró que la calidad promedio fue de 2.65 (ver Tabla 19), lo cual indica que los usuarios opinaron que la versión evaluada de la aplicación tenía una buena calidad, en cuanto a que ésta mantenía un desempeño adecuado en caso de fallas o errores. Pero se encontró la necesidad de mejorar este aspecto como parte del Control de Errores general de la aplicación.

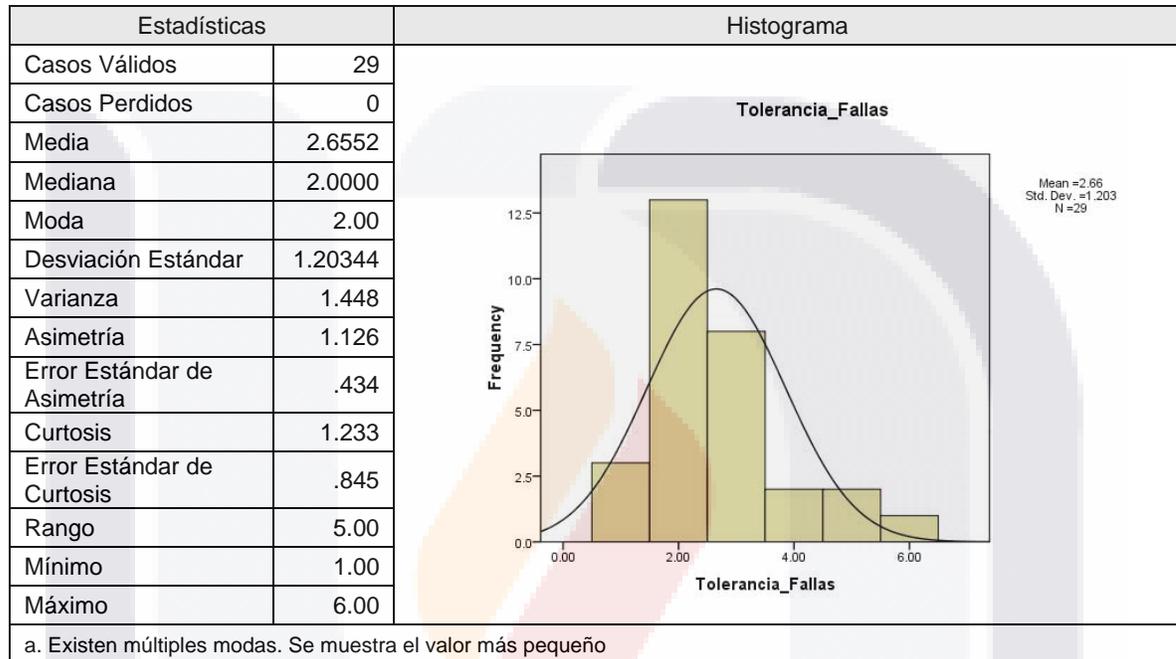


Tabla 19. Estadísticas e Histograma del Factor Tolerancia a Fallas - Primera Iteración

En el factor *Recuperabilidad*, la calidad promedio fue de 3.27 (ver Tabla 20), lo cual indica que los usuarios - al igual que en el factor anterior - opinaron que esa versión de la aplicación evaluada, tenía una calidad baja en cuanto a estar preparada para recuperarse en caso de fallas de forma fácil y rápida. Por lo anterior, se consideró necesario realizar mejoras en este aspecto como parte del Control de Errores general de la aplicación.

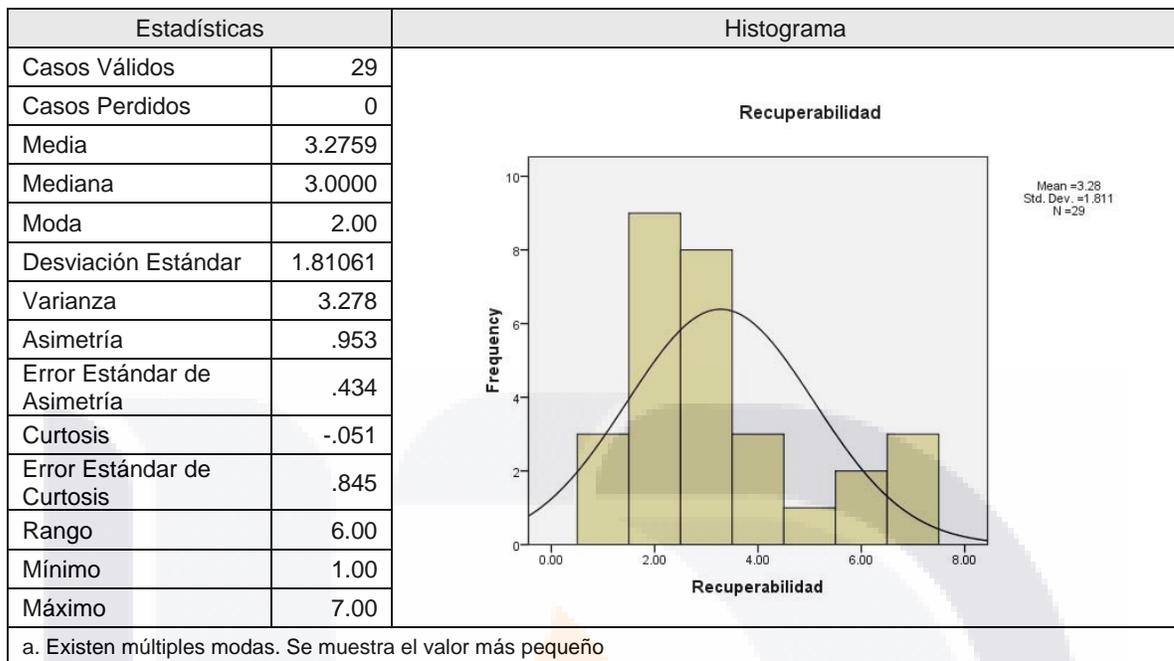


Tabla 20. Estadísticas e Histograma del Factor Recuperabilidad - Primera Iteración

En cuanto al factor *Comprensibilidad*, se encontró que la calidad promedio fue de 2.22 (ver Tabla 21), lo que refleja que los usuarios opinaron que la calidad de la aplicación en la versión evaluada era buena, en relación a que la consideraban fácil de entender. Aquí se consideró importante lograr una mejora para maximizar la calidad.

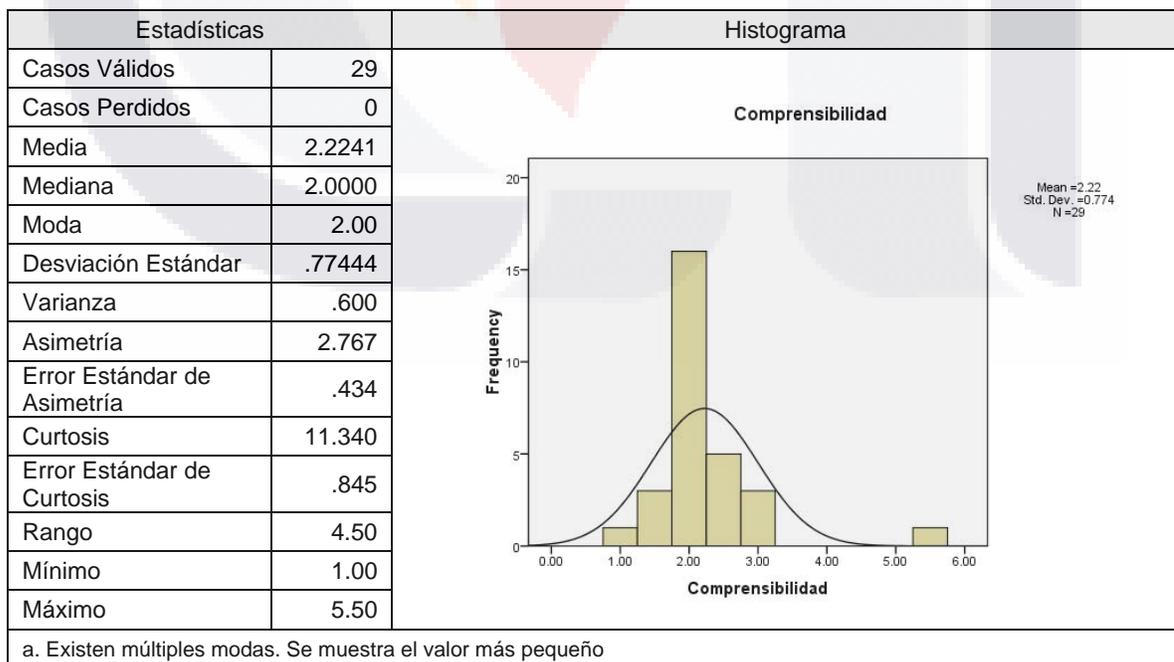


Tabla 21. Estadísticas e Histograma del Factor Comprensibilidad - Primera Iteración

Respecto al factor *Facilidad de Aprendizaje*, la calidad promedio fue de 3.17 (ver Tabla 22), lo cual indica que los usuarios opinaron que la versión evaluada de la aplicación tenía una calidad mediocre, en cuanto a su facilidad para aprender a utilizarla. Esto generó un punto de análisis importante de mejora por ser de los aspectos que se consideran que más pueden impactar en la percepción de la calidad por parte del usuario.

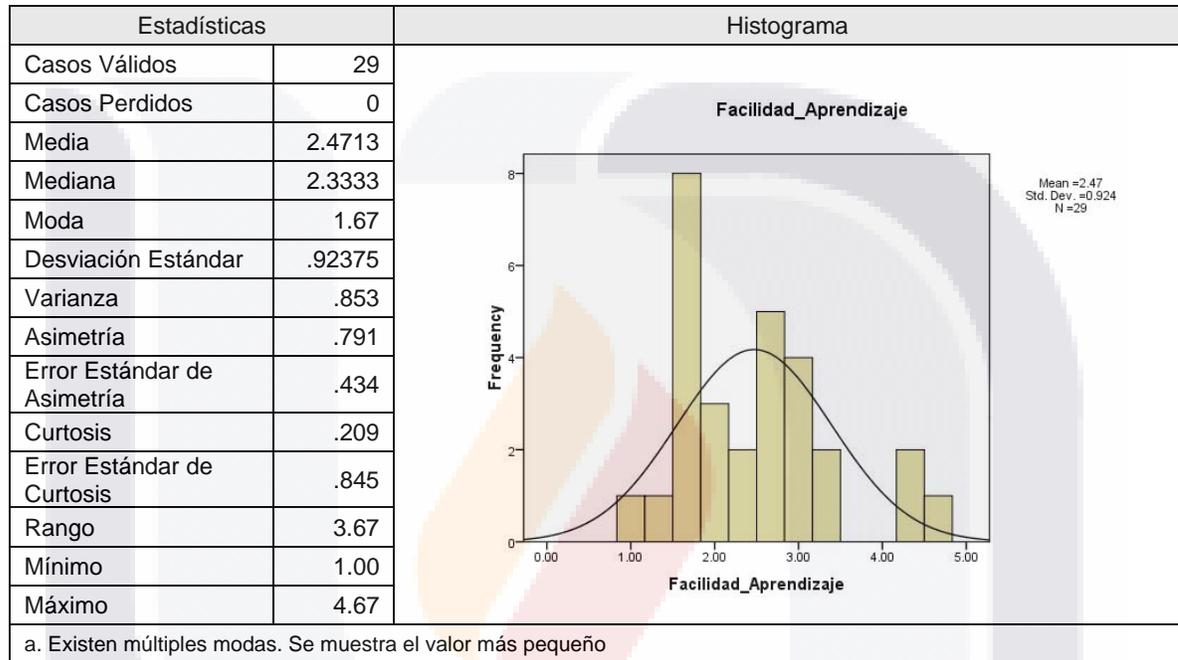


Tabla 22. Estadísticas e Histograma del Factor Facilidad de Aprendizaje - Primera Iteración

En el factor de *Operabilidad*, la calidad promedio fue de 2.60 (ver Tabla 23), lo cual indica que los usuarios tuvieron una buena opinión respecto a la calidad de la aplicación en la versión evaluada, en relación a si ésta les facilitaba su operación y control. Este aspecto se consideró importante para mejorar, porque también es de los más tangibles por parte del usuario.

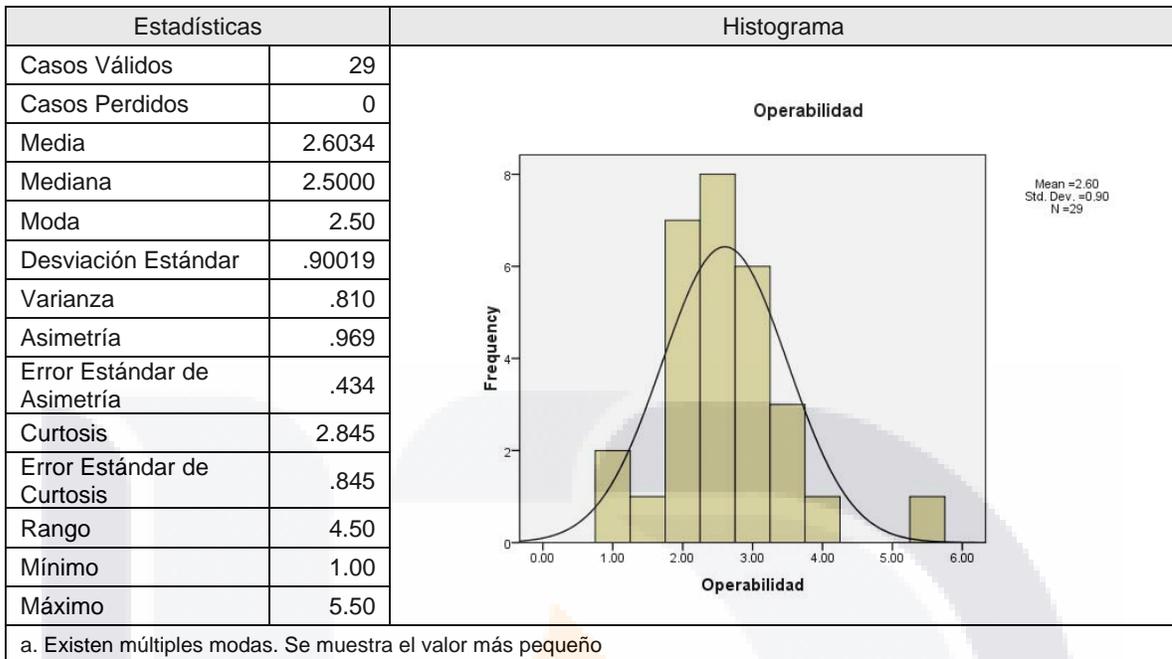


Tabla 23. Estadísticas e Histograma del Factor Operabilidad - Primera Iteración

En el factor *Tiempo de Respuesta*, la calidad promedio fue de 2.65 (ver Tabla 24), lo cual indica que los usuarios tuvieron también una buena opinión respecto a la calidad de la de la aplicación en relación a sus tiempos de respuesta. Sin embargo, este factor fue de los que más se recibieron comentarios ya que los usuarios deseaban una mayor velocidad.

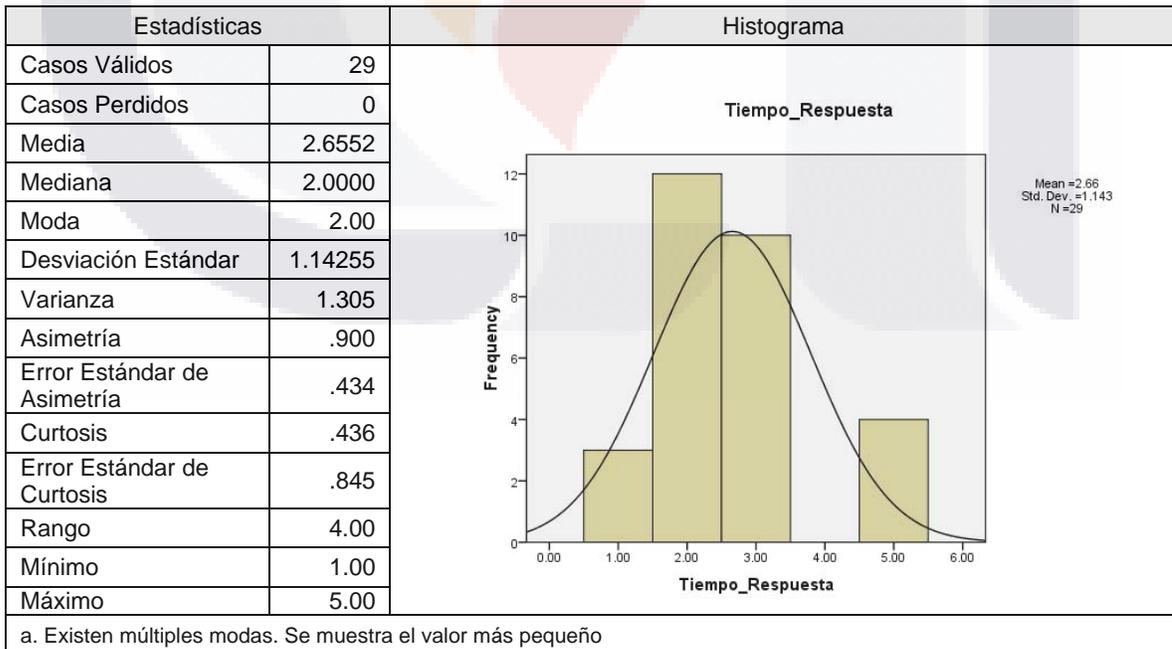


Tabla 24. Estadísticas e Histograma del Factor Tiempo de Respuesta - Primera Iteración

Finalmente, en cuanto al factor *Adaptabilidad* (ver Tabla 25) la calidad promedio fue de 3.27, lo cual indica que los usuarios opinaron que la versión evaluada de la aplicación tenía calidad mediocre, esto en relación a que la aplicación no es tan adaptable a diferentes entornos en cuestión de su interfaz, por lo que es un aspecto que se debe atacar.

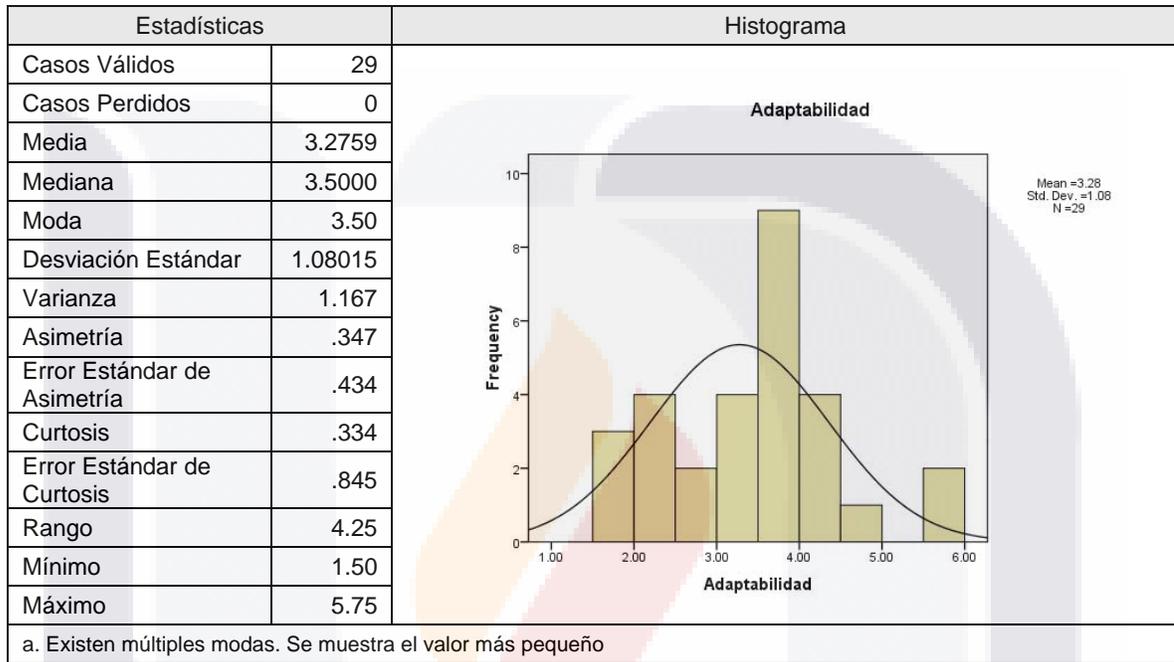


Tabla 25. Estadísticas e Histograma del Factor Adaptabilidad - Primera Iteración

5.2 Segunda Iteración

La presente sección muestra tanto las pruebas de hipótesis como un análisis comparativo en base a la estadística descriptiva, de cada uno de los factores de calidad estudiados. Las pruebas de hipótesis se hicieron a través de “Muestras Independientes” entre ambas iteraciones. Los resultados encontrados se discuten a continuación:

En relación al factor de *Conformidad* se encontró una diferencia de 0.43528 entre la calidad promedio de una iteración a otra (ver Tabla 26), lo cual indica que si se mejoró este aspecto en la segunda versión de la aplicación en un 6.22% aproximadamente.

Estadísticas	Iteraciones	
	1ª	2ª
Casos Válidos	29	26
Casos Perdidos	0	0
Media	2.2276	1.7923
Mediana	2.0000	1.7000
Moda	2.00	1.00 ^a
Desviación Estándar	.64081	.65356
Varianza	.411	.427
Asimetría	.349	.804
Error Estándar de Asimetría	.434	.456
Curtosis	-.163	.358
Error Estándar de Curtosis	.845	.887
Rango	2.60	2.40
Mínimo	1.20	1.00
Máximo	3.80	3.40
a. Existen múltiples modas. Se muestra el valor más pequeño		

Tabla 26. Estadísticas comparativas entre iteraciones del Factor Conformidad

En este caso, se acepta la hipótesis H1a de que en la nueva versión de la aplicación, el factor de calidad *Conformidad* mejoró significativamente ($p \leq 0.016$) (Ver Tabla 27). Esto indica que las modificaciones realizadas a la aplicación, para mejorar la calidad de acuerdo a la percepción del usuario, en cuanto al tipo y tamaño de fuentes utilizado para los textos, así como el diseño de la interfaz en relación a sus necesidades y habilidades, mejoraron sustantivamente.

	Prueba de Levene para Igualdad de Varianzas		Prueba t para Igualdad de Medias						
	F	Sig.	t	df	Sig. (2-filas)	Diferencia Media	Diferencia Error Est.	95% Intervalo de confianza de la Dif.	
								Inferior	Superior
Igualdad de varianzas asumida	.000	.987	2.492	53	.016	.43528	.17470	.08487	.78569
Igualdad de varianzas No asumida			2.489	52.106	.016	.43528	.17490	.08434	.78621

Tabla 27. Prueba de Muestras Independientes del Factor Conformidad

En relación al factor *Exactitud* se encontró una diferencia de 0.51459 entre la calidad promedio de una iteración a otra (ver Tabla 28), lo cual indica que si se mejoró este aspecto en la segunda versión de la aplicación en un 7.35% aproximadamente.

Estadísticas	Iteraciones	
	1 ^a	2 ^a
Casos Válidos	29	26
Casos Perdidos	0	0
Media	2.2069	1.6923
Mediana	2.0000	2.0000
Moda	2.00	2.00
Desviación Estándar	.90156	.54913
Varianza	.813	.302
Asimetría	.503	-.074
Error Estándar de Asimetría	.434	.456
Curtosis	-.259	-.524
Error Estándar de Curtosis	.845	.887
Rango	3.00	2.00
Mínimo	1.00	1.00
Máximo	4.00	3.00
a. Existen múltiples modas. Se muestra el valor más pequeño		

Tabla 28. Estadísticas comparativas entre iteraciones del Factor Exactitud

Aquí se acepta la hipótesis H1b de que en la nueva versión de la aplicación, la calidad de la aplicación percibida por parte del usuario en cuanto al factor *Exactitud*, mejoró significativamente ($p \leq 0.015$) (Ver Tabla 29). Esto indica que el usuario quedó más satisfecho en relación a que la aplicación realiza las operaciones especificadas inicialmente obteniendo resultados correctos.

	Prueba de Levene para Igualdad de Varianzas		Prueba t para Igualdad de Medias						
	F	Sig.	t	df	Sig. (2-filas)	Diferencia Media	Diferencia Error Est.	95% Intervalo de confianza de la Dif.	
								Inferior	Superior
Igualdad de varianzas asumida	3.461	.068	2.520	53	.015	.51459	.20420	.10501	.92416
Igualdad de varianzas No asumida			2.585	46.961	.013	.51459	.19906	.11412	.91506

Tabla 29. Prueba de Muestras Independientes del Factor Exactitud

En relación al factor de *Seguridad* se encontró una diferencia de 0.32162 entre la calidad promedio de una iteración a otra (ver Tabla 30), lo cual indica que si se mejoró este aspecto en la segunda versión de la aplicación en un 4.59% aproximadamente.

Estadísticas	Iteraciones	
	1 ^a	2 ^a
Casos Válidos	29	26
Casos Perdidos	0	0
Media	1.8793	1.5577
Mediana	1.7500	1.3750
Moda	1.00 ^a	1.00
Desviación Estándar	.79513	.59711
Varianza	.632	.357
Asimetría	1.041	.811
Error Estándar de Asimetría	.434	.456
Curtosis	.680	-.197
Error Estándar de Curtosis	.845	.887
Rango	3.00	2.00
Mínimo	1.00	1.00
Máximo	4.00	3.00
a. Existen múltiples modas. Se muestra el valor más pequeño		

Tabla 30. Estadísticas comparativas entre iteraciones del Factor Seguridad

Aquí se rechaza la hipótesis H1c de que en la nueva versión de la aplicación, la *Seguridad* percibida por parte del usuario haya mejorado su calidad significativamente ($p=0.099$) (Ver Tabla 31). Esto era un resultado esperado debido a que no se realizaron mejoras ni en la forma de autenticación de los usuarios, ni en su configuración de perfiles y permisos dentro de la aplicación.

	Prueba de Levene para Igualdad de Varianzas		Prueba t para Igualdad de Medias						
	F	Sig.	t	df	Sig. (2-filas)	Diferencia Media	Diferencia Error Est.	95% Intervalo de confianza de la Dif.	
								Inferior	Superior
Igualdad de varianzas asumida	.963	.331	1.680	53	.099	.32162	.19139	-.06227	.70551
Igualdad de varianzas No asumida			1.707	51.487	.094	.32162	.18845	-.05663	.69986

Tabla 31. Prueba de Muestras Independientes del Factor Seguridad

En relación al factor de *Madurez* se encontró una diferencia de 0.86472 entre la calidad promedio de una iteración a otra (ver Tabla 32), lo cual indica que si se mejoró este aspecto en la segunda versión de la aplicación en un 12.35% aproximadamente.

Estadísticas	Iteraciones	
	1 ^a	2 ^a
Casos Válidos	29	26
Casos Perdidos	0	0
Media	3.1724	2.3077
Mediana	3.0000	2.0000
Moda	2.00	2.00
Desviación Estándar	1.53690	.73589
Varianza	2.362	.542
Asimetría	.768	2.038
Error Estándar de Asimetría	.434	.456
Curtosis	-.549	6.491
Error Estándar de Curtosis	.845	.887
Rango	5.00	4.00
Mínimo	1.00	1.00
Máximo	6.00	5.00
a. Existen múltiples modas. Se muestra el valor más pequeño		

Tabla 32. Estadísticas comparativas entre iteraciones del Factor Madurez

En este caso se acepta la hipótesis H2a de que en la nueva versión de la aplicación, la *Madurez* percibida por parte del usuario mejoró significativamente su calidad ($p \leq 0.012$) (Ver Tabla 33). Este resultado indica que el usuario considera que el sistema maneja adecuadamente los errores que él puede cometer.

	Prueba de Levene para Igualdad de Varianzas		Prueba t para Igualdad de Medias						
	F	Sig.	t	df	Sig. (2-filas)	Diferencia Media	Diferencia Error Est.	95% Intervalo de confianza de la Dif.	
								Inferior	Superior
Igualdad de varianzas asumida	11.623	.001	2.611	53	.012	.86472	.33115	.20052	1.52892
Igualdad de varianzas No asumida			2.704	41.138	.010	.86472	.31981	.21892	1.51053

Tabla 33. Prueba de Muestras Independientes del Factor Madurez

En relación al factor de *Tolerancia a Fallas* se encontró una diferencia de 0.53979 entre la calidad promedio de una iteración a otra (ver Tabla 34), lo cual indica que si se mejoró este aspecto en la segunda versión de la aplicación en un 7.71% aproximadamente.

Estadísticas	Iteraciones	
	1 ^a	2 ^a
Casos Válidos	29	26
Casos Perdidos	0	0
Media	2.6552	2.1154
Mediana	2.0000	2.0000
Moda	2.00	2.00
Desviación Estándar	1.20344	.65280
Varianza	1.448	.426
Asimetría	1.126	-.113
Error Estándar de Asimetría	.434	.456
Curtosis	1.233	-.461
Error Estándar de Curtosis	.845	.887
Rango	5.00	2.00
Mínimo	1.00	1.00
Máximo	6.00	3.00
a. Existen múltiples modas. Se muestra el valor más pequeño		

Tabla 34. Estadísticas comparativas entre iteraciones del Factor Tolerancia a Fallas

Se acepta la hipótesis H2b de que en la nueva versión de la aplicación, la *Tolerancia a Fallas* percibida por el usuario mejoró significativamente su calidad ($p \leq 0.047$) (ver Tabla 35). En este caso los resultados indican que los usuarios encuestados consideran que los mensajes que se muestran dentro de la aplicación cuando ocurre algún error, son manejados e informados de manera adecuada.

	Prueba de Levene para Igualdad de Varianzas		Prueba t para Igualdad de Medias						
	F	Sig.	t	df	Sig. (2-filas)	Diferencia Media	Diferencia Error Est.	95% Intervalo de confianza de la Dif.	
								Inferior	Superior
Igualdad de varianzas asumida	7.386	.009	2.033	53	.047	.53979	.26547	.00732	1.07225
Igualdad de varianzas No asumida			2.096	44.078	.042	.53979	.25755	.02076	1.05882

Tabla 35. Prueba de Muestras Independientes del Factor Tolerancia a Fallas

En relación al factor *Recuperabilidad* se encontró una diferencia de 1.00663 entre la calidad promedio de una iteración a otra (ver Tabla 36), lo cual indica que si se mejoró este aspecto en la segunda versión de la aplicación en un 14.38% aproximadamente.

Estadísticas	Iteraciones	
	1 ^a	2 ^a
Casos Válidos	29	26
Casos Perdidos	0	0
Media	3.2759	2.2692
Mediana	3.0000	2.0000
Moda	2.00	2.00
Desviación Estándar	1.81061	.82741
Varianza	3.278	.685
Asimetría	.953	.357
Error Estándar de Asimetría	.434	.456
Curtosis	-.051	-.098
Error Estándar de Curtosis	.845	.887
Rango	6.00	3.00
Mínimo	1.00	1.00
Máximo	7.00	4.00
a. Existen múltiples modas. Se muestra el valor más pequeño		

Tabla 36. Estadísticas comparativas entre iteraciones del Factor Recuperabilidad

De igual manera, se acepta la hipótesis H2c de que en la nueva versión de la aplicación, la *Recuperabilidad* percibida por el usuario mejoró significativamente su calidad ($p \leq 0.012$) (ver Tabla 37). Los resultados obtenidos muestran que los usuarios consideran que la aplicación muestra los mensajes necesarios en caso de error y que el contar con una ayuda adecuada le facilita su operación y recuperación.

	Prueba de Levene para Igualdad de Varianzas		Prueba t para Igualdad de Medias						
	F	Sig.	t	df	Sig. (2-filas)	Diferencia Media	Diferencia Error Est.	95% Intervalo de confianza de la Dif.	
								Inferior	Superior
Igualdad de varianzas asumida	10.484	.002	2.600	53	.012	1.00663	.38716	.23009	1.78317
Igualdad de varianzas No asumida			2.696	40.125	.010	1.00663	.37333	.25217	1.76109

Tabla 37. Prueba de Muestras Independientes del Factor Recuperabilidad

En relación al factor *Comprensibilidad* se encontró una diferencia de 0.39721 entre la calidad promedio de una iteración a otra (ver Tabla 38), lo cual indica que si se mejoró este aspecto en la segunda versión de la aplicación en un 5.67% aproximadamente.

Estadísticas	Iteraciones	
	1 ^a	2 ^a
Casos Válidos	29	26
Casos Perdidos	0	0
Media	2.2241	1.8269
Mediana	2.0000	2.0000
Moda	2.00	2.00
Desviación Estándar	.77444	.64718
Varianza	.600	.419
Asimetría	2.767	.231
Error Estándar de Asimetría	.434	.456
Curtosis	11.340	-.669
Error Estándar de Curtosis	.845	.887
Rango	4.50	2.00
Mínimo	1.00	1.00
Máximo	5.50	3.00
a. Existen múltiples modas. Se muestra el valor más pequeño		

Tabla 38. Estadísticas comparativas entre iteraciones del Factor Comprensibilidad

También se acepta la hipótesis H3a de que en la nueva versión de la aplicación, la *Comprensibilidad* percibida por el usuario mejoró significativamente su calidad ($p \leq 0.045$) (ver Tabla 39), con las modificaciones realizadas a la aplicación para utilizar componentes más adecuados y facilitar el recordar cómo llevar a cabo las diferentes acciones que la aplicación puede realizar.

	Prueba de Levene para Igualdad de Varianzas		Prueba t para Igualdad de Medias						
	F	Sig.	t	df	Sig. (2-filas)	Diferencia Media	Diferencia Error Est.	95% Intervalo de confianza de la Dif.	
								Inferior	Superior
Igualdad de varianzas asumida	.083	.775	2.051	53	.045	.39721	.19371	.00868	.78575
Igualdad de varianzas No asumida			2.071	52.758	.043	.39721	.19181	.01245	.78198

Tabla 39. Prueba de Muestras Independientes del Factor Comprensibilidad

En relación al factor *Facilidad de Aprendizaje* se encontró una diferencia de 0.65075 entre la calidad promedio de una iteración a otra (ver Tabla 40), lo cual indica que si se mejoró este aspecto en la segunda versión de la aplicación en un 9.30% aproximadamente.

Estadísticas	Iteraciones	
	1 ^a	2 ^a
Casos Válidos	29	26
Casos Perdidos	0	0
Media	2.4713	1.8205
Mediana	2.3333	2.0000
Moda	1.67	2.00
Desviación Estándar	.92375	.55961
Varianza	.853	.313
Asimetría	.791	.347
Error Estándar de Asimetría	.434	.456
Curtosis	.209	.899
Error Estándar de Curtosis	.845	.887
Rango	3.67	2.33
Mínimo	1.00	1.00
Máximo	4.67	3.33
a. Existen múltiples modas. Se muestra el valor más pequeño		

Tabla 40. Estadísticas comparativas entre iteraciones del Factor Facilidad de Aprendizaje

La hipótesis H3b de que en la nueva versión de la aplicación la *Facilidad de Aprendizaje* percibida por el usuario mejoró significativamente su calidad, se acepta ($p \leq 0.003$) (ver Tabla 41). Esto indica que los usuarios consideran que la forma de acceder a las diferentes tareas de la aplicación les parece adecuada, así como la cantidad de movimientos para realizarlas y que les es fácil de recordar cómo llevarlas a cabo, por lo que el manejo de la aplicación puede considerarse fácil y rápido de aprender.

	Prueba de Levene para Igualdad de Varianzas		Prueba t para Igualdad de Medias						
	F	Sig.	t	df	Sig. (2-filas)	Diferencia Media	Diferencia Error Est.	95% Intervalo de confianza de la Dif.	
								Inferior	Superior
Igualdad de varianzas asumida	6.465	.014	3.114	53	.003	.65075	.20895	.23166	1.06985
Igualdad de varianzas No asumida			3.196	46.827	.002	.65075	.20364	.24104	1.06046

Tabla 41. Prueba de Muestras Independientes del Factor de Facilidad de Aprendizaje

En relación al factor *Operabilidad* se encontró una diferencia de 0.77653 entre la calidad promedio de una iteración a otra (ver Tabla 42), lo cual indica que si se mejoró este aspecto en la segunda versión de la aplicación en un 11.09% aproximadamente.

Estadísticas	Iteraciones	
	1 ^a	2 ^a
Casos Válidos	29	26
Casos Perdidos	0	0
Media	2.6034	1.8269
Mediana	2.5000	2.0000
Moda	2.50	2.00
Desviación Estándar	.90019	.63154
Varianza	.810	.399
Asimetría	.969	.591
Error Estándar de Asimetría	.434	.456
Curtosis	2.845	.761
Error Estándar de Curtosis	.845	.887
Rango	4.50	2.50
Mínimo	1.00	1.00
Máximo	5.50	3.50
a. Existen múltiples modas. Se muestra el valor más pequeño		

Tabla 42. Estadísticas comparativas entre iteraciones del Factor Operabilidad

Igualmente, se acepta la hipótesis H3c de que en la nueva versión de la aplicación, la *Operabilidad* percibida por el usuario mejoró significativamente su calidad ($p \leq 0.001$)

(ver Tabla 43) al tratar de simplificar la forma de navegación entre las diferentes opciones (pestañas) que se utilizan dentro de la aplicación, lo que facilita su operación.

	Prueba de Levene para Igualdad de Varianzas		Prueba t para Igualdad de Medias						
	F	Sig.	t	df	Sig. (2-filas)	Diferencia Media	Diferencia Error Est.	95% Intervalo de confianza de la Dif.	
								Inferior	Superior
Igualdad de varianzas asumida	1.336	.253	3.663	53	.001	.77653	.21202	.35127	1.20178
Igualdad de varianzas No asumida			3.732	50.228	.000	.77653	.20805	.35870	1.19435

Tabla 43. Prueba de Muestras Independientes del Factor de Operabilidad

En relación al factor *Tiempo de Respuesta* se encontró una diferencia de 0.69363 entre la calidad promedio de una iteración a otra (ver Tabla 44), lo cual indica que si se mejoró este aspecto en la segunda versión de la aplicación en un 9.91% aproximadamente.

Estadísticas	Iteraciones	
	1 ^a	2 ^a
Casos Válidos	29	26
Casos Perdidos	0	0
Media	2.6552	1.9615
Mediana	2.0000	2.0000
Moda	2.00	2.00
Desviación Estándar	1.14255	.77360
Varianza	1.305	.598
Asimetría	.900	.068
Error Estándar de Asimetría	.434	.456
Curtosis	.436	-1.269
Error Estándar de Curtosis	.845	.887
Rango	4.00	2.00
Mínimo	1.00	1.00
Máximo	5.00	3.00
a. Existen múltiples modas. Se muestra el valor más pequeño		

Tabla 44. Estadísticas comparativas entre iteraciones del Factor Tiempo de Respuesta

Se acepta también, la hipótesis H4a de que en la nueva versión el *Tiempo de Respuesta* percibido por el usuario mejoró significativamente su calidad ($p \leq 0.012$) (ver Tabla 45), ya que los usuarios encontraron mayor la rapidez con las que la aplicación realiza las operaciones solicitadas.

	Prueba de Levene para Igualdad de Varianzas		Prueba t para Igualdad de Medias						
	F	Sig.	t	df	Sig. (2-filas)	Diferencia Media	Diferencia Error Est.	95% Intervalo de confianza de la Dif.	
								Inferior	Superior
Igualdad de varianzas asumida	3.163	.081	2.605	53	.012	.69363	.26627	.15957	1.22770
Igualdad de varianzas No asumida			2.659	49.469	.011	.69363	.26083	.16960	1.21767

Tabla 45. Prueba de Muestras Independientes del Factor Tiempo de Respuesta

En relación al factor *Adaptabilidad* se encontró una diferencia de 0.85279 entre la calidad promedio de una iteración a otra (ver Tabla 46), lo cual indica que si se mejoró este aspecto en la segunda versión de la aplicación en un 12.18% aproximadamente.

Estadísticas	Iteraciones	
	1ª	2ª
Casos Válidos	29	26
Casos Perdidos	0	0
Media	3.2759	2.4231
Mediana	3.5000	2.2500
Moda	3.50	2.00
Desviación Estándar	1.08015	.82997
Varianza	1.167	.689
Asimetría	.347	.271
Error Estándar de Asimetría	.434	.456
Curtosis	.334	-.258
Error Estándar de Curtosis	.845	.887
Rango	4.25	3.25
Mínimo	1.50	1.00
Máximo	5.75	4.25
a. Existen múltiples modas. Se muestra el valor más pequeño		

Tabla 46. Estadísticas comparativas entre iteraciones del Factor Adaptabilidad

En este caso, también se acepta la hipótesis H5a de que en la nueva versión, la *Adaptabilidad* percibida por el usuario mejoró significativamente su calidad ($p \leq 0.002$) (ver Tabla 47). Esto en relación a la modificación de los colores utilizados en los elementos de la interfaz.

	Prueba de Levene para Igualdad de Varianzas		Prueba t para Igualdad de Medias						
	F	Sig.	T	df	Sig. (2-filas)	Diferencia Media	Diferencia Error Est.	95% Intervalo de confianza de la Dif.	
								Inferior	Superior
Igualdad de varianzas asumida	1.330	.254	3.254	53	.002	.85279	.26204	.32721	1.37836
Igualdad de varianzas No asumida			3.301	51.841	.002	.85279	.25831	.33440	1.37117

Tabla 47. Prueba de Muestras Independientes del Factor Adaptabilidad

Finalmente, la Tabla 48 muestra de manera concentrada los resultados de las pruebas de hipótesis realizadas, donde se puede observar que se aceptaron 10 de las 11 hipótesis planteadas, lo que confirma que de manera general la calidad de la aplicación en la segunda versión si se mejoró en base a la evaluación y modificación de la versión inicial.

Hipótesis	Resultado	Significancia
H1a	Se acepta	$p \leq 0.016$
H1b	Se acepta	$p \leq 0.015$
H1c	Se Rechaza	$p = 0.099$
H2a	Se acepta	$p \leq 0.012$
H2b	Se acepta	$p \leq 0.047$
H2c	Se acepta	$(p \leq 0.012)$
H3a	Se acepta	$p \leq 0.045$
H3b	Se acepta	$p \leq 0.003$
H3c	Se acepta	$p \leq 0.001$
H4a	Se acepta	$p \leq 0.012$
H5a	Se acepta	$p \leq 0.002$

Tabla 48. Resultados concentrados de la Prueba de Hipótesis

6. CONCLUSIONES

6.1 Conclusiones Generales

De acuerdo a los resultados encontrados, se pudo comprobar que el integrar al usuario en la evaluación de una aplicación de software, si ayudó a incrementar su calidad en una siguiente versión, en 10 de los 11 factores analizados. A continuación se detalla las conclusiones relacionadas a cada factor evaluado:

La *Conformidad*, que forma parte de la característica general de *Funcionalidad*, se mejoró significativamente entre una versión y otra, principalmente en los aspectos de diseño de la interfaz: al realizar un reacomodo de los elementos de la pantalla, reorganizar el orden de los apartados, etc. De esta forma, al desarrollar una nueva aplicación de software, se debe cuidar primero que de manera general la aplicación realice todas las operaciones especificadas como requerimientos y después que la interfaz sea adecuada a las habilidades y necesidades de los usuarios, que el tipo y tamaño de fuentes que se utilizan les sea agradable, etc.

La *Exactitud*, que también forma parte de la característica general de *Funcionalidad*, se mejoró significativamente entre una versión y otra, a pesar de no haber realizado modificaciones del funcionamiento base. Esto nos indica que la aplicación analizada cumple con satisfacer la mayoría de las necesidades identificadas por sus diferentes usuarios y aunque de manera cualitativa, los usuarios expresaron diferentes requerimientos, esto se interpreta como modificaciones de mejora e incremento de funcionalidad. En este aspecto, al desarrollar nuevas aplicaciones de software, se debe cuidar que éstas siempre proporcionen al usuario resultados exactos y precisos de acuerdo a la precisión establecida en los requerimientos.

Por otro lado, en el aspecto de *Seguridad*, se rechazó la hipótesis de mejora en la segunda versión. Este era un resultado esperado ya que no se realizaron modificaciones ni a la forma de la autenticación de los usuarios, ni a las restricciones de acceso por tipo de usuario, etc. Aunque como se añadió un archivo de ayuda donde se explicaba la funcionalidad general de la aplicación, algunos usuarios incluso calificaron la segunda versión con un criterio menor al de la versión original, porque identificaron funcionalidad

que pudiera interesarles y con la que actualmente no cuentan debido a su perfil de usuario especificado. En este punto también es importante destacar que debido a que la aplicación evaluada es de escritorio, este factor *Seguridad* no parece ser tan relevante para el usuario. Pero en caso de que el instrumento de evaluación se hubiese realizado sobre una aplicación web, cambiaría el panorama del usuario por ser considerado de manera natural un medio vulnerable y se vuelve inherente que la aplicación sea desarrollada para que funcione de manera segura y confiable.

En cuanto a lo *Madurez, Tolerancia a Fallas y Recuperabilidad* que forman parte de la característica general de *Confiabilidad*, se encontró que todas mejoraron en la segunda versión, por lo que se deduce que el sistema cuenta con un control adecuado de errores y que el agregar la ayuda en línea, reforzó este punto. Sin embargo, es importante mencionar que este aspecto es de los que más requieren mejoras futuras ya que de manera cualitativa los usuarios expresaron que algunos mensajes de error no les son del todo claro porque están expresados de manera muy técnica. De esta forma, se puede ver que los factores que se relacionan al Control de Errores de una aplicación son de los más importantes para el usuario para determinar su calidad, y por lo tanto, los desarrolladores deben cuidar ampliamente este aspecto. Sobre todo en cuestión de qué tan entendibles son los mensajes para el usuario y en el cómo reaccionará la aplicación ante los errores o fallas que pudieran suceder para que el usuario lo considere confiable, además de siempre incluir una ayuda en línea completa.

En relación a la *Comprensibilidad*, la cual forma parte de la característica general de *Facilidad de Uso*, si se encontró mejora en la segunda versión de la aplicación. Siendo este uno de los puntos que más se planeó mejorar al reacomodar la distribución de algunos componentes de la interfaz para hacerla más entendible, añadiendo títulos más claros en los diferentes paneles de datos etc.

La *Facilidad de Aprendizaje* que también forma parte de la *Facilidad de Uso*, si fue mejorada en la segunda versión. Esto se logró al buscar que las acciones que se pueden realizar en la aplicación, se adecuaron a los estándares usados en el sistema general, para ayudar a que se recordara su operación de manera más natural y agilizar su aprendizaje. En este factor, se identificó que como se partió de una versión que ha sido

utilizada desde ya hace algunos años, los usuarios al ver la nueva versión buscaban el orden o forma que tenía originalmente y algunos expresaron que por ser diferente se les complicó un poco su utilización, pero en la práctica con el uso constante, esto debe tender a normalizarse.

La *Operabilidad* también parte de la característica general de *Facilidad de Uso*, si mejoró en la segunda versión con las modificaciones realizadas, ya que se buscó la forma de optimizar el manejo de la aplicación, sobre todo en la pestaña de control principal.

De manera general, en base a los tres factores seleccionados para evaluar la *Facilidad de Uso* de una aplicación, se debe considerar que al realizar nuevas aplicaciones, los desarrolladores cuiden que éstas realmente sean fáciles de usar, aprender y operar: estableciendo estándares en la forma de realizar las operaciones, buscando la forma de navegación más sencilla, utilizando componentes similares para acciones similares, buscando qué tipo de componentes es el más adecuado para cada acción, tratando de que la interfaz tenga una distribución agradable, etc. Para atender a estos aspectos es muy importante que los desarrolladores analicen y consideren la demografía de los usuarios. Es decir, que las interfaces sean desarrolladas atendiendo a las características de edad, género, educación, capacidades físicas, entre otras, de los usuarios finales.

En cuanto al *Tiempo de Respuesta* que forma parte de la característica general de *Eficiencia*, si se mejoró en la segunda versión, pero es importante indicar que también es un aspecto que puede mejorarse todavía más. Esta mejora dependerá de un análisis integrado tanto de la codificación y estructura de la aplicación, como de la infraestructura tecnológica involucrada en cuestión de base de datos, servidores de red y aplicaciones, etc. De acuerdo a este factor, al desarrollarse nuevas aplicaciones de software, éstas deben planearse en el sentido de que la infraestructura y codificación de las mismas puede impactar en la velocidad con las que se van a ejecutar las diferentes acciones que la aplicación puede llevar a cabo y que éste es uno de los puntos que el usuario más toma en cuenta para determinar su percepción de calidad referente a la aplicación.

En relación a la *Adaptabilidad* que forma parte de la característica general de *Portabilidad*, se encontró que la segunda versión de la aplicación si mejoró, lo que significa que las

características de la interfaz en cuanto a colores, elementos, etc. fueron percibidas más atractivas por parte del usuario. Aunque de acuerdo a los resultados individuales de cada iteración, se pudo identificar que los usuarios están conformes con estos aspectos, pero su grado de aceptación no llega a ser total, porque ésta se basa principalmente en el criterio de que se está tomando un estándar y no tanto en que los colores e interfaz realmente les parezcan agradables. Este punto es importante al desarrollar nuevas aplicaciones, porque el cuidar detalles como que la aplicación cuente con un módulo que le permita al usuario personalizar su configuración puede influir positivamente en su percepción general de la calidad de la aplicación.

Finalmente, de manera general con todo el análisis realizado, se pudo comprobar que la evaluación de productos de software en base a factores deseables de calidad establecidos e involucrando de manera activa a los usuarios finales, puede ayudar a obtener productos con una calidad mayor y cumplir como se mencionó desde un inicio, con uno de los principales objetivos de la calidad, que es la satisfacción del cliente.

6.2 Limitaciones

Durante el desarrollo de la aplicación del instrumento diseñado se encontraron las siguientes limitantes:

La aplicación que se utilizó como caso de estudio, fue programada originalmente por un desarrollador diferente a quien aplicó la evaluación, lo cual podría haber impactado. Sobre todo en el tiempo de realización de las modificaciones necesarias. De igual manera, esta limitante no permitió realizar mejoras más significativas en la segunda versión ya que se requería de un tiempo mayor de análisis de la aplicación para identificar las necesidades específicas a atacar.

Otra limitante para poder lanzar una versión con cambios significativos, fue que la codificación de la aplicación no se encuentra concentrada en su lógica de negocio dentro de la base de datos sino en el lenguaje de programación utilizado para la interfaz, lo cual impedía optimizar tiempos de respuesta a menos que se replanteara en general su codificación.

Para la mejora de los aspecto de confiabilidad de manera más específica, se encontró la limitante de que como el Control de Errores es un módulo completo independiente dentro del sistema general, el realizar los cambios necesarios para mejorar la percepción del usuario en este punto, involucraba el análisis y modificación de una segunda aplicación, lo cual no fue posible debido a los tiempos con los que se contaba para esta investigación.

Por otro lado, otra limitante fue que la versión a evaluar para la segunda iteración no se incluyó dentro del menú formal del sistema general (SIIMA) ya que siguiendo la metodología de desarrollo utilizada actualmente por el Área de Sistemas, al no ser una aplicación 100% probada no se puede publicar ni sustituir la versión actual, por lo que fue necesario evaluar la aplicación de forma externa o suelta, lo que ocasionó que algunos usuarios mostraran mas renuencia a su evaluación o en su caso argumentaban que era necesario tenerla en su menú formal para poder utilizarla ya de manera oficial y poder evaluarla de manera más específica.

Una limitante no esperada, fue el punto de que en las áreas administrativas donde se aplicó el instrumento cuentan con políticas de restricción en sus equipos de cómputo, por lo que la segunda iteración fue más complicada de realizar al tener que ejecutar la aplicación de manera suelta por el punto mencionada anteriormente.

Finalmente, una limitante general fue la carga de trabajo de los usuarios involucrados, ya que esto alentó la recopilación de datos y no permitió obtener un mayor número de encuestados en relación a la lista de usuarios planeada inicialmente.

6.3 Trabajos Futuros

En los resultados obtenidos a través de esta investigación, se puede observar que aunque las iteraciones realizadas muestran niveles de significancia considerables, que indican una mejora general de la calidad del producto entre la primera y segunda versión de la aplicación de software evaluada, es recomendable la realización de una tercera iteración. El objetivo de esta siguiente iteración, sería realizar cambios importantes a la aplicación que ayuden a confirmar la mejora de la calidad en el producto que se está evaluando. Para identificar qué cambios realizar, se seguiría el mismo esquema de analizar los datos obtenidos con la evaluación de la versión actual, pero considerando además, minimizar las limitantes encontradas en el presente estudio y poder obtener resultados más exactos.

Es necesario revisar los requerimientos generales y adecuar la aplicación para mejorar la conformidad, unificar la lógica de negocio dentro de la base de datos para optimizar tiempos de respuesta, publicar la versión dentro del menú oficial de SIIMA para facilitar su evaluación y evitar problemas de restricciones por políticas y de manera inherente minimizar el tiempo empleado por parte del usuario para evaluar la aplicación. Por último se consideraría la adecuación del Módulo General de Control de Errores, para tratar de hacerlo más entendible para el usuario final, que como se pudo observar al aplicar el instrumento, es de los factores que los usuarios más fácilmente pueden identificar.

Por otro lado, para realizar una evaluación de la calidad de un producto de software de manera más completa, es recomendable integrar la evaluación de los elementos del modelo que no fueron considerados durante el presente estudio, por ser considerados aspectos propios del proceso de desarrollo de software que no podían ser evaluados tangiblemente por el usuario. Entre estos elementos se encuentra: a) La *interoperabilidad* que es parte de la característica de calidad de *Funcionalidad*, b) el *uso óptimo de recursos* que forma parte de la característica de calidad de *Eficiencia*, c) los subatributos que definen que un producto de software cumple con la característica de calidad de *Facilidad de Mantenimiento*, tales como *Facilidad de Análisis*, *Facilidad de cambio*, *Estabilidad*, *Facilidad de Prueba* y *Facilidad de Reutilización*, d) el subatributo de *Capacidad de Instalación* que forma parte de la característica de calidad de *Portabilidad*. Todos estos elementos pueden evaluarse a través de la utilización de métricas de software (tales como

Cohesividad, Acoplamiento, Complejidad Ciclomática, etc.) para poder determinar su impacto en el nivel de calidad final del producto.

De manera general, se espera que en base a los resultados encontrados durante esta investigación, se considere la integración del instrumento diseñado como parte de la metodología de desarrollo formal del Área de sistemas de la Universidad Autónoma de Aguascalientes, lo anterior como herramienta de apoyo para ayudar a ser más eficiente la liberación de nuevas aplicaciones, aprovechando el marco general del Sistema de Gestión de Calidad que se está llevando actualmente dentro de la institución.



BIBLIOGRAFIA

1. Amasaki, S. T. Yoshitomi, et al. : “A new Challenge for Applying Time Series Metrics Data to Software Quality Estimation”, Software Quality Journal, pp. 177-193. (2005)
2. Deming, W.E., "Calidad, Productividad y Competitividad". Ediciones Díaz de Santos. (1989)
3. DeMarco, T. : “Management Can Make Quality (Im)possible”, Cutter IT Summit, Boston, (Abril 1999)
4. Dolado, C., J.J., Fernández, S. L.: Medición para la gestión en la Ingeniería del Software, Ra-Ma (2000)
5. Ejiogu, L. : Software Engineering with Formal Metrics, QED Publishing (1991)
6. Feigenbaum, A. V: Control total de la calidad, CECSA, México (1986)
7. Fenton E. Norman: Software Metrics A rigorous approach, Chapman & Hall, Primera Edición, (1991)
8. Garvin D. A. : Managin quality, Free Press (1988)
9. Glass, R. : “Defining Quality Intuitively”, IEEE Software, pp. 103-104,107(Mayo 1998)
10. Gómez Reynoso, J., M. and Brizuela Sandoval M. R, Participación de los Usuarios en la Evaluación de Sistemas de Información. Americas Conference on Information Systems Toronto, ON, Canada, Association of Information Systems:10 (2008)
11. González D. H., “Las métricas de software y su uso en la región”, Tesis de Licenciatura, Departamento de Ingeniería en Sistemas Computacionales, Escuela de Ingeniería, Universidad de las Américas-Puebla (2001).

12. Hernández, R., Fernández, C., Baptista, P. : Metodología de la Investigación, Segunda Edición, Mc Graw Hill. (1998).
13. Humphrey, W., : Managing the Software Process, Reading, MA: Adisson-Wesley. Caps. 24,25 (1989)
14. IEEE Software Engineering Standards Collection, Standard Glossary of Software Engineering Terminology. IEEE, Std 610.12-190 (1994).
15. ISO 9000:2000, "Sistema de Gestión de la Calidad - Principios y Vocabulario" (2000).
16. ISO/IEC 9126 : Information Technology - Software Product Evaluation – Quality characteristics and guidelines for their use (1991)
17. ISO/IEC 90003:2004, Software e Ingeniería de Sistemas – Guía para la aplicación de la Norma ISO 9001:2000 para el software (2004).
18. Juran, M.: Juran y la planificación de la calidad, Madrid: Diaz de Santos (1990).
19. Lutowski, R. : Software Requirements Encapsulation, Quality and Reuse, Auerbach Publicaciones Taylor& Francis group (2005)
20. McCall, J., Richards, P., Walters, G. : Factors in Software Quality, Tres volúmenes, NTIS AD-A049-014, 015, 055, (1977).
21. Moreno M, A., Sánchez, S., M.: "Patrones de Usabilidad: Mejora de la Usabilidad del Software desde el momento de Arquitectónico", Universidad Politécnica de Madrid, España (2003)
22. Ortiz, U., García M. :Metodología de la investigación : el proceso y sus técnicas, Limusa México (2003)

23. Piattini, M., García, F. : Calidad en el desarrollo y mantenimiento del software, Ra-Ma Editorial, Madrid (2003)
24. Pfleeger, S. L. : Ingeniería de Software Teoría y Práctica, Primera Edición, Prentice Hall, (2002)
25. Pressman, R. : Ingeniería del Software, Un Enfoque Práctico, Sexta Edición, McGraw Hill Interamericana, México, (2005)
26. Ramani, A. : "Measuring Client Satisfaction Using Quality in Use", The quarterly journal of the TickIT software quality certification scheme, (Febrero 2007)
27. Sommerville, I. : Ingeniería de Software, Sexta Edición, Addison Wesley, México (2002)
28. Zayaraz, G., P. Thambidurai, et al.: "Software Quality Assurance Through COSMIC FFP", ACM SIGSOFT Software Engineering Notes (2005).