



UNIVERSIDAD AUTÓNOMA DE AGUASCALIENTES

CENTRO DE CIENCIAS BÁSICAS

MAESTRÍA EN CIENCIAS EXACTAS SISTEMAS, Y DE LA INFORMACIÓN

ÁREA: INGENIERÍA DE SOFTWARE

**TRABAJO DE GRADO PARA OBTENER EL TÍTULO DE MAESTRO EN
INGENIERIA DE SOFTWARE**

TESIS

**“ADAPTACIÓN DE UN MODELO DE DESARROLLO DE
SOFTWARE PARA LA CONSTRUCCIÓN DE SOFTWARE
USABLE CON CALIDAD”**

PRESENTA

I.C. Fabio Armando García Toribio

DIRECTOR DE TESIS

Juan Manuel Gómez Reynoso, Ph.D.

ASESORES

Dra. Laura Garza González

Dra. Estela Lizbeth Muñoz Andrade

Cd. Universitaria – Noviembre de 2010

Centro de Ciencias Básicas

**I.C. FABIO ARMANDO GARCÍA TORIBIO
PASANTE DE LA MAESTRIA EN CIENCIAS EXACTAS,
SISTEMAS Y DE LA INFORMACIÓN
P R E S E N T E .**

Estimado (a) alumno (a) García:

Por medio de este conducto me permito comunicar a Usted que habiendo recibido los votos aprobatorios de los revisores de su trabajo de tesis y/o caso práctico titulado: **"ADAPTACIÓN DE UN MODELO DE DESARROLLO DE SOFTWARE PARA LA CONSTRUCCIÓN DE SOFTWARE USABLE CON CALIDAD"**, hago de su conocimiento que puede imprimir dicho documento y continuar con los trámites para la presentación de su examen de grado.

Sin otro particular me permito saludarle muy afectuosamente.

A T E N T A M E N T E
Aguascalientes, Ags., 22 de noviembre de 2010
"SE LUMEN PROFERRE"
EL DECANO

DR. FRANCISCO JAVIER ÁLVAREZ RODRÍGUEZ



c.c.p.- Archivo
FJAR,mjda

Dr. Francisco Javier Álvarez Rodríguez

Decano de Ciencias Básicas

Por medio de este conducto, autorizo al tesista:

I.C. Fabio Armando García Toribio

La impresión de su documento de tesis titulado "**Adaptación de un Modelo de Desarrollo de Software para la Construcción de Software Usable con Calidad**", ya que cumple con los requisitos de contenido y formas exigidas por la Universidad Autónoma de Aguascalientes.

Sin más por el momento, quedo a sus órdenes para cualquier aclaración al respecto.

ATENTAMENTE

Juan Manuel Gómez Reynoso, Ph.D.

Director de Tesis

Dra. Laura Garza González

Asesor de Tesis

Dra. Estela Lizbeth Muñoz Andrade

Asesor de Tesis

Aguascalientes, Ags, Noviembre 2010.

AGRADECIMIENTOS

“Aunque ande en valle de muerte y de sombra no temeré mal alguno, porque tú estás conmigo, tu vara y tu cayado me infundirán aliento.” salmo 23.

Antes que nada quiero agradecer al altísimo, *Dios padre* celestial, por permitirme alcanzar una meta más en mi vida.

Agradezco a *Cristo Jesús* señor nuestro, escudo y adarga en mi vida. Porque con tu amor infinito, y con tu sangre preciosa nos hiciste salvos, a ti agradezco por tu amor y guía no sólo durante el transcurso de mi carrera y culminación de ésta, sino durante toda mi vida.

A ti que has estado conmigo desde el momento en que nací, tú que con tus inspiraciones me guías por el buen camino, tú que has salvado mi vida en las situaciones de peligro, gracias *ángel de mi guarda*.

Agradezco a mis padres por su motivación constante, su confianza en mí y por su amor incondicional.

A mi hermana por su apoyo no sólo en la elaboración de esta tesis, sino a lo largo de toda mi vida.

De una manera muy particular agradezco a mi director de tesis Juan Manuel Gómez Reynoso, Ph.D., ya que gracias a su apoyo técnico y moral, logré la consumación de esta investigación, gracias por su dedicación y paciencia a lo largo de este proyecto.

Finalmente a todas las personas que me apoyaron durante el trayecto de mi posgrado, y proceso de titulación.

RESUMEN

El software se ha vuelto un elemento indispensable para toda organización, un término común y entendido aun para aquellos ajenos al área. En la actualidad, es común encontrar sistemas informáticos en casi cualquier lugar, robustos sistemas bancarios, sistemas contables, y administrativos. Cada uno enfocado a un dominio en particular, haciendo de aquellos trabajos complejos actividades simples de realizar. Sin embargo, esto no es tan sencillo dado que un punto fundamental para la correcta aplicación de los sistemas informáticos estriba en el grado de eficiencia en el cual, el usuario puede interactuar y completar las tareas que desea de forma satisfactoria. Por ello se vuelve indispensable establecer mecanismos de evaluación detallada de los productos de software, para detectar inconvenientes antes de su liberación a los usuarios.

En esta investigación se buscó 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, así como de métricas que miden aspectos propios del desarrollo del producto. Para lograr esto se hizo una integración con dos modelos de calidad, determinando cuales aspectos debían medirse con el usuario y cuales con métricas en el desarrollo. Luego de esto se diseñó una herramienta de evaluación para medir la percepción de calidad del usuario, y se definieron métricas para el desarrollo del producto.

Se evaluó un sistema web de carácter turístico desarrollado para esta investigación, la evaluación se hizo por medio de muestras pareadas, se realizaron 2 iteraciones, tomando los resultados de la primera evaluación para hacer las modificaciones pertinentes, buscando garantizar que los productos de software se liberen de manera formal con un nivel de calidad óptimo.

Se evaluaron 16 hipótesis establecidas de las cuales se aprobaron 14 y se rechazaron 2, cada hipótesis equivale a un factor de calidad medido.

Finalmente se utilizó estadística descriptiva para el análisis de los datos recolectados.

ÍNDICE

CAPÍTULO 1. INTRODUCCIÓN	2
CAPÍTULO 2. REVISIÓN DE LITERATURA.....	4
2.1 Descripción de Teorías Base.....	4
2.1.1 Calidad.....	4
2.1.2 Calidad en Software.....	6
2.1.3 Calidad del proceso y del producto.	8
2.1.4 Aseguramiento de Calidad	9
2.1.5 Modelos de Calidad	10
2.1.5.1 El Modelo de McCall.....	11
2.1.5.2 Modelo ISO/IEC 9126.....	14
2.1.5.4 Relación entre modelos.....	16
2.1.5.5 Factores de Calidad	18
2.1.6 Métricas de software.....	19
2.1.7 Garantía de Calidad.....	21
CAPÍTULO 3. FORMULACIÓN DEL PROBLEMA DE INVESTIGACIÓN	23
3.1. Descripción del problema de Investigación y su relevancia	23
3.2 Pregunta de Investigación	25
3.3 Objetivo General	25
3.3.2 Objetivos Específicos.....	25
3.4 Hipótesis	26
CAPÍTULO 4. DESARROLLO DE LA INVESTIGACIÓN.....	29
4.1 Materiales y Métodos	29
4.2 Estudio Piloto	32

4.3 Estudio Final	40
4.3.1 Perfil del Usuario.....	40
4.3.2 Muestra	41
4.3.3 Recopilación de Información	42
4.3.3.1 Primer Iteración	43
4.3.3.2 Segunda Iteración	47
CAPÍTULO 5. ANÁLISIS DE DATOS	50
5.1 Evaluación por parte del usuario.....	50
5.1.2 Primera iteración.....	50
5.1.2 Segunda iteración	65
5.2 Calculo de Métricas en el Desarrollo	95
5.2.1 Facilidad de análisis.....	95
5.2.1.1 Primera iteración	97
5.2.1.2 Segunda Iteración	99
5.2.2 Facilidad de Cambio	99
5.2.2.1 Primera Iteración	100
5.2.2.2 Segunda Iteración	100
5.2.3 Facilidad de Prueba	101
5.2.3.1 Primera Iteración	102
5.2.3.2 Segunda Iteración	103
5.2.4 Estabilidad	104
5.2.4.1 Primera Iteración	104
5.2.4.2 Segunda Iteración	105
5.2.5 Adaptabilidad	105
5.2.5.1 Primera Iteración	107
5.2.5.2 Segunda Iteración	107
5.2.6 Conformidad	108
5.2.6.1 Primera Iteración	110
5.2.6.2 Segunda Iteración	111
5.3 Resultados de las Hipótesis	112
6. CONCLUSIONES	113

6.1 Conclusiones Generales 113

6.2 Limitaciones 116

6.3 Trabajos Futuros 118

7. BIBLIOGRAFÍA 119



ÍNDICE DE FIGURAS

Figura 1. Modelo de Calidad de McCall (adaptado de Pressman 2005).	12
Figura 2. Equivalencias McCall – ISO 9126 (adaptado de Castañeda, Gómez-Reynoso et al. 2009).	17
Figura 3. Modelo de Desarrollo Base (adaptado de Gomez-Reynoso y Brizuela-Sandoval 2008).	30
Figura 4. Encuesta Nacional sobre Disponibilidad y Uso de las Tecnologías de la Información en los Hogares (INEGI 2008).	41
Figura 5. Genero del perfil del Internauta según AMIPCI (2009).	41
Figura 6. Edad – Primer Iteración	43
Figura 7. Género– Primer Iteración	44
Figura 8. Nivel de Estudios – Primer Iteración	44
Figura 9. Experiencia uso de PC – Primer Iteración	45
Figura 10. Uso semanal Internet – Primer Iteración	46
Figura 11. Uso semanal PC – Primer Iteración	46
Figura 12. Edad – Segunda Iteración.	47
Figura 13. Género – Segunda Iteración.	48
Figura 14. Experiencia uso PC – Segunda Iteración.	48
Figura 15. Uso semanal Internet – Segunda Iteración	49
Figura 16. Histogramas - ítems factor <i>Operabilidad</i>	52
Figura 17. Histogramas - ítems factor <i>facilidad de aprendizaje</i>	53
Figura 18. Histogramas - ítems factor <i>Atractivo</i>	55
Figura 19. Histogramas - ítems factor <i>Comprensibilidad</i>	56
Figura 20. Histogramas - ítems factor <i>conformidad</i>	57
Figura 21. Histogramas - ítems factor <i>Exactitud</i>	59
Figura 22. Histogramas - ítems factor <i>Madurez</i>	60
Figura 23. Histogramas - ítems factor <i>tolerancia a fallas</i>	62
Figura 24. Histogramas - ítems factor <i>Recuperabilidad</i>	63
Figura 25. Histogramas - ítems factor <i>Tiempo de Respuesta</i>	65
Figura 26. Comparativa de Histogramas entre iteraciones de <i>Operabilidad</i>	67
Figura 27. Comparativa de Histogramas entre iteraciones de <i>Facilidad de Aprendizaje</i> . ..	70
Figura 28. Comparativa de Histogramas entre iteraciones del <i>Atractivo</i>	73

Figura 29. Comparativa de Histogramas entre iteraciones de *comprensibilidad*. 76

Figura 30. Comparativa de Histogramas entre iteraciones de *Conformidad*. 79

Figura 31. Comparativa de Histogramas entre iteraciones de *Exactitud*. 82

Figura 32. Comparativa de Histogramas entre iteraciones de *Madurez*. 85

Figura 33. Comparativa de Histogramas entre iteraciones de *Tolerancia a Fallas*. 88

Figura 34. Comparativa de Histogramas entre iteraciones de *Recuperabilidad*. 91

Figura 35. Comparativa de Histogramas entre iteraciones de *Tiempo de Respuesta*. 93



ÍNDICE DE TABLAS

Tabla 1. Factores de calidad del modelo de McCall (adaptado de Pressman 2005).	13
Tabla 2. Características modelo ISO/IEC 9126-1(ISO 2001).	15
Tabla 3. Factores de calidad en que se basa la presente investigación.	19
Tabla 4. Equivalencias McCall – ISO 9126.	31
Tabla 5. Factores que se midieron en la investigación.	32
Tabla 6. Resultados la Prueba Piloto	33
Tabla 7. Histogramas del estudio piloto.	39
Tabla 8. Tabla de Frecuencias de Edad – Primer Iteración.....	43
Tabla 9. Tabla de frecuencias de Género – Primer Iteración	44
Tabla 10. Tabla de frecuencias de Experiencia uso de PC – Primer Iteración	45
Tabla 11. Tabla de frecuencias uso semanal de Internet – Primer Iteración	45
Tabla 12. Tabla de frecuencias uso semanal de Internet – Primer Iteración	46
Tabla 13. Tabla de Frecuencias de Edad – Segunda Iteración	47
Tabla 14. Tabla de Frecuencias de Género – Segunda Iteración.....	47
Tabla 15. Tabla de Frecuencias de Experiencia uso PC – Segunda Iteración	48
Tabla 16. Tabla de Frecuencias de uso semanal Internet – Segunda Iteración	49
Tabla 17. Estadísticas del factor <i>Operabilidad</i>	51
Tabla 18. Estadísticas del factor <i>Facilidad de Aprendizaje</i>	52
Tabla 19. Estadísticas del factor <i>Atractivo</i>	54
Tabla 20. Estadísticas del factor <i>Comprensibilidad</i>	55
Tabla 21. Estadísticas del factor <i>Conformidad</i>	57
Tabla 22. Estadísticas del factor <i>Exactitud</i>	58
Tabla 23. Estadísticas del factor <i>Madurez</i>	59
Tabla 24. Estadísticas del factor <i>Tolerancia a Fallas</i>	61
Tabla 25. Estadísticas del factor <i>Recuperabilidad</i>	62
Tabla 26. Estadísticas del factor <i>Tiempo de Respuesta</i>	64
Tabla 27. Estadística comparativa entre iteraciones del factor <i>Operabilidad</i>	66
Tabla 28. Prueba de muestras pareadas factor de calidad <i>Operabilidad</i>	68
Tabla 29. Estadística comparativa entre iteraciones del factor <i>Facilidad de Aprendizaje</i> . 69	
Tabla 30. Prueba de muestras pareadas factor <i>Facilidad de Aprendizaje</i>	71
Tabla 31. Estadística comparativa entre iteraciones del factor <i>Atractivo</i>	72

Tabla 32. Prueba de muestras pareadas factor <i>Atractivo</i>	74
Tabla 33. Estadística comparativa entre iteraciones del factor <i>Comprensibilidad</i>	75
Tabla 34. Prueba de muestras pareadas factor <i>Comprensibilidad</i>	77
Tabla 35. Estadística comparativa entre iteraciones del factor <i>Conformidad</i>	78
Tabla 36. Prueba de muestras pareadas factor <i>Conformidad</i>	80
Tabla 37. Estadística comparativa entre iteraciones del factor <i>Exactitud</i>	81
Tabla 38. Prueba de muestras pareadas factor <i>Exactitud</i>	83
Tabla 39. Estadística comparativa entre iteraciones del factor <i>Madurez</i>	84
Tabla 40. Prueba de muestras pareadas factor <i>Madurez</i>	86
Tabla 41. Estadística comparativa entre iteraciones del factor <i>Tolerancia a Fallas</i>	87
Tabla 42. Prueba de muestras pareadas factor <i>Tolerancia a Fallas</i>	89
Tabla 43. Estadística comparativa entre iteraciones del factor <i>Recuperabilidad</i>	90
Tabla 44. Prueba de muestras pareadas factor <i>Recuperabilidad</i>	91
Tabla 45. Estadística comparativa entre iteraciones para el factor de <i>Tiempo de Respuesta</i>	92
Tabla 46. Prueba de muestras pareadas factor <i>Tiempo de Respuesta</i>	94
Tabla 47. Interpretación del resultado de la métrica.....	97
Tabla 48. Evaluación de las características por 4 desarrolladores (1a iteración).	98
Tabla 49. Evaluación de las características por 4 desarrolladores (2a iteración).	99
Tabla 50. Riesgos medidos por la Complejidad Ciclomática (Anderson 2004).....	102
Tabla 51. Cálculo de la complejidad Ciclomática para la primera versión.....	102
Tabla 52. Cálculo de la complejidad Ciclomática para la segunda versión.....	103
Tabla 53. Relación de errores para la primera versión del sistema.	104
Tabla 54. Relación de errores para la segunda versión del sistema.	105
Tabla 55. Porcentaje navegadores web más populares (MarketShare 2010).....	106
Tabla 56. Escenarios de prueba primera Iteración.	107
Tabla 57. Escenarios de prueba segunda Iteración.	108
Tabla 58. Resultados concentrados de la prueba de hipótesis	112



CAPÍTULO 1. INTRODUCCIÓN

El software se ha vuelto un elemento indispensable para toda organización, un término común y entendido aun para aquellos ajenos al área.

En la actualidad, es común encontrar sistemas informáticos casi en cualquier lugar, robustos sistemas bancarios, sistemas contables, y administrativos. Cada uno enfocado a un dominio particular, haciendo de aquellos trabajos complejos actividades simples de realizar. Sin embargo, esto no es tan sencillo dado que un punto fundamental para la correcta aplicación de los sistemas informáticos estriba en el grado de eficiencia en el cual, el usuario puede interactuar y completar las tareas que desea de forma satisfactoria.

Sin embargo, las prácticas de desarrollo de software en muchas ocasiones no contemplan a quiénes serán los responsables de operar el sistema, es decir, el usuario final. Todo esto provocando que los sistemas sean difíciles de manejar, y en ocasiones, colapsos totales debido a un software ineficiente.

Debido a todo esto, es inminente que los sistemas de software sean sometidos a una cuidadosa revisión antes de su entrega definitiva. De esta manera, se buscaría cumplir con una meta primordial en el software: Usuarios satisfechos en sus expectativas (Castañeda, Gómez-Reynoso et al. 2009).

Para los creadores del software es indispensable contar con modelos que aseguren la calidad, los cuales miden de manera eficiente aquello que los usuarios necesitan. Dichos modelos listan un conjunto de atributos que los productos de software con alta calidad deben poseer (por ejemplo: confiabilidad, mantenibilidad, usabilidad) (Ward y Venkataraman 1999).

Tal es el caso del modelo ISO/IEC 9126, el cual define un modelo de calidad de productos de software, características de calidad, y métricas relacionadas (Zeiss y Vega 2007). Estas medidas se pueden utilizar como base para evaluar un producto

de software, cabe destacar que el tipo de métricas estará determinado según el tipo de software que se desee evaluar.



CAPÍTULO 2. REVISIÓN DE LITERATURA

2.1 Descripción de Teorías Base

2.1.1 Calidad

El ser humano ha estado ligado a la calidad desde tiempos remotos. En sus inicios el hombre ha buscado la manera de facilitar su trabajo, y en esta búsqueda ha inventado herramientas que lo han ayudado a lograr sus metas, siempre buscando que tales artefactos cumplan de la manera más adecuada sus objetivos, y con el paso del tiempo buscar mejoras continuas en estos.

La práctica de la mejora de calidad se remonta a mucho tiempo antes de Cristo. Por ejemplo, en 1760 A.C. el Rey Hammurabi mandó tallar un bloque de basalto que contenía 282 leyes que regulaban diversas actividades, la número 229 establecía (Gómez-Pantoja 1998):

“Un arquitecto que haya construido una casa que se desplome sobre sus ocupantes y les haya causado la muerte es condenado a la pena de muerte.”

La calidad por lo tanto era motivo de preocupación. Los fenicios por su parte, utilizaban un programa de acción correctiva, la calidad era seguida por inspectores y cuando ésta era deficiente simplemente se cortaba la mano del responsable (Pérez 2009).

En la edad media surgieron mercados en base al prestigio en la calidad, se popularizó la costumbre de poner marcas (las sedas de Damasco, la porcelana China, etc.) (Zarazua 2005). Debido a la naturaleza artesanal de los productos, la revisión de calidad era tarea del mismo productor.

Zarazua (2005) explica que con la revolución industrial el proceso de producción en serie empezó a tomar auge, la idea de calidad toma más importancia y por ello la inspección de calidad llega a formar parte esencial del proceso de producción, la cual era realizada por el mismo operario (señalaba los productos que no se

apegaban a los estándares deseados). Durante las primeras décadas del siglo XX, el objetivo primordial fue la producción en masa. El cambio, en el proceso de producto trajo cambios en la organización de la empresa. En consecuencia, fue necesario introducir en las fabricas procesos específicos para atender la calidad de los productos.

El control de calidad se practica desde hace bastante tiempo. Sin embargo, fueron los japoneses quienes se dieron cuenta que para ser competitivos en el mercado debían producir y vender mejores productos que sus competidores internacionales, tal como los Estados Unidos, Inglaterra, Francia y Alemania (Clery 2009). Esto los llevó a perfeccionar el concepto de calidad desde el diseño hasta la entrega del producto al consumidor, pasando por todas las acciones, no solo las que incluyen el proceso de manufactura, sino también actividades administrativas y comerciales, en especial las que tienen que ver con el ciclo de atención al cliente incluyendo todo servicio posterior.

Por lo tanto, se puede entender que la calidad es aquella que se interese por garantizar que un producto satisfaga íntegramente las necesidades del cliente que le solicita.

El diccionario Webster la define de la siguiente manera: “El carácter esencial de algo, un carácter inherente o distinguido, el grado de excelencia” (Lewis 2009).

Para la presente investigación adoptaremos la definición propuesta por la norma ISO-8402 que establece como calidad lo siguiente (Bertoa,Troya et al. 2002).

“La totalidad de aspectos y características de un producto o servicio que tienen que ver con su habilidad para satisfacer las necesidades declaradas o implícitas”.

Dado que la calidad es un conjunto de características deseables de un producto, es importante resaltar que puede medirse directamente tal como en la manufactura de productos, en donde la calidad viene dada por la similitud del producto desarrollado y su especificación (Sommerville 2006). Sin embargo, la medición de la calidad

puede tornarse algo más complejo para otro tipo de productos, tal es el caso del software. Este no se manufactura sino que se diseña y su proceso de desarrollo es algo más creativo que mecánico. Para la presente investigación gran parte de los criterios que definen la calidad serán medidos por el usuario, ya que el éxito del sistema dependerá de su agrado o rechazo por el software.

2.1.2 Calidad en Software

La calidad del software se define de la siguiente manera: ISO/DIS 9126 (ISO/IEC 1990) “La totalidad de características de un producto de software que conlleva en su habilidad de satisfacer necesidades existentes o implícitas”

Dentro de la literatura computacional existen generalmente dos significados aceptados para el término calidad (Lewis 2009). El primero es que la calidad significa “conocer los requerimientos”. Con esta definición, para tener un producto de calidad, éstos deben ser medibles, y pueden ser conocidos o no. Dentro de esta definición, la calidad es un estado binario, esto quiere decir que un producto posee calidad o no. Los requerimientos pueden ser completos o simples, pero mientras sean medibles, puede determinarse en donde los requerimientos de calidad se han cumplido. Esta es la vista de calidad del productor, a través del cumplimiento de las especificaciones y requerimientos. Cumplir con los requerimientos se vuelve un fin en sí mismo.

Para el usuario la calidad está presente en el producto de software que hace aquello que necesita exactamente (Lewis 2009). Otra forma de explicarlo es como decir “hecho a la medida”. También debe existir una descripción del propósito del software, típicamente documentado en el “documento de especificaciones de usuario”. Los requerimientos son el documento más importante, y el sistema de calidad gira en torno a él. En adición, los atributos de calidad son descritos en las especificaciones de requerimientos del cliente. Ejemplos incluyen a la usabilidad, la facilidad con la que el usuario se comunica con el sistema; portabilidad, la capacidad del sistema de ser ejecutado a través de un diverso rango de arquitectura

de hardware; y reusabilidad, la habilidad de transformar componentes de software contruidos en un sistema hacia otro.

La calidad del software es algo complejo que no puede medirse con una fórmula mágica, ni se puede establecer en una escala de 0 a 10. La calidad del software es algo mucho más complicado, es el trabajo organizado y estructurado, orientado a la petición de cambio del cliente. Por lo tanto, la calidad del software por sí misma no puede medirse. Sin embargo, los factores que afectan a la calidad del software si se pueden identificar y mejorar a través de una medición ya sea directa o indirecta.

De acuerdo a Somerville (2006), los factores que determinan la calidad del software se dividen en 2 grandes grupos.

- Los que se miden directamente (los que se descubren durante la prueba).
- Los que se miden indirectamente (usabilidad, mantenimiento).

Los factores que se miden directamente están asociados a la fase de desarrollo, es decir, pueden medirse aun si el software no ha sido entregado al usuario, algunos ejemplos de esto son errores de programación que se descubren durante las pruebas del software antes de su liberación. La confiabilidad puede ser medida al observar el número de fallas ocurridas en un periodo específico de tiempo, durante las pruebas de software (Nigel 2004).

Sin embargo, existen factores cuya forma de medición resulta ser indirecta y de mayor complejidad, tal como la usabilidad en donde interviene el usuario y su percepción del sistema: En este tipo de casos dichos factores deben ser evaluados por el usuario final, para lograrlo existen diferentes métodos que serán utilizados según el tipo de usuario y sistema a medir.

Literatura previa (Somerville 2006) asegura que la calidad puede lograrse definiendo estándares organizacionales que comprueba si estos son seguidos por el equipo de desarrollo. El argumento es que dichos estándares deben encapsular las buenas prácticas las cuales llevan inevitablemente a productos de alta calidad. En la

práctica, sin embargo, es más importante la calidad misma que el uso de estándares.

2.1.3 Calidad del proceso y del producto.

La calidad del software puede referirse a la calidad del producto de software o a la calidad en el proceso de desarrollo (Sivamuni y Rengaramanujam 2008).

Los estándares de la Organización Internacional para la Estandarización (ISO) se han establecido tanto para la calidad en el proceso de desarrollo, como en la calidad en el producto de software. Por ejemplo, el ISO/IEC JTC1 ISO9126 (ISO 2001) habla sobre la calidad del producto de software, mientras que el ISO/IEC JTC1, ISO-15504-5 (ISO 2004) tratan sobre la calidad en el proceso de desarrollo de software.

Hay un vínculo claro entre la calidad del proceso y del producto en producción debido a que el proceso es relativamente fácil de estandarizar y monitorear. Cada sistema de producción es calibrado, y debe producir productos de alta calidad. Sin embargo, el desarrollo de software es creativo y no mecánico; en él la experiencia y habilidad individuales son importantes (Sommerville 2006).

En el desarrollo de software, por lo tanto, la relación entre calidad de proceso y de producto es muy compleja. Los atributos de calidad del software tal como la mantenibilidad son difíciles de medir aun luego de usar el software por un largo periodo de tiempo. En consecuencia, es difícil explicar cómo influyen las características del proceso en estos atributos. Además, debido al papel del diseño y creatividad en el proceso de software, no se puede predecir la influencia de los cambios en el proceso para la calidad del producto (Sommerville 2006).

Una profunda suposición de la gestión de calidad es que la calidad del proceso de desarrollo afecta directamente en la calidad del producto derivado (Sommerville 2006). La calidad se centra en identificar productos que cumplan con ciertos criterios, y posteriormente, analizar el proceso desarrollado para crear tales productos con el fin de aplicarlo posteriormente a otros proyectos. Sin embargo, tal

relación entre calidad y proceso de desarrollo es sumamente compleja en materia de software. Los cambios llevados a cabo en el proceso de desarrollo no siempre significan mejoras en el producto.

La calidad en el proceso de desarrollo de software se logra con el establecimiento de buenas prácticas que indican técnicas y procedimientos necesarios (documentación de código, llenado de formatos, etc.) que ayudan a la estandarización del proceso de desarrollo.

La calidad del producto se centra en el software mismo y no en el seguimiento de meticulosos pasos para estandarizar su desarrollo. La calidad del producto se logra con el establecimiento de criterios de calidad y su medición. La calidad existe cuando las mediciones realizadas muestren valores significativos que aseguren la calidad en el producto, esta investigación se centra en la calidad del producto de software, no de su proceso.

2.1.4 Aseguramiento de Calidad

El aseguramiento de calidad es una aproximación estructurada a la evaluación y estimación de la calidad en el software (Khan, Mustafa et al. 2006). Se asegura que estándares y procedimientos previamente definidos sean llevados a cabo durante el transcurso del ciclo de vida de desarrollo a través de monitoreo y auditorías. La relación entre aseguramiento de calidad, control de calidad, auditoría y pruebas de software es frecuentemente confundida (Lewis 2009).

El aseguramiento de calidad es el conjunto de actividades de apoyo para proporcionar la seguridad necesaria de que los procesos están establecidos y continuamente mejorados para asegurar productos que cumplan con las especificaciones y son aptos para uso (Lewis 2009). Control de calidad es el proceso a través del cual la calidad del producto es comparada con estándares aplicados y se toma acción al detectar una inconformidad. La auditoría es una inspección que verifica la conformidad con los planes, políticas y procedimientos.

La palabra “aseguramiento” significa que si los procesos son seguidos la gestión puede asegurarse de la calidad del producto (Lewis 2009). Los objetivos de la calidad del software son típicamente alcanzados al seguir un plan de aseguramiento de calidad que establezca los métodos que el proyecto empleará para asegurarse que los documentos o productos revisados en cada “milestone” (evento importante en un proyecto) poseen una alta calidad. Ello asegura que todos los pasos han sido tomados para alcanzar la calidad del software y proporciona la administración con documentación para tales acciones. El aseguramiento de calidad de software existe debido a que ésta generalmente es cara, y debe ser incorporada en el manejo de riesgos del proyecto.

Los estándares de calidad son los que permiten definir un conjunto de criterios de desarrollo que guían la forma en que se aplica la ingeniería del software (Piattini y García 2003). 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.

De acuerdo a Castañeda (2009) para lograr el aseguramiento de calidad, no sólo es necesario hacer uso de estándares establecidos, sino también de modelos de calidad, o bien, la utilización de un nuevo modelo que combine las características de ambos.

2.1.5 Modelos de Calidad

Los modelos de calidad de software, se definen de la siguiente manera (Sivamuni y Rengaramanujam 2008): Un conjunto de características y sub-características, así como la relación entre ellos que provee las bases para especificar requerimientos de calidad y evaluación de calidad.

Asimismo, un modelo de calidad es una buena herramienta para evaluar la calidad de un producto de software. Firesmith (2005) lo define de una manera más concisa:

Es un modelo jerárquico (colección por capas de abstracciones relacionadas o simplificadas) para formalizar el concepto de calidad de un sistema en términos de:

- *Factores de calidad.*- Atributos de un sistema que capturan los mejores aspectos de su calidad (usabilidad, interoperabilidad, seguridad, confianza, etc).
- *Subfactores de Calidad.*- Componente mayor de un factor de calidad que captura un aspecto subordinado de la calidad de un sistema (respuesta, tiempo, etc.).
- *Criterios de Calidad.*- Descripción específica de un sistema que proporciona evidencia a favor o en contra de la existencia de un factor específico de calidad.
- *Medidas de Calidad.*- Medidas que califican los atributos de calidad y los hacen medibles, objetivos e inambiguos.

Los modelos de Factores-Criterios-Métricas (FCM) se han aceptado como base para las estimaciones del software (Khan, Mustafa et al. 2006). El principio básico de estos modelos es que cada atributo puede descomponerse en un conjunto de factores, los cuales pueden ser descompuestos, a su vez, en una serie de criterios. Estos valores de criterios pueden ser evaluados por un conjunto de mediciones relacionadas al software, es decir métricas de software. Los modelos FCM más conocidos son el modelo McCall (McCall, Richards, et al. 1977) y el Boehm (Boehm, Brown, et al. 1976), ambos fueron presentados a finales de los años 70s. Estos modelos son frecuentemente criticados por la falta en determinación de cuales factores deben incluirse en la definición de calidad y que criterios deben relacionarse con factores específicos. En años recientes el modelo ISO/IEC 9126 se propuso, el cual es una derivación del modelo McCall, y sobre los cuales se centra la presente investigación.

2.1.5.1 El Modelo de McCall

Uno de los primeros modelos de calidad fue propuesto por McCall Richards, y Walters en 1977. Este modelo describe la calidad como nacida de una relación jerárquica entre los factores de calidad, criterios de calidad, y métricas de calidad.

Se describió una aproximación sistemática para cuantificar la calidad como se muestra a continuación (Firesmith 2005):

- 1.- Determinar los factores que tendrán un efecto en la calidad del software.
- 2.- Identificar los criterios para juzgar cada factor.
- 3.- Definir métricas para cada criterio y establecer una función de normalización que defina la relación entre las métricas de todos los criterios pertenecientes a cada factor.
- 4.- Evaluar las métricas.
- 5.- Correlacionar las métricas con un conjunto de líneas guía que todo equipo de desarrollo pueda seguir.
- 6.- Desarrollar recomendaciones para la colección de métricas.

El modelo de McCall propone una clasificación útil de los factores que afectan la calidad los cuales se concentran en tres puntos clave (ver Figura 1).

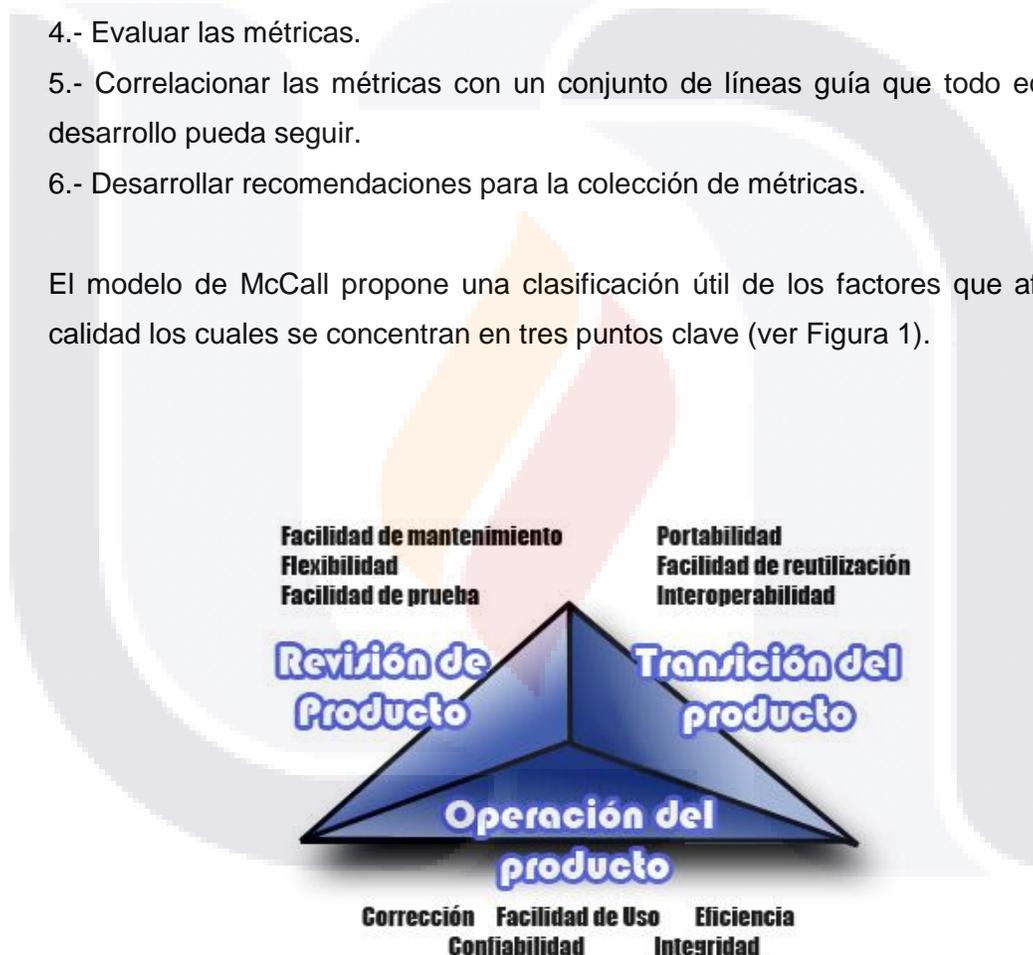


Figura 1. Modelo de Calidad de McCall (adaptado de Pressman 2005).

El modelo establece tres áreas principales que intervienen en la calidad del

software (Calero y Coral 2005):

- *Revisión de la calidad del producto de software.* Tiene como objetivo realizar revisiones durante el proceso de desarrollo para detectar los errores que afecten a la operación del producto.
- *Transición del producto.* Requiere de la definición de estándares y procedimientos que sirvan como base para el desarrollo del software.
- *Calidad en la operación del producto.* Requiere que el software pueda ser entendido fácilmente, que opere eficientemente y que los resultados obtenidos sean los requeridos inicialmente por el usuario.

A continuación se muestra una tabla con los factores propuestos en el modelo McCall (ver Tabla 1):

Características	Sub-Características	Explicación
Operación del Producto	Corrección	¿Hace lo que el usuario quiere?
	Confiabilidad	¿Es seguro?
	Facilidad de uso	¿Es fácil de utilizar?
	Eficiencia	¿El comportamiento es el adecuado?
	Integridad	¿Lo que el producto hace lo hace de forma fiable?
Revisión del producto	Facilidad de mantenimiento	¿Es tolerante a corrección de errores?
	Flexibilidad	¿Se puede cambiar?
	Facilidad de prueba	¿Se puede probar?
Transición del producto	Portabilidad	¿Puede ser utilizado en diferentes ambientes?
	Facilidad de reutilización	¿Se puede volver a utilizar algún apartado?
	Interoperabilidad	¿Puede interactuar con otros sistemas?

Tabla 1. Factores de calidad del modelo de McCall (adaptado de Pressman 2005).

Pese a que el modelo de calidad de McCall fue concebido a finales de los años 70s, sigue siendo en la actualidad un marco de trabajo reconocido para el establecimiento de estándares de calidad en materia de software. Se ha utilizado

como base para la creación de otros modelos como el caso del ISO/IEC 9126, y el modelo de Boehm, por lo cual destaca la importancia de dicho modelo en la presente investigación.

2.1.5.2 Modelo ISO/IEC 9126

El Modelo ISO/IEC 9126 es un estándar para la evaluación de software, y es uno de los más conocidos en la comunidad de software. Propone modelos como los artefactos que dan seguimiento a la aseguración de los factores de calidad que son de interés particular (Carvallo y Franch 2006).

El primer segmento de ISO/IEC 9126 contiene un modelo de dos partes: una fracción es aplicable para modelar la calidad externa e interna del producto de software, mientras que la otra intenta modelar la calidad en uso del producto de software. Típicamente la calidad interna del software se obtiene por medio de la revisión de documentos de especificación, modelos de revisión, o por análisis de código fuente. La calidad externa se refiere a las cualidades del software que interactúan con su entorno. En contraste, la calidad en uso se refiere a la calidad percibida por un usuario final que utiliza el software bajo un contexto específico (Zeiss y Vega 2007). De tal manera, las métricas internas pueden utilizarse para predecir la calidad del producto final.

Para modelar la calidad interna y externa el ISO/IEC 9126 define el mismo marco. Entonces puede ser instanciado como un modelo para calidad interna o calidad externa utilizando para ello diferentes métricas (Zeiss y Vega 2007). El estándar está basado en seis características: Funcionalidad, Confiabilidad, Usabilidad, Eficiencia, Mantenibilidad, y Portabilidad (ver Tabla 2).

Características	Sub-Características	Explicación
Funcionalidad	Conformidad	¿El software realiza las tareas deseadas?
	Exactitud	¿Es el resultado esperado?
	Interoperabilidad	¿Puede interactuar con otros sistemas?
	Seguridad	¿Prevé acceso no autorizado?
Confiabilidad	Madurez	¿La mayoría de los defectos se eliminaron en tiempo?
	Tolerancia a Fallas	¿Puede manejar errores?
	Recuperabilidad	¿Puede continuar con el trabajo y restablecer los datos después de una falla?
Usabilidad	Comprensibilidad	¿El usuario sabe cómo utilizar el sistema sin complicaciones?
	Facilidad de Aprendizaje	¿El usuario aprende fácilmente a utilizar el software?
	Operabilidad	¿El usuario utiliza el sistema sin mucho esfuerzo?
	Atractivo	¿La interface luce bien?
Eficiencia	Tiempo de Respuesta	¿Qué tan rápido responde el sistema?
	Utilización de recursos	¿Utiliza los recursos eficientemente?
Mantenibilidad	Análisis	¿Las fallas pueden ser diagnosticadas fácilmente?
	Cambio	¿Puede ser fácilmente modificado?
	Estabilidad	¿Continúa funcionando si algún cambio es realizado?
	Pruebas	¿Es fácil de probar?
Portabilidad	Adaptabilidad	¿Puede moverse a otros entornos?
	Instalación	¿Es fácil de instalar?
	Conformidad	¿Cumple con estándares de portabilidad?
	Reemplazabilidad	¿Puede reemplazar sin complicaciones a otro software?

Tabla 2. Características modelo ISO/IEC 9126-1(ISO 2001).

Abran, Khelifi, Suryn y Seffah (2003) argumentan que aunque se pensó, no es exhaustivo, estas series constituyen el modelo de calidad de software más extenso creado hasta la fecha, es también un modelo simple de utilizar para aquellos no considerados como especialistas.

Éste estándar incluye la visión del usuario e introduce el concepto de calidad en uso. La presente investigación se centra en el desarrollo de software con calidad, y

por tanto la importancia del usuario toma un papel substancial para la elaboración de software exitoso, por ello el modelo de calidad ISO/IEC 9126 es base para este trabajo de tesis. El ISO 9126-1 (ISO 2001) define la usabilidad como: La capacidad del producto de software de ser entendido, aprendido y gustado por el usuario al ser usado bajo condiciones específicas.

Las especificaciones de usabilidad y su evaluación deben contemplar varios entornos del usuario, donde el software pueda afectar, incluyendo tanto preparación de uso y resultados de evaluación (Abran, Khelifi et al. 2003).

Para especificar la calidad del software, el cliente necesita un modelo y herramientas analíticas para comunicar de forma precisa sus requerimientos concernientes al producto a desarrollar (Abran, Khelifi et al. 2003). De manera similar, un desarrollador de software necesita verificar con confianza en donde el producto proporciona los niveles de calidad deseados o no. El modelo ISO/IEC 9126 puede ser utilizado como referencia para acuerdos entre el cliente y el desarrollador de software, y se puede utilizar para eliminar malentendidos.

La principal ventaja de un modelo de calidad establecido, es que clarifica la idea de los factores y propone mediciones para proporcionar evidencia objetiva de su logro. El modelo ISO/IEC 9126 puede ser utilizado para especificar y verificar las propiedades que el software debe exhibir antes de ser puesto en servicio.

Es importante poder relacionar las mediciones del software al seguimiento de proyectos y enfocar valores al momento de la entrega del software. Además, el modelo no proporciona ninguna guía para el uso de mediciones y atributos en la identificación y clasificación de los riesgos (Hyatt y Rosenberg 1996).

2.1.5.4 Relación entre modelos

Es necesario enfatizar que entre ambos modelos descritos, existe una clara relación en cuanto a factores se refiere. Si bien el modelo McCall fue definido a finales de los 70s, es un marco de referencia que aun en la actualidad es utilizado de manera

amplia debido a la flexibilidad que provee a los desarrolladores, y es que como modelo base posibilita a los desarrolladores de software a utilizarlo según sea conveniente en una variedad de situaciones diversas. Este modelo es tan importante que ha sido base para la creación de otros modelos de calidad, tales como el de Boehm (Boehm, Brown et al. 1976), y el SQM (Murine 1988).

El modelo ISO/IEC 9126 es de reciente creación en comparación con el modelo de McCall (McCall, Richards, et al. 1977) y de mayor uso en cuanto a proyectos de calidad de software, sin embargo, es notorio destacar que éste también tomó como base de su creación al modelo de McCall. Por ello, ambos son de gran importancia para esta investigación. Para el desarrollo de la investigación se realizó un cruce de integración de los atributos de cada modelo, identificando cuáles de ellos podrían evaluarse desde el punto de vista de los usuarios finales y cuáles no. A continuación se muestra la relación que existe entre ambos modelos (ver figura 2):

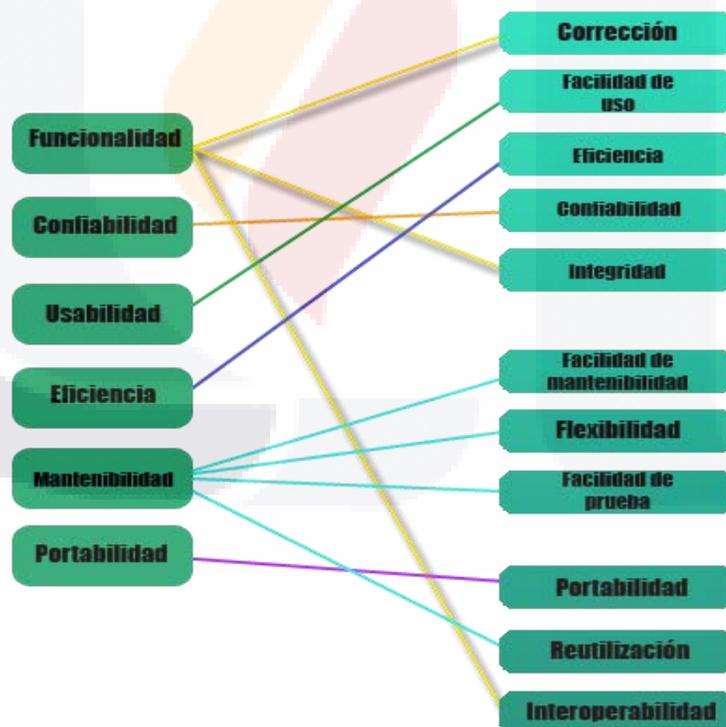


Figura 2. Equivalencias McCall – ISO 9126 (adaptado de Castañeda, Gómez-Reynoso et al. 2009).

2.1.5.5 Factores de Calidad

Los factores de calidad han sido utilizados en literatura desde antiguos modelos jerárquicos de calidad tal como el propuesto por Boehm (Boehm, Brown et al. 1976). Los estándares recomiendan una serie de factores como usabilidad, portabilidad, funcionalidad entre otros. Las personas tienden a resistirse a evaluar planes compuestos de una gran cantidad de factores, debido a recursos limitados u horarios muy apretados.

Cavano y McCall (1978) realizaron una vista de calidad del producto de software como si se tuvieran tres dimensiones: operación del producto, transición del producto y revisión del producto. El primer factor, operación de producto, trata factores de calidad como correctividad, confiabilidad y eficiencia. La transición del producto trata factores de calidad como portabilidad e interoperabilidad. La revisión del producto concierne a los aspectos relacionados a modificación de software, incluyendo factores como mantenibilidad y portabilidad.

En este trabajo se define una totalidad de 6 factores, estos son: Funcionalidad, Confiabilidad, Usabilidad, Eficiencia, Mantenibilidad, y Portabilidad. Estos han sido tomados en base a los modelos McCall e ISO 9126 debido a las siguientes consideraciones:

- Es necesario que el software haga lo que se supone debe hacer.
- Debe hacerlo de manera correcta.
- El usuario debe utilizar el sistema sin complicaciones.
- Debe ser fácil de aprender.
- Debe ser confiable.

La forma de evaluar los factores antes mencionados será por medio del usuario final y de desarrolladores. Para medir los factores con ayuda del usuario se utiliza una herramienta de medición que consta de una serie de preguntas concernientes a los factores a utilizar, esta herramienta ha sido previamente validada. La Tabla 3 muestra cómo se mide cada uno de los sub atributos de los factores a evaluar.

Ejes McCall	Atributo	Sub-atributo	Cómo se Mide
Operación del Producto	Confiabilidad	Madurez	Usuario
		Tolerancia a fallas	
		Recuperabilidad	
	Usabilidad	Comprensibilidad	Usuario
		Facilidad de aprendizaje	
		Operabilidad	
		Atractivo	
Eficiencia	Tiempo de respuesta	Usuario	
Funcionalidad	Conformidad	Usuario	
	Exactitud		
Transición del Producto	Portabilidad	Adaptabilidad	Desarrollador
		Conformidad	
	Mantenibilidad	Facilidad de análisis	Desarrollador
Facilidad de cambio			
Facilidad de prueba			
Estabilidad			
Revisión del Producto			

Tabla 3. Factores de calidad en que se basa la presente investigación.

2.1.6 Métricas de software

La meta primordial de la ingeniería de software es producir un sistema, aplicación o producto de alta calidad, dentro de un marco temporal que satisfaga una necesidad. El logro de esta meta requiere que los ingenieros apliquen métodos eficaces junto con herramientas modernas en el contexto de un proceso de software maduro (Pressman 2005). Además un buen ingeniero de software debe ser capaz de medir si se logró la alta calidad. Las métricas derivadas de la medición del software, proporcionan un inicio de la efectividad de la garantía de la calidad del software.

La medición de software se refiere a derivar un valor numérico desde algún atributo del software o del proceso de software (Sommerville 2006). Esta permite obtener una visión del proyecto y del proceso pues proporciona un mecanismo para lograr una evaluación objetiva (Pressman 2005).

Según Sommerville (2006), las mediciones del software pueden utilizarse para:

- Hacer predicciones generales del sistema. Al hacer mediciones de los componentes del sistema, reuniendo sus características, se puede entonces derivar una estimación general de algunos atributos.

- Identificar componentes anómalos. Mediante las mediciones se puede identificar los componentes anormales.

Las métricas de software han sido una investigación por más de dos décadas. En los años 1970s comenzó la era para las métricas del software, apareciendo líneas de código, y puntos de función para medir el tamaño y funcionalidad de un producto de software (Khan, Mustafa et al. 2006).

Una métrica de software es cualquier tipo de medida relacionada con un sistema, proceso o documentación de software (Sommerville 2006). Adicionalmente, estas son herramientas de medición así como son un intento por cuantificar algunos aspectos del producto generado durante un proyecto de software (Khan, Mustafa et al. 2006). Generalmente caracterizan los procesos de la ingeniería de software, como las actividades de análisis, diseño y programación.

Elijou (1991) define un conjunto de atributos para obtener métricas efectivas de software, la métrica obtenida debe cumplir con las siguientes características:

- *Simples y calculables.* Deben ser fáciles de aprender a derivar, y su cálculo no debe exigir demasiado esfuerzo.
- *Empírica.* Deben satisfacer las nociones intuitivas del ingeniero acerca del atributo del producto que se está construyendo.
- *Consistentes y objetivas.* Siempre deben arrojar resultados que no permitan ambigüedad.
- *Independientes del lenguaje.* Deben basarse en el modelo de análisis o diseño, o en la estructura del programa y nunca en el lenguaje de programación.
- *Mecanismos efectivos para la retroalimentación de alta calidad.* La métrica debe llevar a un producto final de la más alta calidad.

Algunas métricas son básicamente funciones que se definen en términos de una fórmula, pero otras son simplemente datos contables. Las métricas contables se pueden extraer de datos “raw” (datos crudos) recolectados de varios sitios, tales

como archivos de registros, observaciones en video, entrevistas o estudios detallados. Ejemplos de métricas contables incluyen el porcentaje de una tarea completa, la proporción de éxitos de tareas a las fallas, la frecuencia de uso del apartado ayuda, el tiempo utilizado en lidiar con errores del sistema, y el número de elementos en pantalla (Seffah, Donyaee et al. 2006).

Las métricas calculables son el resultado de cálculos matemáticos, algoritmos, o heurísticas basadas en datos observacionales raw de métricas contables. Por ejemplo, una fórmula propuesta por Bevan y MacCleod (1994) para calcular la efectividad es:

$$TE = Cantidad \times Calidad / 100$$

Donde cantidad es la proporción de las tareas completadas y calidad es la proporción de las metas logradas. La proporción mencionada son las métricas contables que crean la métrica calculable TE.

2.1.7 Garantía de Calidad

Hablando de software, se puede decir que cuando se examina su calidad por tratarse de elementos medibles se dice que se habla de dos tipos de calidad. La calidad del diseño y la calidad de concordancia (Pressman 2005).

De acuerdo a Pressman (2005) la calidad del diseño es aquella que se refiere a las características que los diseñadores especifican para un elemento, mientras que la calidad de concordancia se refiere al grado en que las especificaciones de diseño se aplican durante la fabricación.

Para la calidad de diseño en software se incluyen: requisitos, especificaciones, y el diseño del sistema. La calidad de concordancia está enfocada a la implementación. Si ésta sigue el diseño y el sistema resultante satisface sus requisitos la calidad de concordancia es alta (Pressman 2005).

Por lo anterior se puede decir que la calidad es de vital importancia en el diseño y construcción de software, por ello garantizarla debe ser una tarea exhaustiva del ingeniero de software.

De acuerdo a Sommerville (2006) la garantía de la calidad es el proceso que define cómo lograr la calidad del software y cómo la organización del desarrollo conoce el nivel requerido en el software. Se pueden seleccionar herramientas y métodos que apoyen a los estándares requeridos para el aseguramiento de calidad, en base a ello se pueden definir los siguientes:

- *Estándares del producto.* Estos estándares se aplican sobre el producto de software que se comienza a desarrollar. Incluyen estándares de documentación, como cabecera de comentarios estándar para definición de clases, y estándares de codificación, que definen cómo debe utilizarse el lenguaje de programación.
- *Estándares de Proceso.* Estos estándares definen los procesos que deben seguirse durante el desarrollo del software. Pueden incluir definiciones de procesos de especificación, diseño y validación, así como una descripción de los documentos que deben escribirse en el curso de esos procesos.

Con ello se denota que la calidad de software es de gran importancia, por ello garantizarla es una tarea de gran importancia tanto para desarrolladores así como para el usuario final, esto debido al gran crecimiento que la industria de software tiene día a día, y al impacto que representa dentro de nuestras vidas.

CAPÍTULO 3. FORMULACIÓN DEL PROBLEMA DE INVESTIGACIÓN

3.1. Descripción del problema de Investigación y su relevancia

En la actualidad, la mayoría de proyectos de software, realizan tareas para implementar planes de calidad, ello genera una forma ordenada y controlada de llevar a cabo el proceso de desarrollo de software (Castañeda, Gómez-Reynoso et al. 2009). Sin embargo, esto no determina la calidad en el producto final, debido a que tal afirmación depende del usuario final.

Los sistemas de software se desarrollan de manera intuitiva pensando en una generalización para el usuario, lo cual generalmente provoca que el desarrollador tienda a dejarlo de lado durante el análisis. Existen aspectos propios del usuario tales como estilo de aprendizaje, herramientas favoritas, e incluso diferencias culturales, los cuales pueden impactar en la eficiencia del sistema en cuestión (Macracken y Wolfe 2004). El resultado pueden ser sistemas de información difíciles de utilizar por los usuarios. Algunos de ellos contienen textos técnicos que el usuario desconoce, en otros simplemente no se lograrán las metas que se deseaba realizar al utilizar dicho sistema.

Algunos problemas que acarrea el no realizar un diseño centrado en el usuario pueden ser los siguientes (Macracken y Wolfe 2004):

- Pérdida significativa de tiempo en desarrollo, incluyendo en el sistema apartado que el usuario nunca utilizará porqué desconoce que existen o simplemente no sabe cómo hacer uso de ellos.
- Altos costos en diseño y mantenimiento ya que software mal construido requerirá futuras modificaciones que significan costos extra.

- Pérdidas económicas, debido a que los cambios efectuados en los sistemas para corregir errores sobrepasan los presupuestos establecidos al inicio.

Por lo anterior, se puede decir que la calidad de un producto de software no depende únicamente del seguimiento en los procesos de desarrollo sino en el grado de satisfacción del usuario al utilizar dicho producto.

De acuerdo a Pfleeger (2002) un sistema puede tener un desempeño excelente al realizar una función, pero si los usuarios no logran entender cómo utilizarlo, el sistema es un fracaso.

Dado lo anterior, reafirmarnos la importancia del aseguramiento de la calidad. De acuerdo a Lewis (2009) de tal definición se puede decir que tales actividades y funciones de seguimiento no están especificadas, es decir varían según el tipo de proyecto, y personal que da seguimiento, sin embargo es importante destacar que en este seguimiento de calidad el usuario debe ser indispensable, propuesta que establece este trabajo de tesis, ya que la falta de calidad en el software (que determina el usuario final) acarrea varios inconvenientes.

Algunos problemas que la poca calidad en el software puede ocasionar son los siguientes (Lewis 2009):

- Fallas frecuentes en el software.
- Las consecuencias de las fallas de software son inaceptables, desde lo económico a escenarios de vida.
- Frecuentemente el software no está disponible para lo que era su propósito original.
- Las mejoras del software suelen ser muy costosas.
- El costo de detectar y corregir errores es excesivo.

Por ello se han definido modelos de calidad que sirven como referencia para conocer las cualidades necesarias para alcanzar la calidad de un producto de software. Aunque los modelos de calidad definen criterios que evalúan la calidad de

cierta forma, es necesario establecer métodos de evaluación cuantitativa que permitan obtener resultados de manera más objetiva, integrando tanto elementos técnicos a evaluar como involucrando al usuario en la determinación de la calidad del software (Castañeda, Gómez-Reynoso et al. 2009).

3.2 Pregunta de Investigación

Por lo anterior, se decide la importancia de investigar el efecto de la participación de los usuarios dentro de la calidad final de un producto de software; en base a esto, se plantea la siguiente pregunta de investigación:

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

3.3 Objetivo General

Realizar la adaptación de un método para la evaluación de un producto de software donde los usuarios finales y desarrolladores participen conjuntamente, ello en base a los factores de calidad de los modelos descritos, que permita mediante su aplicación iterada y con la utilización de técnicas estadísticas, la obtención de un producto de calidad hasta lograr el nivel ideal para su liberación a los usuarios finales.

3.3.2 Objetivos Específicos

- Determinar los requerimientos principales de calidad de un producto de software en términos de factores de calidad.

- Lograr una integración adecuada de los requerimientos generales con los requerimientos de calidad deseables en la aplicación del software.
- Establecer los criterios de evaluación de los factores de calidad que se aplicaran en cada iteración para medir en términos de calidad identificados por el usuario.
- Establecer las métricas adecuadas para la medición de los factores de calidad del lado del desarrollo del software.
- Establecer los criterios en base a los cuales se considerará que la aplicación está lista para su liberación.
- 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.
- Evaluar la calidad por parte de los desarrolladores de forma tal que se identifiquen desviaciones en el producto que deben ser corregidas por el equipo de desarrollo.

3.4 Hipótesis

En necesario hacer el planteamiento de una serie de hipótesis ello debido al tipo de investigación realizada, con ellas se puede identificar si la participación de los usuarios finales y del equipo de desarrollo 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 la 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, y desarrollador 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 Usabilidad por parte del usuario final mejora significativamente la calidad del producto de software.

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

H1b. La evaluación de la Facilidad de Aprendizaje por parte del usuario final mejora significativamente la calidad del producto de software.

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

H1d. La evaluación de la Comprensibilidad por parte del usuario final mejora significativamente la calidad del producto de software.

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

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

H2b. La evaluación de la Exactitud por parte del usuario final mejora significativamente la calidad del producto de software.

H3. La evaluación de la Confiabilidad por parte del usuario final mejora significativamente la calidad del producto de software.

H3a. La evaluación de la Madurez por parte del usuario final mejora significativamente la calidad del producto de software.

H3b. La evaluación de la Tolerancia a Fallas por parte del usuario final mejora significativamente la calidad del producto de software.

H3c. La evaluación de la Recuperabilidad a Fallas por parte del usuario final mejora significativamente la calidad del producto de software.

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

H4a. La evaluación del Comportamiento del Tiempo de Respuesta por parte del usuario final mejora significativamente la calidad del producto de software.

H5. La evaluación de la Mantenibilidad por parte del desarrollador mejora significativamente la calidad del producto de software.

H5a. La evaluación de la Facilidad de Análisis por parte del desarrollador mejora significativamente la calidad del producto de software.

H5b. La evaluación de la Facilidad de Cambio por parte del desarrollador mejora significativamente la calidad del producto de software.

H5c. La evaluación de la Facilidad de Prueba por parte del desarrollador mejora significativamente la calidad del producto de software.

H5d. La evaluación de la Estabilidad por parte del desarrollador mejora significativamente la calidad del producto de software.

H6. La evaluación de la Portabilidad por parte del desarrollador mejora significativamente la calidad del producto de software.

H6a. La evaluación de la Adaptabilidad por parte del desarrollador mejora significativamente la calidad del producto de software.

H6b. La evaluación de la Conformidad por parte del desarrollador mejora significativamente la calidad del producto de software.

CAPÍTULO 4. DESARROLLO DE LA INVESTIGACIÓN

El enfoque general de la investigación es aplicada, ya que según Padron (2006) uno de los sentidos específicos para este tipo de investigación, es donde se consideran los estudios que explotan teorías científicas previamente validadas para la solución de problemas prácticos, como la producción de bienes y servicios. En el caso de esta investigación se busca lograr la creación de productos de software con un alto grado de calidad, lo cual se visualiza como un problema práctico encajando en la definición antes mencionada.

El tipo de investigación es experimental ya que en los experimentos (Hernandez-Sampieri 1997) se analizan relaciones puras entre las variables de interés, por ello se pueden establecer relaciones causales con mayor precisión. En esta investigación se evalúa el efecto de la variable independiente *estrategia de desarrollo* sobre la variable dependiente *calidad* (factores) del software.

También se considera como un estudio explicativo (Hernandez-Sampieri 1997) ya que los estudios experimentales debido a que analizan la relación entre una o varias variables independientes y una o varias dependientes y los efectos causales de las primeras sobre las segundas son estudios explicativos.

4.1 Materiales y Métodos

En la presente investigación se realizó una adaptación al método creado en el trabajo “Desarrollo de un método de evaluación de software que mejore el producto final” (Castañeda, Gómez-Reynoso et al. 2009), en la cual se realizó una evaluación por parte de usuarios a un sistema de software, esta investigación retoma la exploración en base a los modelos ISO/IEC 9126 y McCall, y la enriquece proponiendo que factores de calidad de un producto de software deben evaluarse desde el punto de vista del usuario final y cuales deben evaluarse desde la perspectiva del desarrollador, así como la forma de medirlos y en qué términos.

Una vez que los criterios fueron definidos, se creó un instrumento de evaluación para la medición por parte del usuario y se definieron métricas para la medición por parte del desarrollador. Posteriormente, se desarrolló un sitio web tomando en cuenta las recomendaciones propuestas por McCracken (2004). Posteriormente, fue necesario definir el perfil del usuario para este sistema, y en base a dicho perfil, seleccionar a los usuarios para formar los grupos de evaluación mediante un determinado número de iteraciones, con la finalidad de lograr en cada iteración una evaluación adecuada del producto. Así mismo, se hicieron los cálculos de las métricas definidas para evaluar los factores desde la perspectiva del desarrollador, para mejorar el software en base a los resultados de la evaluación por parte del usuario y desarrollador en cada iteración y mejorar la calidad en lo subsecuente, hasta alcanzar un nivel óptimo de ésta.

Para la evaluación de los factores a analizar se utilizaron técnicas de estadística descriptiva, las cuales permitieron obtener un resultado medible y comparable de manera que se logre apreciar de forma precisa la calidad asignada por el usuario final al producto evaluado. El proceso de desarrollo de software que se utilizó para la adaptación del modelo (ver figura 3) hace énfasis en el apartado *pruebas de calidad (usuarios/desarrolladores)* con el objeto que al ser parte integral del proceso de desarrollo, la evaluación iterada de usuarios y desarrolladores permitirá la liberación de un producto final óptimo en materia de calidad.

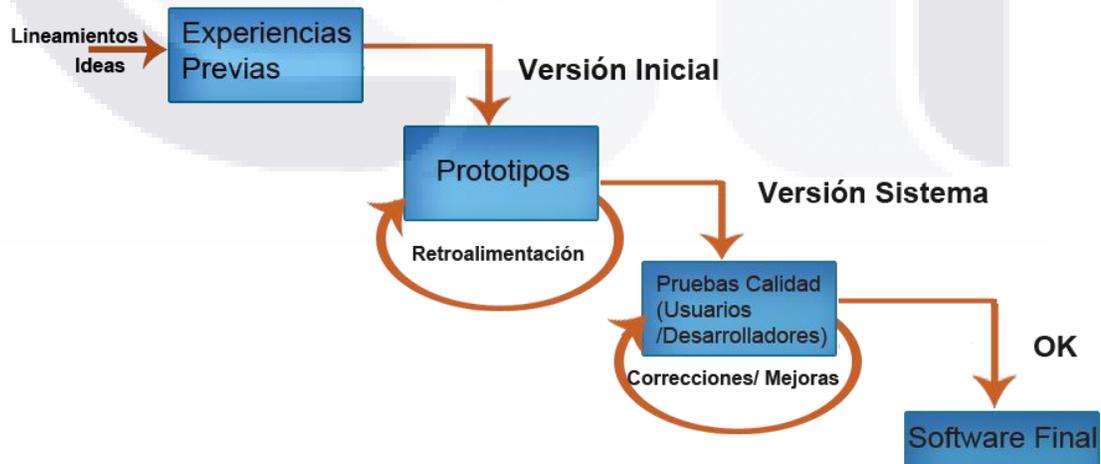


Figura 3. Modelo de Desarrollo Base (adaptado de Gomez-Reynoso y Brizuela-Sandoval 2008).

Para esta investigación se consideraron los modelos de calidad McCall e ISO/IEC 9126, por lo que se analizaron sus factores, identificando cuales pueden evaluarse desde el punto de vista del usuario final y cuales pueden medirse desde la perspectiva del desarrollador (ver Tablas 4 y 5).

Sub - atributos ISO 9126	McCall
Funcionalidad.Conformidad	OperaciónDelProducto.Corrección
Funcionalidad.Seguridad	
Funcionalidad.Cumplimiento	OperaciónDelProducto.Integridad
Funcionalidad.Interoperabilidad	TransiciónDelProducto.Interoperabilidad.
Confiabilidad.Madurez	OperaciónDelProducto.Confiabilidad
Confiabilidad.ToleranciaAFallos	
Confiabilidad.Recuperabilidad	
Usabilidad.Comprendibilidad	OperaciónDelProducto.Facilidad de Uso
Usabilidad.FacilidadDeAprendizaje	
Usabilidad.Operabilidad	
Usabilidad.Atractivo	
Eficiencia.ComportamientoEnElTiempo	OperaciónDelProducto.Eficiencia
Eficiencia.ComportamientoEnLosRecursos	
FacilidadDeMantenimiento.FacilidadDe Analisis	RevisiónDelProducto.FacilidadDeMantenimiento
FacilidadDeMantenimiento.FacilidadDe Cambio	RevisiónDelProducto.Flexibilidad
FacilidadDeMantenimiento.Estabilidad	RevisiónDelProducto.FacilidadDeMantenimiento
FacilidadDeMantenimiento.FacilidadDe Prueba	RevisiónDelProducto.FacilidadDePrueba
Portabilidad.Adaptabilidad	TransiciónDelProducto.Portabilidad
Portabilidad.CapacidadDeInstalacion	TransiciónDelProducto.Portabilidad
Portabilidad.Conformidad	TransiciónDelProducto.Portabilidad
Portabilidad.CapacidadDeReemplazamiento	TransiciónDelProducto.Portabilidad

Tabla 4. Equivalencias McCall – ISO 9126.

ATRIBUTO	SUB - ATRIBUTO	DESCRIPCIÓN
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.
Usabilidad	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.
	Atractivo	El producto de software es estéticamente agradable al usuario.
Eficiencia	Tiempo de respuesta	El producto de software cuenta con tiempos adecuados de respuesta al realizar sus diferentes funciones.
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 del cliente.
Portabilidad	Adaptabilidad	El producto de software puede adaptarse a diferentes entornos sin la necesidad de aplicar acciones específicas diferentes a las establecidas.
	Conformidad	El producto de software se adhiere a estándares o convenciones relativas a la portabilidad.
Mantenibilidad	Facilidad de análisis	El producto de software es diagnosticado por causas de fallas para identificar las partes que se deben modificar.
	Facilidad de cambio	El producto de software debe permitir modificaciones para eliminar fallas.
	Facilidad de prueba	El producto de software permite validar las modificaciones realizadas.
	Estabilidad	El producto de software evita efectos no esperados debido a modificaciones.

Tabla 5. Factores que se midieron en la investigación.

4.2 Estudio Piloto

En base a la revisión de literatura, modelos de calidad y trabajos relacionados se logró identificar cuáles son los factores de calidad que pueden ser evaluados por los usuarios así como aquellos que es más pertinente evaluarlo por parte del desarrollador.

A partir de los elementos que se identificaron como factores clave para la evaluación de la calidad por parte de los usuarios, y utilizando como base herramientas como el Cuestionario para la Satisfacción del Usuario en Interfaces (QUIS), Cuestionario de Usabilidad para Sistemas de Cómputo (CSUQ) y el Sistema de Escala de

Usabilidad (SUS) (Tullis y Stetson 2004). Entonces, se desarrolló un cuestionario consistente de 7 preguntas demográficas y 33 preguntas sobre calidad. Una vez determinadas las preguntas y escalas de respuestas posibles, se desarrolló un cuestionario en línea, que además de recabar la información necesaria, mide el tiempo de su aplicación, esto para descartar cuestionarios contestados en un lapso de tiempo no adecuado para le número de preguntas, lo cual podría indicar que el respondiente no estuviera realmente leyendo y entendiendo cada pregunta.

La encuesta se aplicó a 42 personas utilizando Internet. Se pidió a los usuarios que utilizaran un sistema web por 40 minutos y posteriormente procedieran a contestar las preguntas.

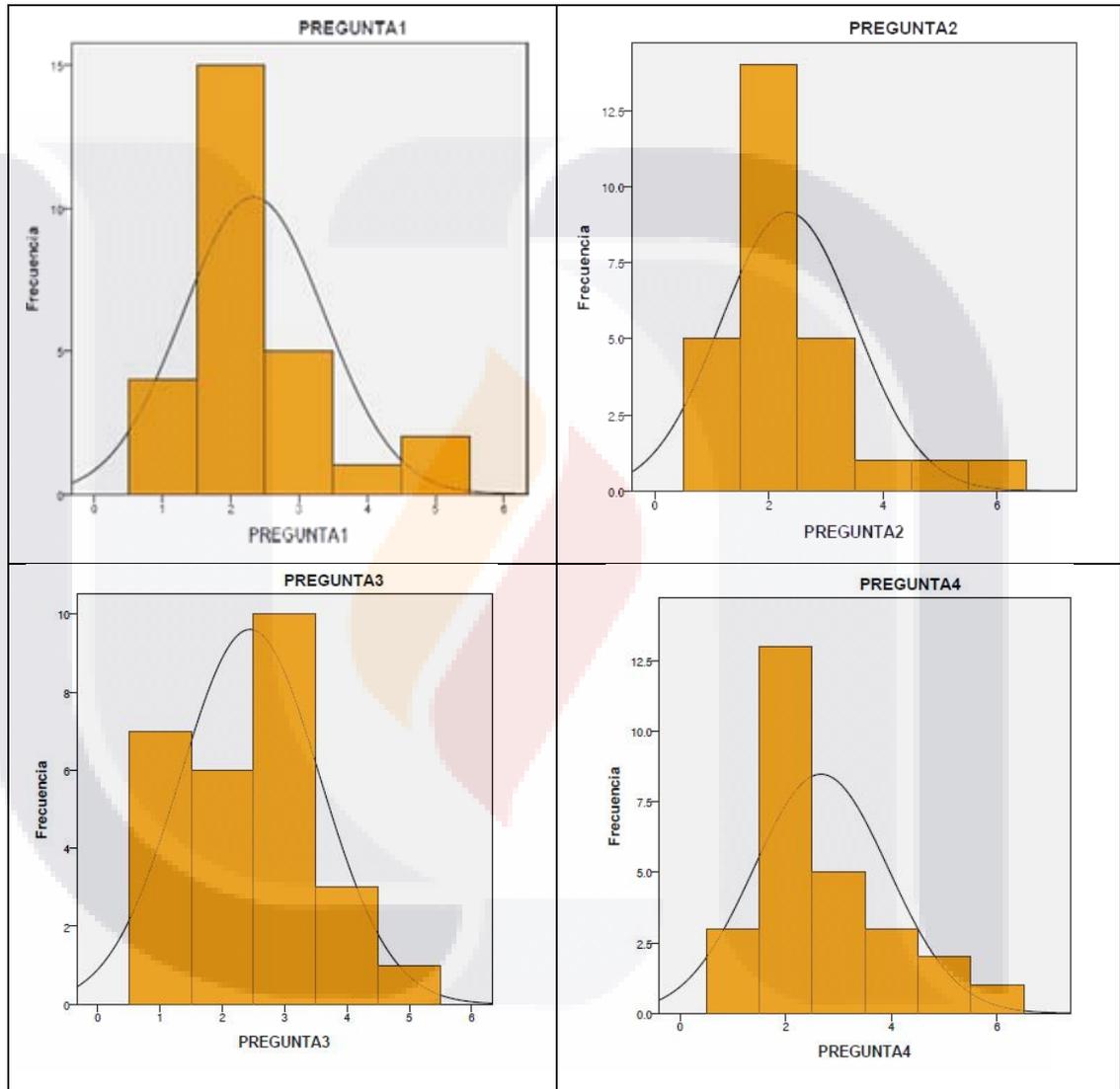
Con la medición del tiempo que se agregó a la encuesta, se descartaron aquellas que tenían un tiempo de contestación demasiado corto y fueron eliminadas del estudio piloto, ya que indicaban que el usuario no leyó con detenimiento la herramienta de evaluación.

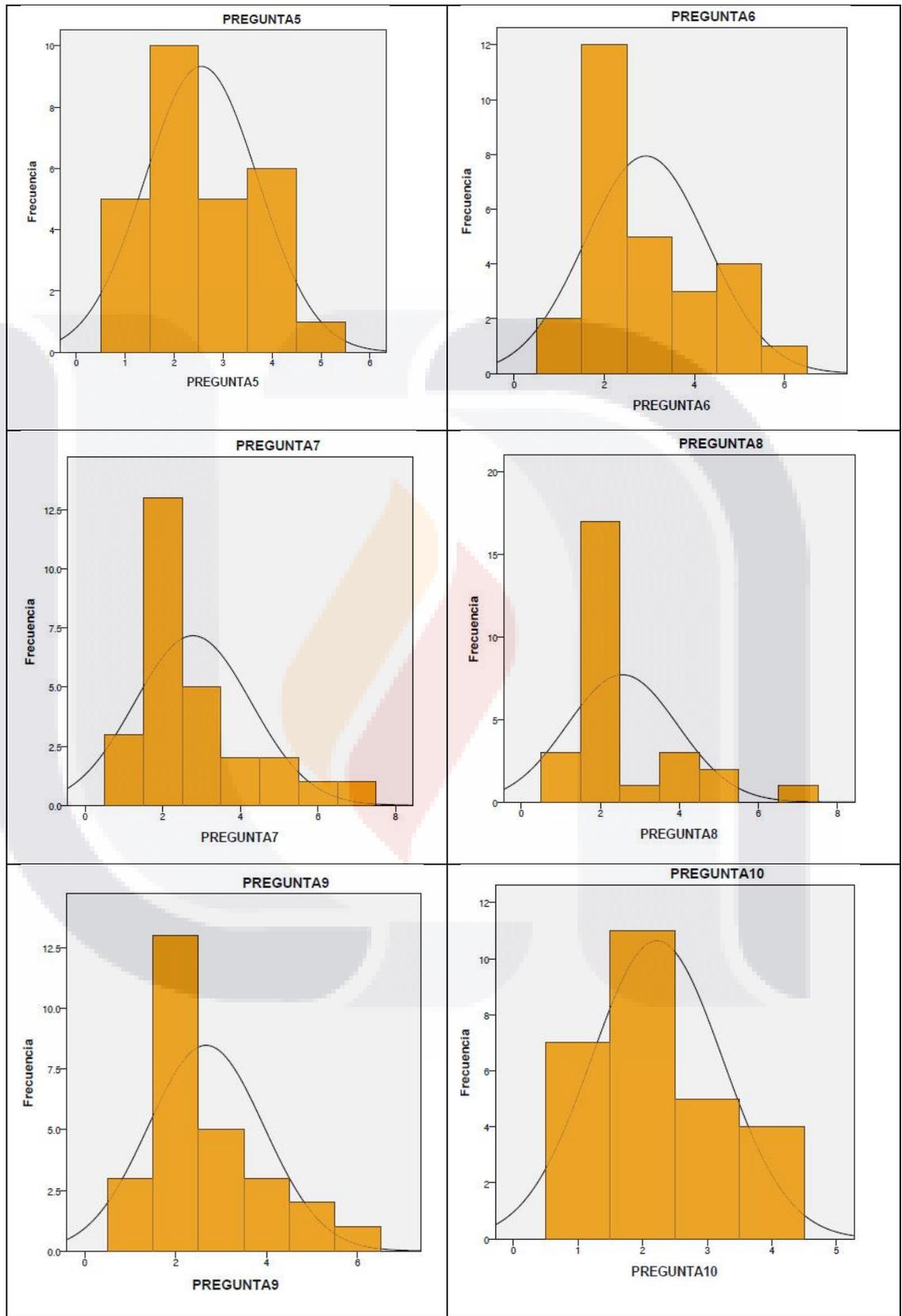
En base a los resultados obtenidos se modificaron y corrigieron problemas potenciales en el diseño de la herramienta y se volvió a repetir la medición con diferentes usuarios siguiendo la misma mecánica. Una vez obtenidos los datos para evaluar la efectividad de la herramienta se realizó la prueba de alfa de Cronbach que arrojó los siguientes resultados mostrados en la Tabla 6:

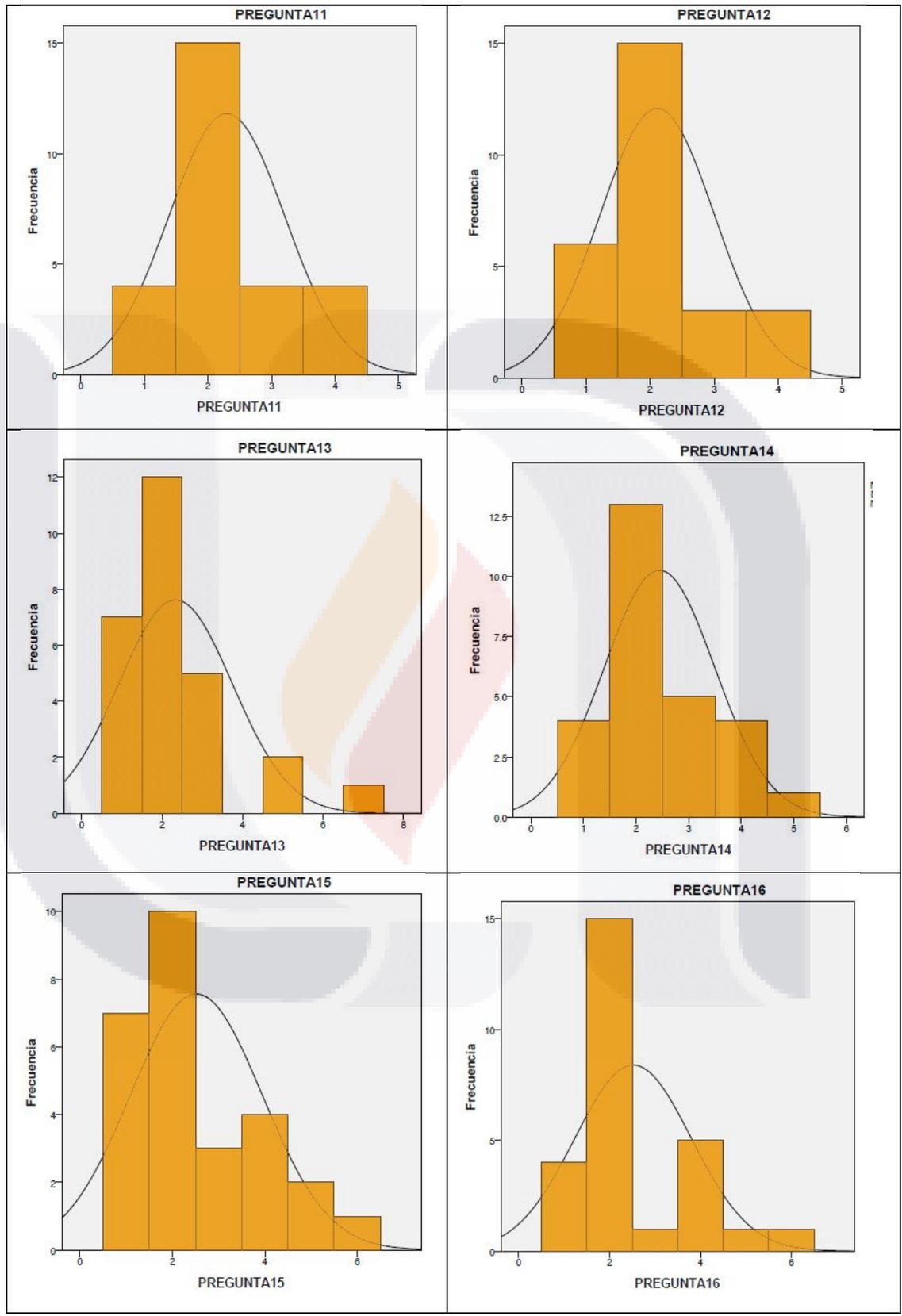
FACTOR	ALFA DE CRONBACH	MEDIA	VARIANZA
Operabilidad	0.887	2.44	1.33
Facilidad De Aprendizaje	0.847	2.70	1.34
Atractividad	0.842	2.19	1.08
Comprensibilidad	0.804	2.45	1.60
Funcionalidad	0.810	2.51	1.60
Exactitud	0.905	2.38	1.15
Madurez	0.878	3.20	1.80
Tolerancia A Fallos	0.843	3.28	1.82
Recuperabilidad	0.947	3.18	1.80
Tiempo De Respuesta	0.868	2.51	1.44

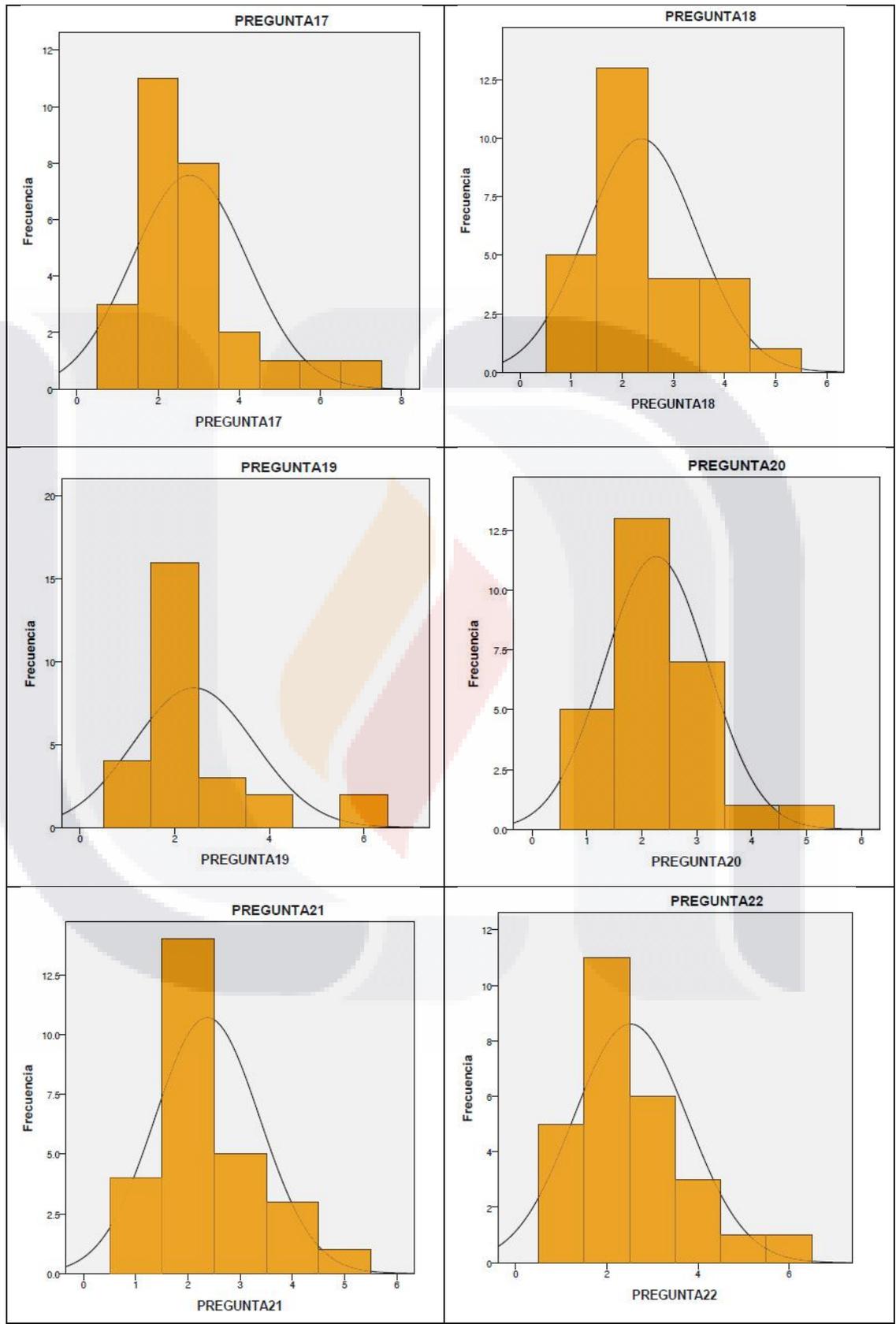
Tabla 6. Resultados la Prueba Piloto

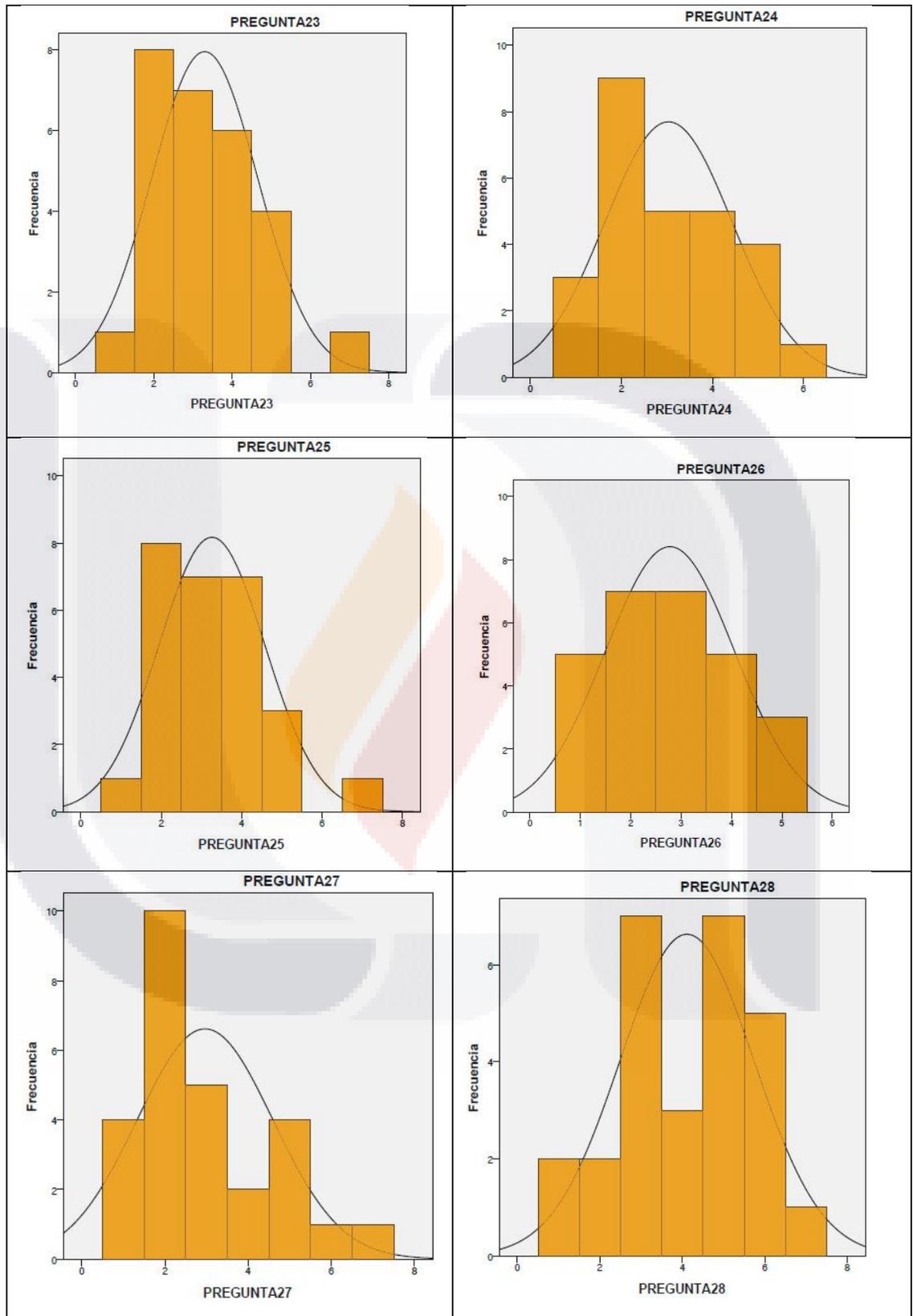
Por lo tanto, debido a que los valores de alfa de Cronbach están por arriba de 0.8; en conclusión, se puede decir que la herramienta posee una consistencia interna adecuada, además al observar el histograma de cada una de las preguntas se aprecia que se cuenta con una distribución normal, como se muestra a en la tabla 7 a continuación.











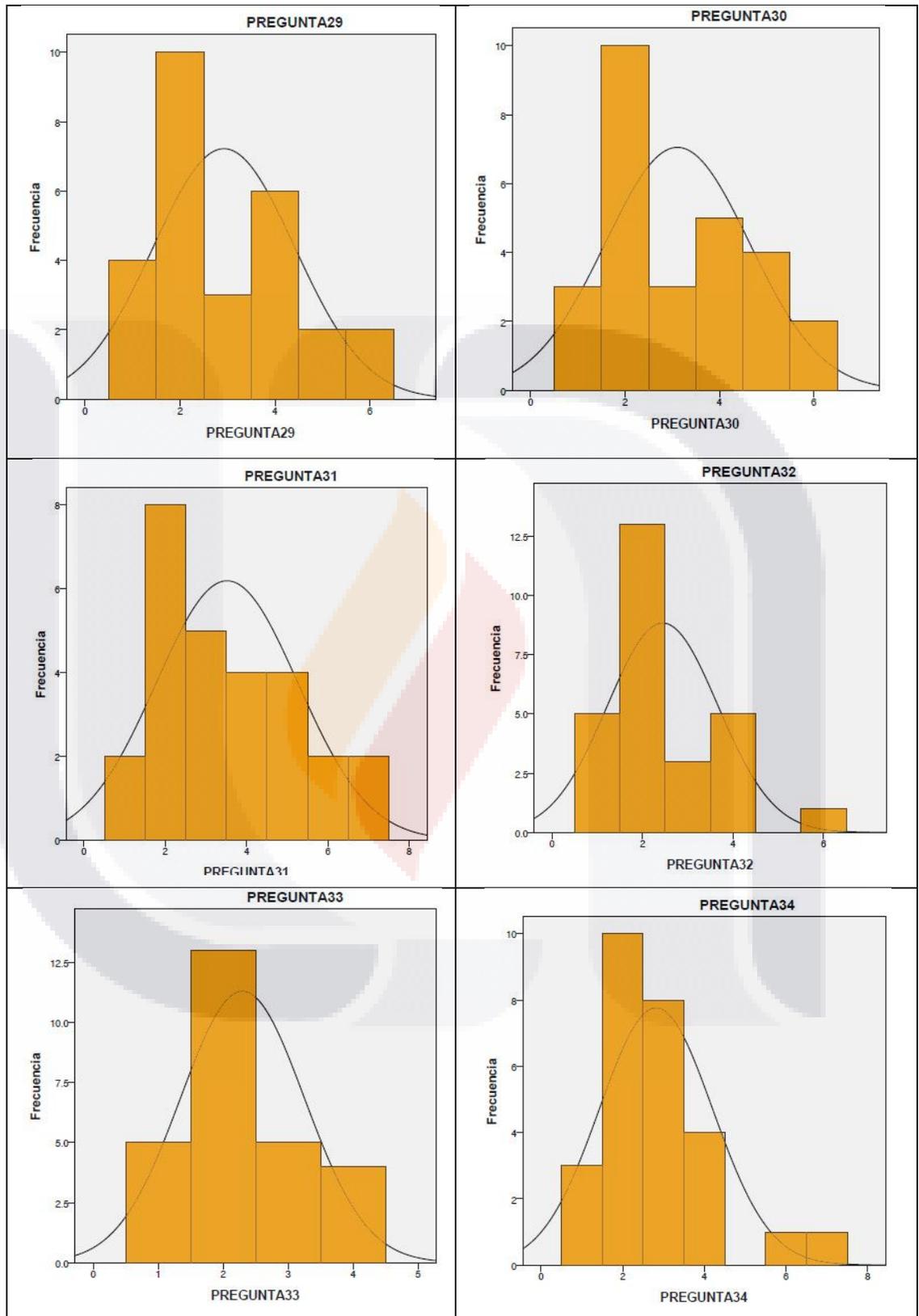


Tabla 7. Histogramas del estudio piloto.

Una vez finalizado el estudio piloto, la herramienta de evaluación quedo lista para ser utilizada en el experimento de esta investigación, la cual quedó conformada por 7 preguntas de carácter demográfico y 34 preguntas concernientes a la calidad del producto de software. Para contestar cada pregunta se utilizó una escala de Likert de 7 puntos, donde la escala se definió como 1) Calidad excelente a 7) Pésima calidad). Al final del cuestionario se colocó un apartado de comentarios que el usuario tenía como opcional agregarlos o no.

4.3 Estudio Final

4.3.1 Perfil del Usuario

Para realizar el experimento se desarrolló un sistema web de carácter turístico, y por lo tanto, orientado al público en general. Esto debido a que las características del usuario serán acorde al perfil de los Internautas en México. Entonces, las características relevantes necesarias para definir el perfil del usuario se toman en base a los siguientes datos:

- **Edad**

La población de usuarios que hacen uso de Internet en México está conformada principalmente por Jóvenes, de acuerdo con cifras de INEGI (2009), 77.3% de los Internautas Mexicanos tiene menos de 35 años; AMIPCI (2009) establece que 61% de los jóvenes entre 20 y 24 años hacen uso del ciberespacio.

- **Escolaridad**

Conforme a un estudio realizado por INEGI (2008), 7 de cada 10 jóvenes que cursan estudios de nivel superior hacen uso frecuente de Internet (ver figura 4).

Proporción de Usuarios de Internet por nivel de escolaridad, 2008

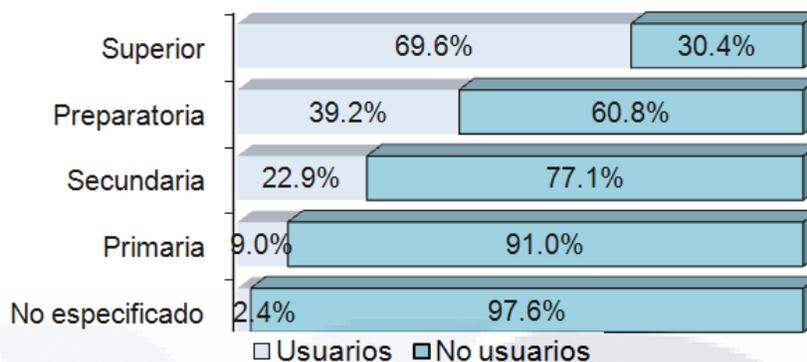


Figura 4. Encuesta Nacional sobre Disponibilidad y Uso de las Tecnologías de la Información en los Hogares (INEGI 2008).

- **Genero**

Según el sexto estudio de los hábitos de los internautas en México realizado por AMIPCI (2009), el perfil del internauta cumple con las siguientes características de género (ver figura 5).



Figura 5. Genero del perfil del Internauta según AMIPCI (2009).

4.3.2 Muestra

Por lo anterior se realizó el estudio con estudiantes de licenciatura de la Universidad Autónoma de Aguascalientes, de la carrera de Ingeniería en Sistemas Computacionales, ya que al ser estudiantes de nivel superior cumplen con el perfil indicado, además su carrera es relacionada al uso de tecnologías de la información

donde se hace mayor uso de Internet, factor clave para la investigación. Además, cumplen el requisito de edad por estar situados entre 20 a 24 años. Por ello, los grupos seleccionados de usuarios cumplen con las características antes mencionadas de forma idónea. Cabe mencionar que el género queda descartado del perfil del usuario para esta investigación ya que el grupo de usuarios no tiene una representatividad de población acorde a este dato, es decir AMIPCI establece el género para estadísticas globales, pero no indica cual es la proporción entre internautas jóvenes.

4.3.3 Recopilación de Información

La aplicación del Instrumento de evaluación se realizó en un laboratorio de cómputo con alumnos de séptimo semestre de la carrera de Ingeniería en Sistemas Computacionales de la Universidad Autónoma de Aguascalientes, la invitación a participar se dio dentro de la materia de Lenguajes de Bases de Datos.

Antes de realizar la evaluación, se explicó a los participantes la naturaleza del estudio, también se les recordó que su participación sería anónima, para ayudarlos a expresarse a la hora de hacer comentarios.

Ellos evaluaron un sistema web que se desarrolló para esta investigación particularmente, este sistema muestra información que es de interés para cualquier Internauta.

En la primera iteración se logró un total de 59 encuestas contestadas. A los usuarios utilizaron se les asignó un tiempo de 40 minutos para revisar el sitio Web y posteriormente contestar la encuesta que se encontraba en línea, para esto se les asignó un tiempo de 30 minutos.

La segunda iteración se realizó a una segunda versión del sistema, la cual se obtuvo al corregir la versión anterior teniendo como base los comentarios y sugerencias hechas por los participantes de la primera iteración. La invitación se realizó de igual forma que en la primera iteración y se logró un total de 56 cuestionarios contestados, teniendo una mortalidad de 3 usuarios.

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

4.3.3.1 Primer Iteración

En la primer iteración participaron 59 usuarios lo cuales, tienen las siguientes características demográficas.

El rango de edad de los usuarios que contestaron la evaluación fue de la siguiente manera: 9 tienen 20 años, 49 tienen una edad en el rango de 21 a 25, y 1 usuario tiene una edad en el rango 26 a 30, como se muestra en la Tabla 8 a continuación.

	Frecuencia	Porcentaje	Porcentaje Válido	Porcentaje Acumulado
20 o menos	9	15.3	15.3	15.3
21 a 25	49	83	83	98.3
26 a 30	1	1.7	1.7	100
Total	59	100	100	

Tabla 8. Tabla de Frecuencias de Edad – Primer Iteración

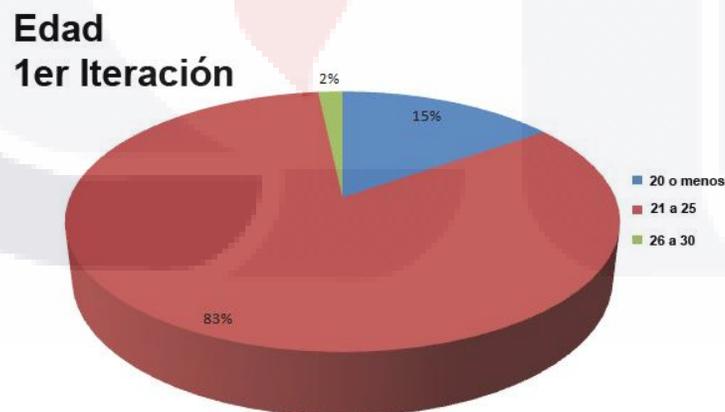


Figura 6. Edad – Primer Iteración

Del total de los usuarios que participaron en la evaluación, la mayoría es de género masculino con un 76% del total (ver tabla 9, y figura 7).

	Frecuencia	Porcentaje	Porcentaje -Válido	Porcentaje Acumulado
Masculino	45	76.3	76.3	76.3
Femenino	14	23.7	23.7	100
Total	59	100	100	

Tabla 9. Tabla de frecuencias de Género – Primer Iteración

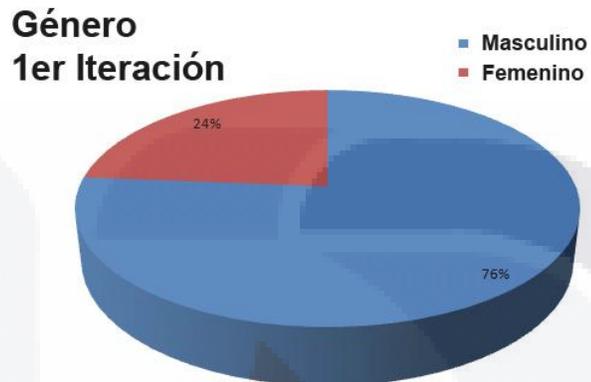


Figura 7. Género– Primer Iteración

En relación al nivel de estudios se hizo la evaluación con estudiantes de licenciatura, por ello el 100% de los usuarios tiene este nivel de estudios.

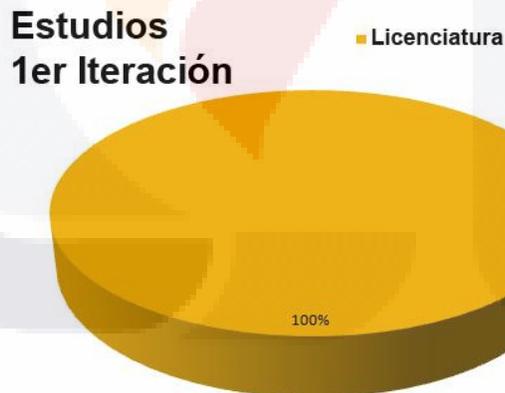


Figura 8. Nivel de Estudios – Primer Iteración

Con respecto a la experiencia con el uso de la computadora en los usuarios, la mayoría tiene una buena experiencia como se muestra a continuación (ver tabla 10 y figura 9).

	Frecuencia	Porcentaje	Porcentaje -Válido	Porcentaje Acumulado
Mucha	17	28.8	28.8	28.8
Suficiente	31	52.6	52.6	81.4
Adecuada	8	13.6	13.6	95
Regular	3	5.0	5.0	100
Poca	0	0	0	
Muy Poca	0	0	0	
Nada	0	0	0	
Total	59	100	100	

Tabla 10. Tabla de frecuencias de Experiencia uso de PC – Primer Iteración

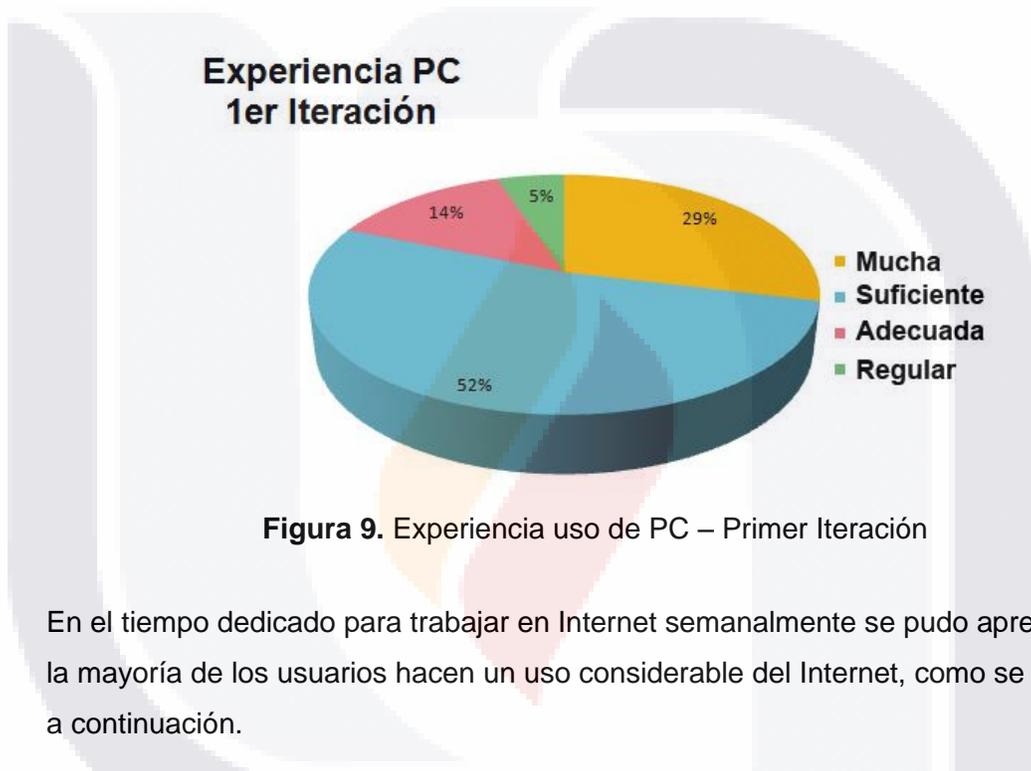


Figura 9. Experiencia uso de PC – Primer Iteración

En el tiempo dedicado para trabajar en Internet semanalmente se pudo apreciar que la mayoría de los usuarios hacen un uso considerable del Internet, como se muestra a continuación.

Rangos	Frecuencia	Porcentaje	Porcentaje -Válido	Porcentaje Acumulado
Menos 1 hr.	0	0	0	0
1 a 4 hrs.	12	20.3	20.3	20.3
4 a 10 hrs.	19	32.2	32.2	52.5
Más de 10	28	47.5	47.5	100
Total	59	100	100	

Tabla 11. Tabla de frecuencias uso semanal de Internet – Primer Iteración

Uso semanal Internet 1er Iteración

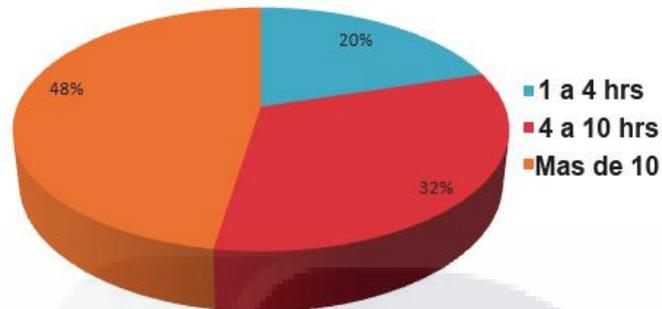


Figura 10. Uso semanal Internet – Primer Iteración

El tiempo semanal que los usuarios dedican para trabajar en la computadora, es en porcentajes similar al uso del Internet.

	Frecuencia	Porcentaje	Porcentaje Válido	Porcentaje Acumulado
Menos 1 hr.	0	0	0	0
1 a 4 hrs.	11	18.6	18.6	18.6
4 a 10 hrs.	18	30.6	30.6	49.2
Más de 10	30	50.8	50.8	100
Total	59	100	100	

Tabla 12. Tabla de frecuencias uso semanal de Internet – Primer Iteración

Uso semanal PC 1er Iteración

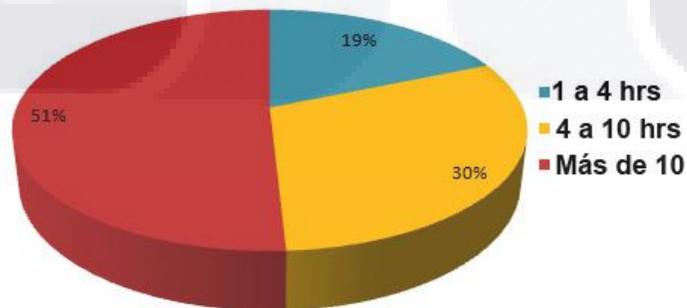


Figura 11. Uso semanal PC – Primer Iteración

4.3.3.2 Segunda Iteración

Para la segunda iteración se utilizó el mismo grupo por tratarse de muestras pareadas, la evaluación se aplicó a un total de 56 usuarios, es decir que existió una mortandad de 3 personas en relación a la primera iteración. Los datos demográficos se mantuvieron en los porcentajes de la primera iteración, pese a la mortandad encontrada, a continuación se muestran la información recopilada.

	Frecuencia	Porcentaje	Porcentaje -Válido	Porcentaje Acumulado
20 o menos	6	10.7	10.7	10.7
21 a 25	49	87.5	87.5	98.2
26 a 30	1	1.8	1.8	100
Total	56	100	100	

Tabla 13. Tabla de Frecuencias de Edad – Segunda Iteración

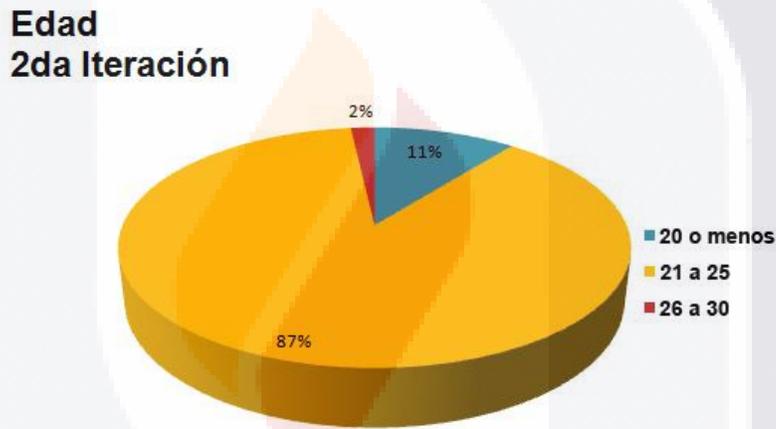


Figura 12. Edad – Segunda Iteración.

Genero segunda iteración.

	Frecuencia	Porcentaje	Porcentaje -Válido	Porcentaje Acumulado
Masculino	44	78.6	78.6	76.3
Femenino	12	21.4	21.4	100
Total	56	100	100	

Tabla 14. Tabla de Frecuencias de Género – Segunda Iteración

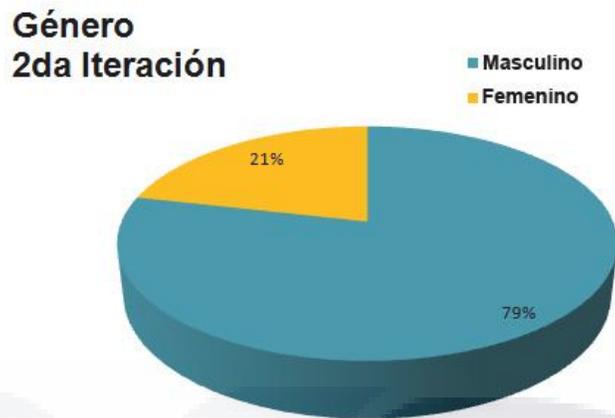


Figura 13. Género – Segunda Iteración.

Experiencia en uso de la computadora.

	Frecuencia	Porcentaje	Porcentaje -Válido	Porcentaje Acumulado
Mucha	19	34	34	34
Suficiente	30	53.3	53.3	87.3
Adecuada	5	9	9	96.3
Regular	2	3.7	3.7	100
Poca	0	0	0	
Muy Poca	0	0	0	
Nada	0	0	0	
Total	56	100	100	

Tabla 15. Tabla de Frecuencias de Experiencia uso PC – Segunda Iteración

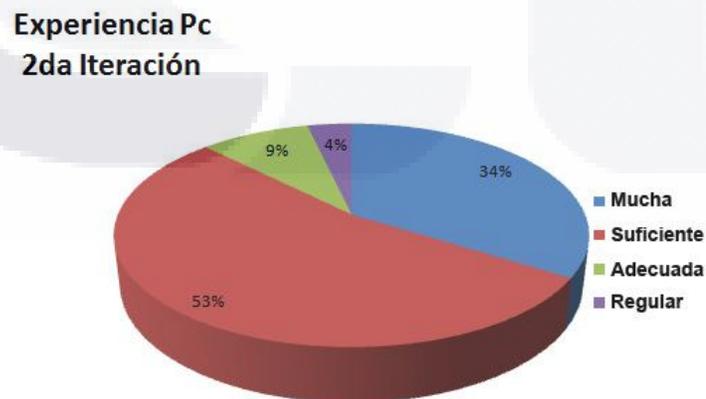


Figura 14. Experiencia uso PC – Segunda Iteración.

Uso semanal de Internet.

	Frecuencia	Porcentaje	Porcentaje Válido	Porcentaje Acumulado
Menos 1 hr	0	0	0	0
1 a 4 hrs	12	21.4	21.4	21.4
4 a 10 hrs	19	34	34	55.4
Más de 10	25	44.6	44.6	100
Total	56	100	100	

Tabla 16. Tabla de Frecuencias de uso semanal Internet – Segunda Iteración

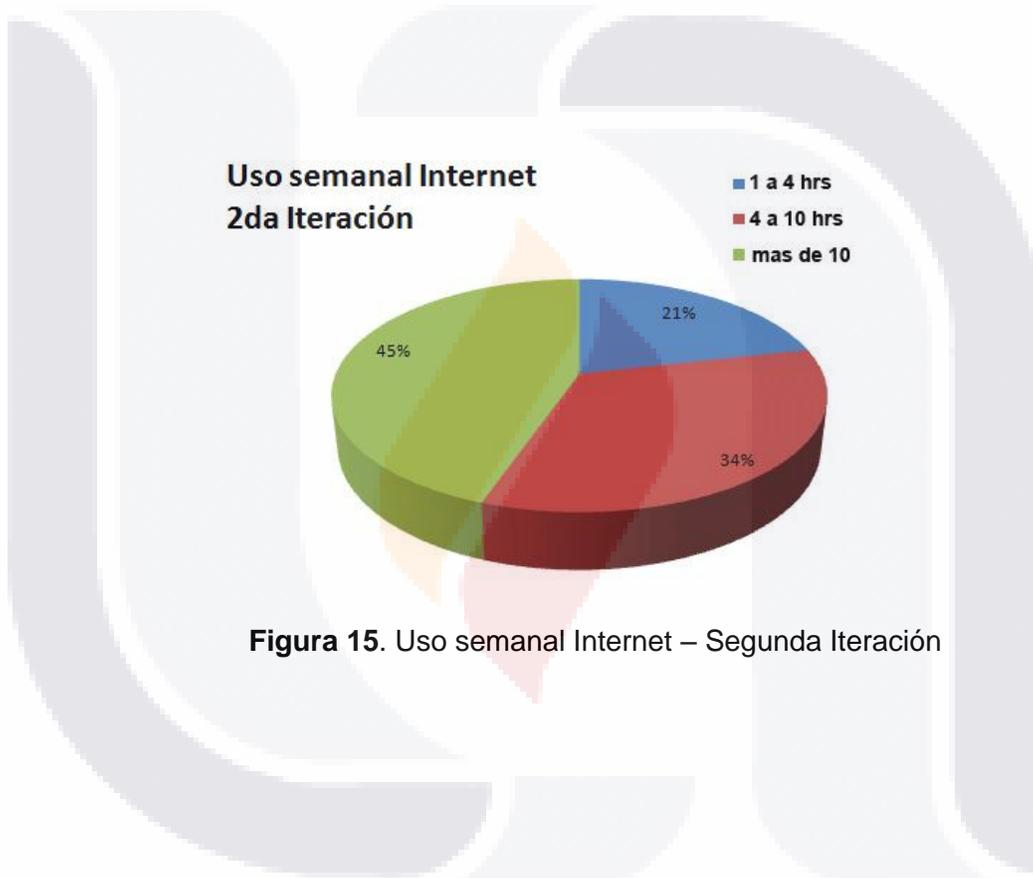


Figura 15. Uso semanal Internet – Segunda Iteración

CAPÍTULO 5. ANÁLISIS DE DATOS

5.1 Evaluación por parte del usuario.

La importancia de conocer la opinión del usuario final en esta investigación es fundamental, ya que de éste depende el éxito o fracaso del producto de software. Por este motivo 5 de los 7 aspectos de calidad se evaluaron desde el punto de vista del usuario final, esto gracias a la herramienta de evaluación que se desarrolló previamente.

Con el fin de evaluar la calidad de cada aspecto y poder tomar decisiones, se considero como Excelente un valor promedio entre 1 y 1.5, 1.5 a 2 como muy buena, 2 a 2.5 mínima aceptable, y superior a 2.5 mala (Gómez-Reynoso, Muñoz-Andrade 2010).

Este capítulo muestra el análisis de datos que se recolectaron en el transcurso de las iteraciones realizadas durante esta investigación, con ello se puede identificar de manera clara los resultados de la participación de los usuarios y desarrolladores en la evaluación de la calidad de los productos de software. Las siguientes secciones describen los hallazgos de la presente investigación.

5.1.2 Primera iteración

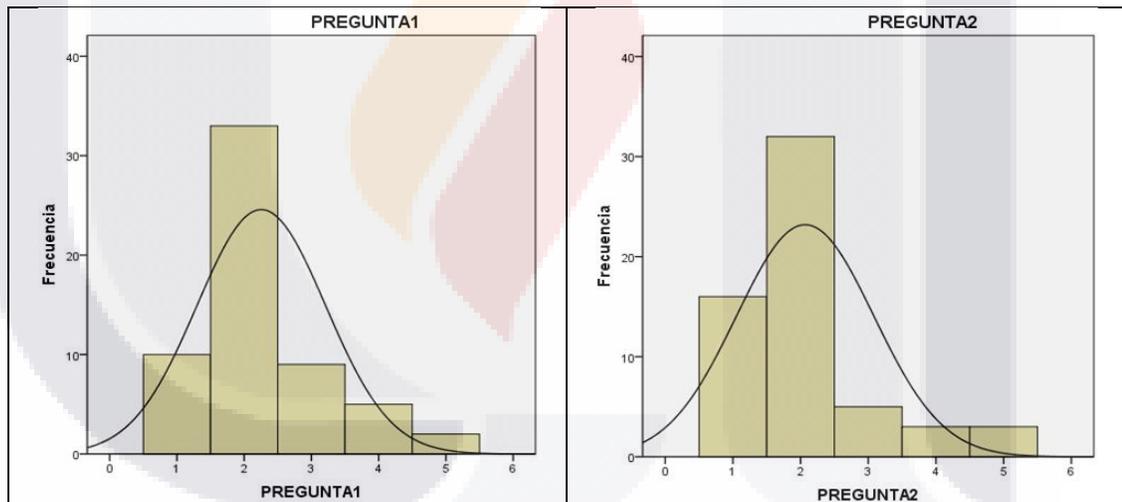
Con respecto al factor *Operabilidad* que es la capacidad del software para permitirle al usuario su operación y control, se evaluó con 4 ítems (preguntas) del cuestionario las cuales arrojaron los siguientes resultados (ver tabla 17).

Estadísticas	ITEM01	ITEM02	ITEM03	ITEM04
Casos validos	59	59	59	59
Casos perdidos	0	0	0	0
Media	2.2542	2.0678	3.6271	3.3220
Mediana	2.0000	2.0000	3.0000	3.0000
Moda	2.00	2.00	3.00	2.00
Desviación Estándar	.95761	1.01479	1.65985	1.37013
Varianza	.917	1.030	2.755	1.877
Asimetría	1.046	1.397	.226	.138
Error estándar de asimetría	.311	.311	.311	.311
Curtosis	1.108	2.034	-1.016	-1.094
Error estándar de curtosis	.613	.613	.613	.613
Rango	4.00	4.00	6.00	5.00
Mínimo	1.00	1.00	1.00	1.00
Máximo	5.00	5.00	7.00	6.00

a. Existen múltiples modas. Se muestra el valor más pequeño.

Tabla 17. Estadísticas del factor *Operabilidad*.

Los histogramas para el factor *Operabilidad* por ítem son como se muestra a continuación (figura 16).



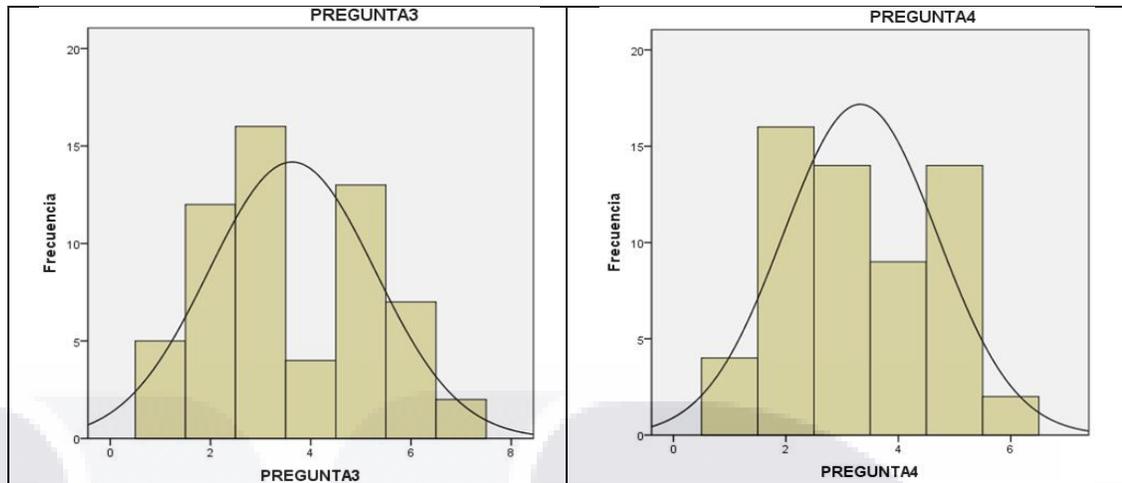


Figura 16. Histogramas - ítems factor *Operabilidad*.

La calidad promedio para el factor *Operabilidad* fue de 2.8, lo cual indica acorde a la escala utilizada que la *Operabilidad* para la primera versión del sistema posee una mala calidad.

Con respecto al factor *Facilidad de Aprendizaje* que es la capacidad del software para permitirle al usuario aprender su aplicación, se evaluó con 4 ítems del cuestionario las cuales arrojaron los siguientes resultados (ver tabla 18).

Estadísticas	ITEM01	ITEM02	ITEM03	ITEM04
Casos validos	59	59	59	59
Casos perdidos	0	0	0	0
Media	2.4746	2.1017	3.3051	3.2881
Mediana	2.0000	2.0000	3.0000	3.0000
Moda	2.00	2.00	3.00	2.00
Desviación Estándar	1.46636	1.19906	1.66355	1.70244
Varianza	2.150	1.438	2.767	2.898
Asimetría	1.699	1.661	.378	.594
Error estándar de asimetría	.311	.311	.311	.311
Curtosis	2.880	3.846	-.872	-.795
Error estándar de curtosis	.613	.613	.613	.613
Rango	6.00	6.00	6.00	6.00
Mínimo	1.00	1.00	1.00	1.00
Máximo	7.00	7.00	7.00	7.00

a. Existen múltiples modas. Se muestra el valor más pequeño.

Tabla 18. Estadísticas del factor *Facilidad de Aprendizaje*.

Los histogramas para el factor *Facilidad de Aprendizaje* por ítem son como se muestra a continuación (figura 17).

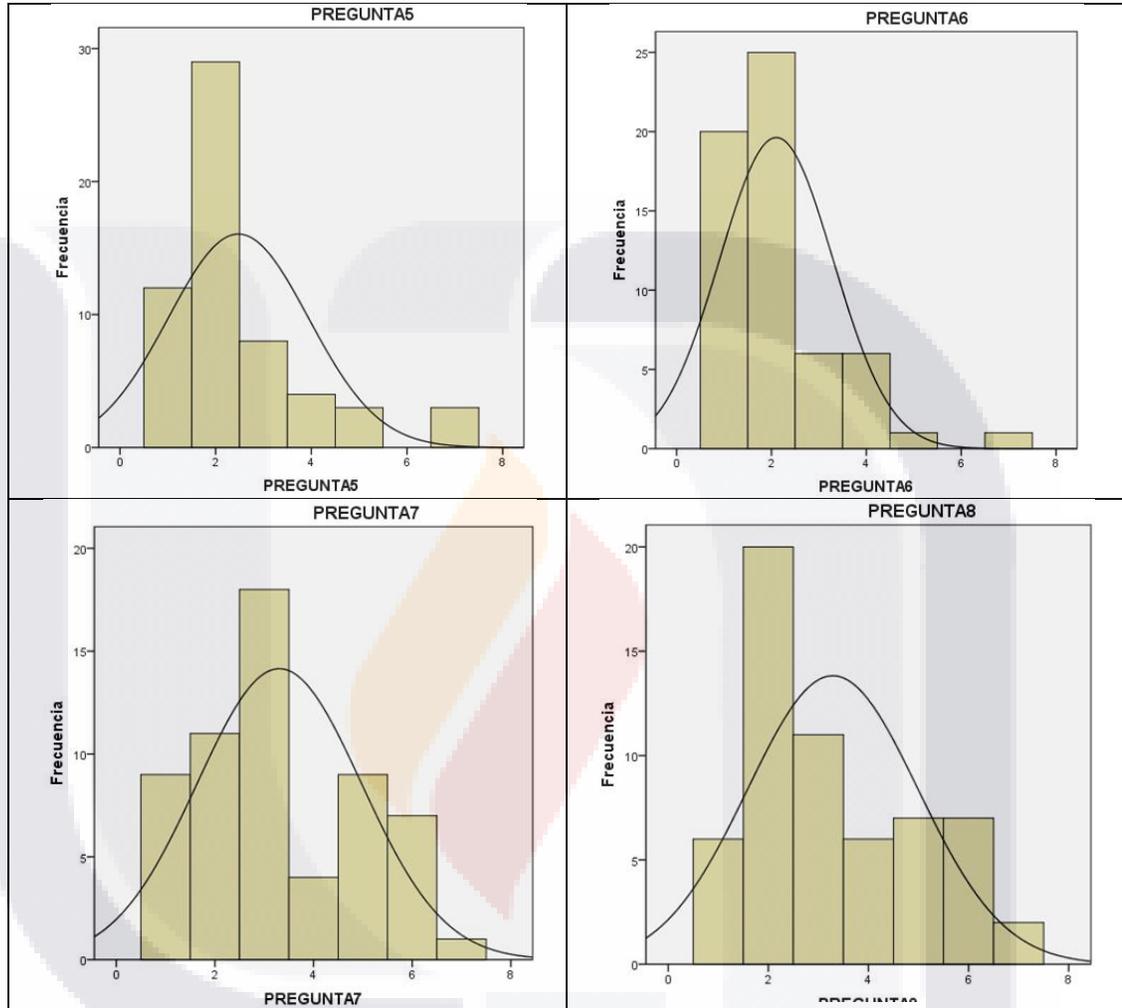


Figura 17. Histogramas - ítems factor *facilidad de aprendizaje*.

La calidad promedio para el factor *Facilidad de Aprendizaje* fue de 2.78, lo cual indica según a la escala utilizada que la *Facilidad de Aprendizaje* para la primera versión del sistema posee una mala calidad.

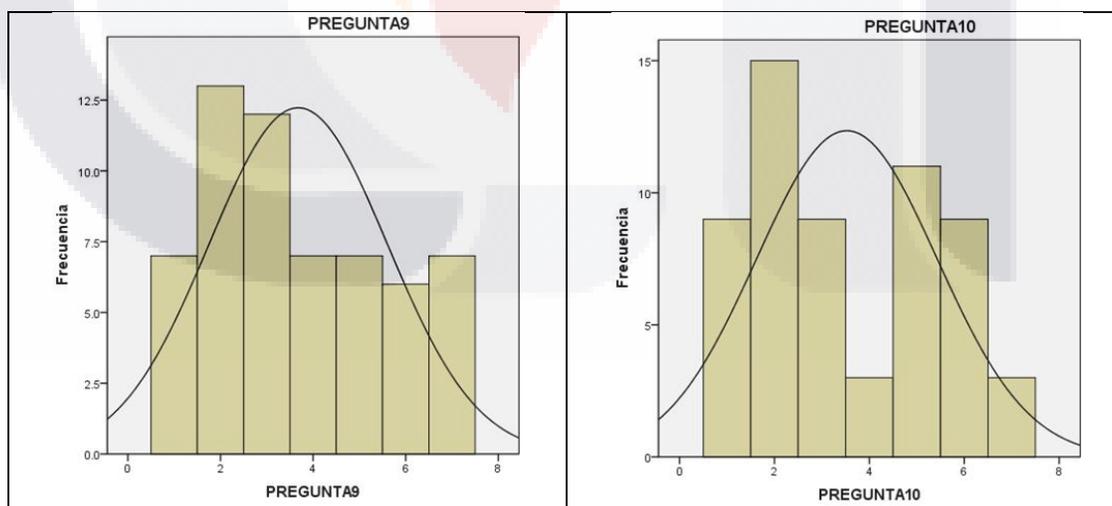
Con respecto al factor *Atractivo* que es la característica del software para influir en la satisfacción de los deseos del usuario, se evaluó con 4 ítems del cuestionario las cuales arrojaron los siguientes resultados (ver tabla 19).

Estadísticas	ITEM01	ITEM02	ITEM03	ITEM04
Casos validos	59	59	59	59
Casos perdidos	0	0	0	0
Media	3.6780	3.5254	3.9661	3.1356
Mediana	3.0000	3.0000	4.0000	2.0000
Moda	2.00	2.00	5.00	2.00
Desviación Estándar	1.92480	1.90603	1.97367	2.01253
Varianza	3.705	3.633	3.895	4.050
Asimetría	.372	.253	.090	.846
Error estándar de asimetría	.311	.311	.311	.311
Curtosis	-1.053	-1.315	-1.133	-.517
Error estándar de curtosis	.613	.613	.613	.613
Rango	6.00	6.00	6.00	6.00
Mínimo	1.00	1.00	1.00	1.00
Máximo	7.00	7.00	7.00	7.00

a. Existen múltiples modas. Se muestra el valor más pequeño.

Tabla 19. Estadísticas del factor *Atractivo*.

Los histogramas para el factor *Atractivo* por ítem son como se muestra a continuación (figura 18).



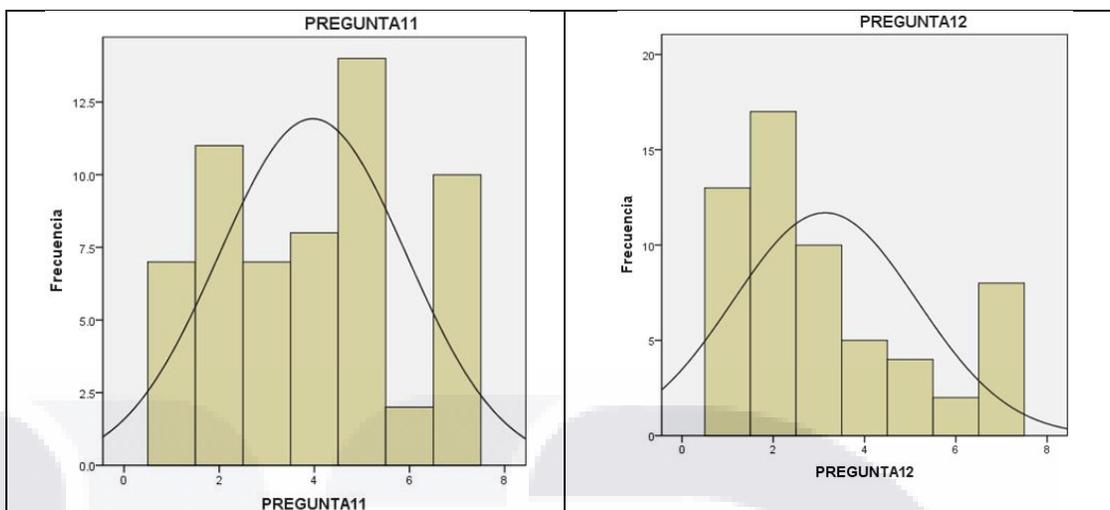


Figura 18. Histogramas - ítems factor *Atractivo*.

La calidad promedio para el factor *Atractivo* fue de 3.57, lo cual indica según a la escala utilizada que el *Atractivo* para la primera versión del sistema posee una mala calidad.

Con respecto al factor *Comprensibilidad* que es la característica del software que influyen en el esfuerzo del usuario para reconocer el concepto lógico y su aplicación, se evaluó con 4 ítems del cuestionario, las cuales arrojaron los siguientes resultados (ver tabla 20).

Estadísticas	ITEM01	ITEM02	ITEM03	ITEM04
Casos validos	59	59	59	59
Casos perdidos	0	0	0	0
Media	2.4915	2.3051	3.3559	3.0339
Mediana	2.0000	2.0000	3.0000	3.0000
Moda	2.00	2.00	2.00	3.00
Desviación Estándar	1.54666	1.28994	1.62696	1.58621
Varianza	2.392	1.664	2.647	2.516
Asimetría	1.782	1.546	.590	.533
Error estándar de asimetría	.311	.311	.311	.311
Curtosis	2.900	2.810	-.453	-.235
Error estándar de curtosis	.613	.613	.613	.613
Rango	6.00	6.00	6.00	6.00
Mínimo	1.00	1.00	1.00	1.00
Máximo	7.00	7.00	7.00	7.00

a. Existen múltiples modas. Se muestra el valor más pequeño.

Tabla 20. Estadísticas del factor *Comprensibilidad*.

Los histogramas para el factor *Comprensibilidad* por ítem son como se muestra a continuación (figura 19).

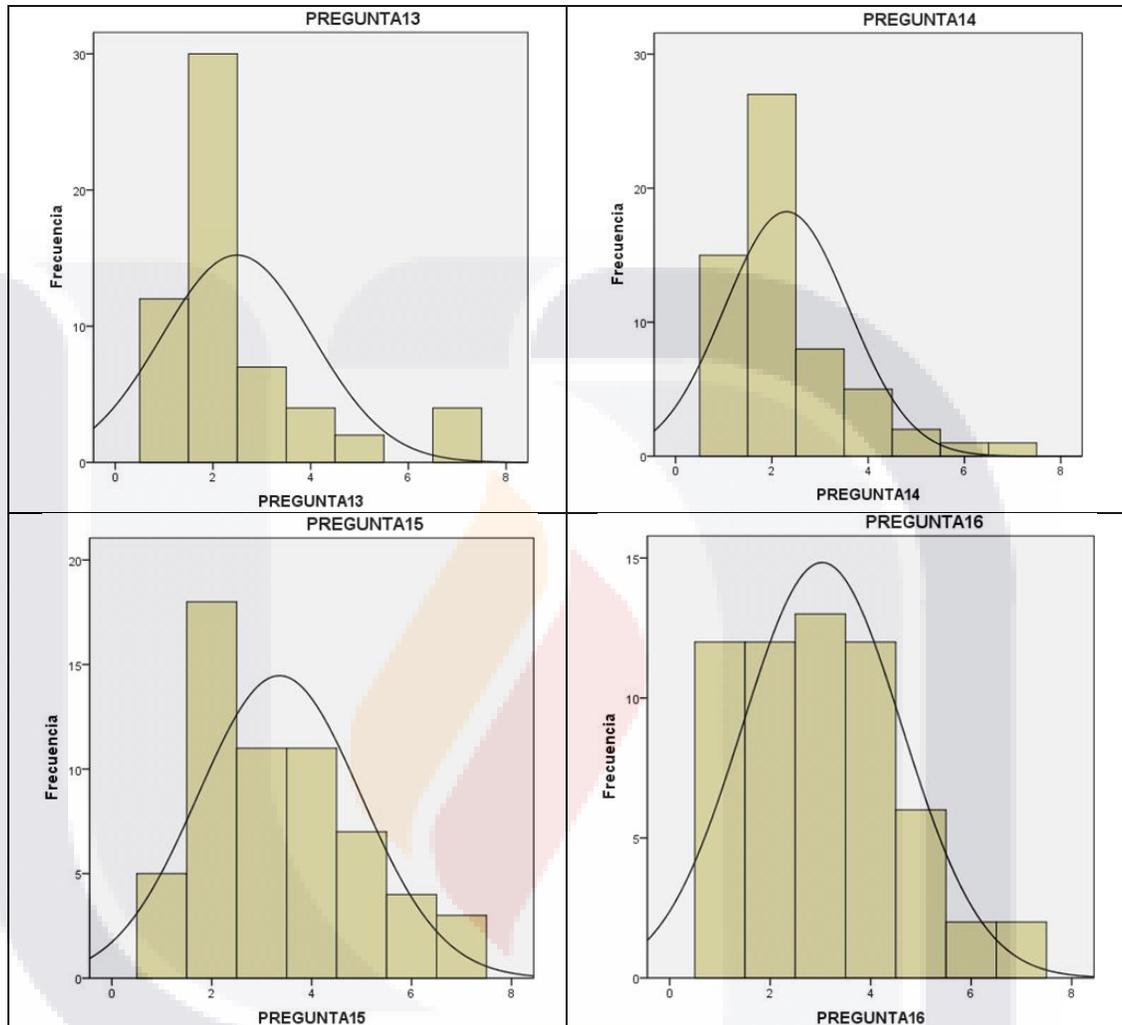


Figura 19. Histogramas - ítems factor *Comprensibilidad*.

La calidad promedio para el factor *Comprensibilidad* fue de 2.79, lo cual indica según a la escala utilizada que la *Comprensibilidad* para la primera versión del sistema posee una mala calidad.

Con respecto al factor *Conformidad* que es la característica del software para proveer funciones adecuadas para tareas específicas, se evaluó con 3 ítems del cuestionario las cuales arrojaron los siguientes resultados (ver tabla 21).

Estadísticas	ITEM01	ITEM02	ITEM03
Casos validos	59	59	59
Casos perdidos	0	0	0
Media	2.4746	3.1525	3.4407
Mediana	2.0000	3.0000	3.0000
Moda	2.00	2.00	2.00
Desviación Estándar	1.10416	1.39967	1.26315
Varianza	1.219	1.959	1.596
Asimetría	.942	.773	.484
Error estándar de asimetría	.311	.311	.311
Curtosis	.937	.140	-.685
Error estándar de curtosis	.613	.613	.613
Rango	5.00	6.00	4.00
Mínimo	1.00	1.00	2.00
Máximo	6.00	7.00	6.00

a. Existen múltiples modas. Se muestra el valor más pequeño.

Tabla 21. Estadísticas del factor *Conformidad*.

Los histogramas para el factor *Conformidad* por ítem son como se muestra a continuación (figura 20).

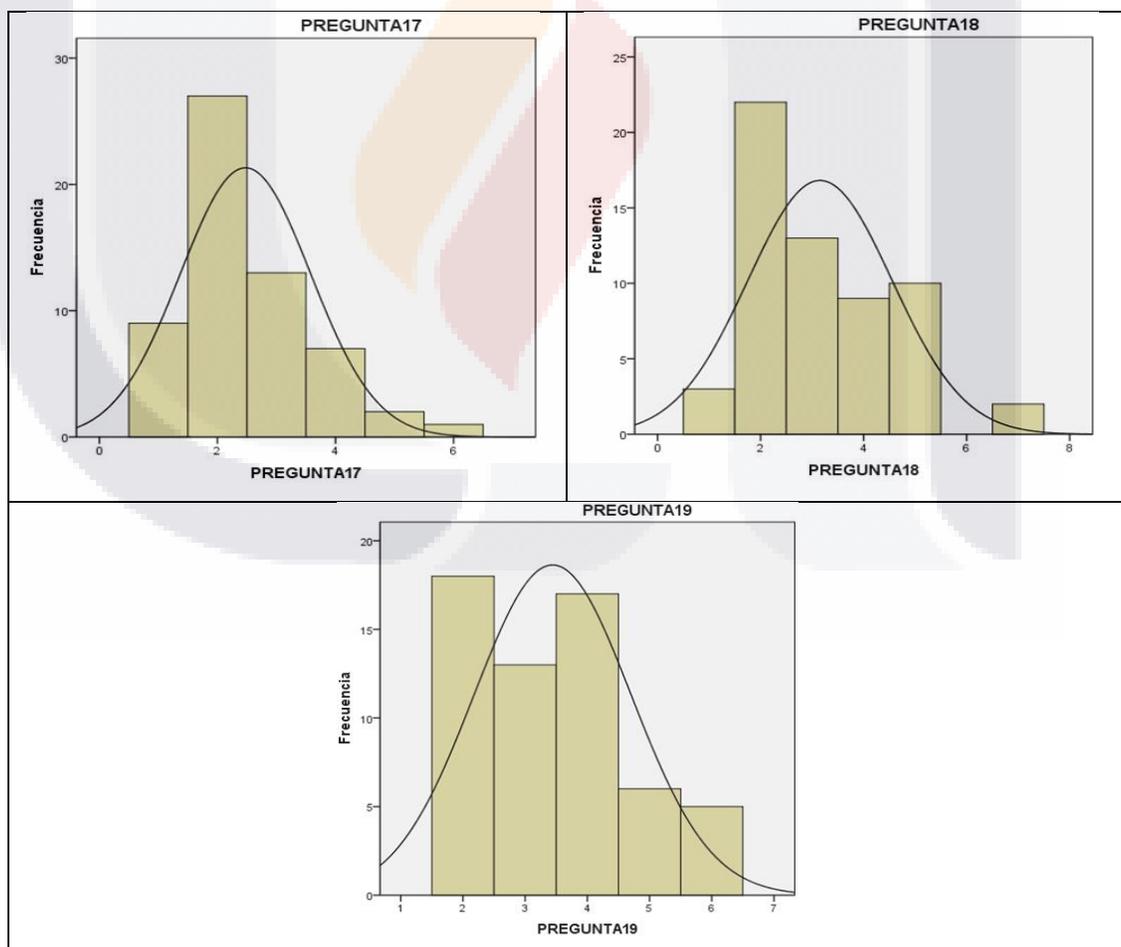


Figura 20. Histogramas - ítems factor *conformidad*.

La calidad promedio para el factor *Conformidad* fue de 3.02, lo cual indica según a la escala utilizada que la *Conformidad* para la primera versión del sistema posee una mala calidad.

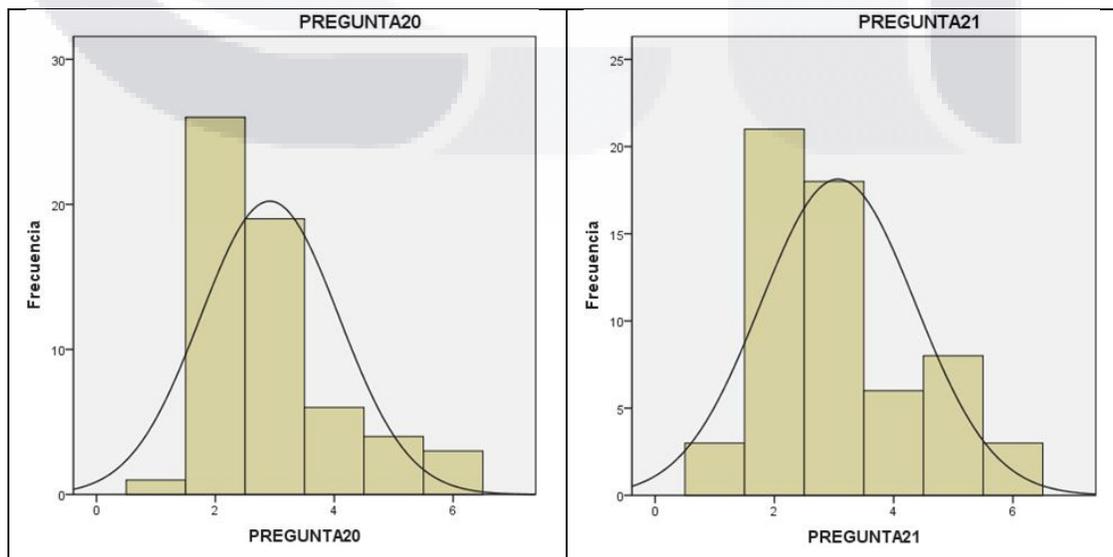
Con respecto al factor *Exactitud* que es la capacidad del software de proporcionar los resultados correctos, precisos y satisface los objetivos del cliente, se evaluó con 3 ítems del cuestionario, los cuales arrojaron los siguientes resultados (ver tabla 22).

Estadísticas	ITEM01	ITEM02	ITEM03
Casos validos	59	59	59
Casos perdidos	0	0	0
Media	2.9153	3.0678	3.2373
Mediana	3.0000	3.0000	3.0000
Moda	2.00	2.00	3.00
Desviación Estándar	1.16394	1.29807	1.26408
Varianza	1.355	1.685	1.598
Asimetría	1.188	.703	.859
Error estándar de asimetría	.311	.311	.311
Curtosis	.872	-.333	.315
Error estándar de curtosis	.613	.613	.613
Rango	5.00	5.00	6.00
Mínimo	1.00	1.00	1.00
Máximo	6.00	6.00	7.00

a. Existen múltiples modas. Se muestra el valor más pequeño.

Tabla 22. Estadísticas del factor *Exactitud*.

Los histogramas para el factor *Exactitud* por ítem son como se muestra a continuación (figura 21).



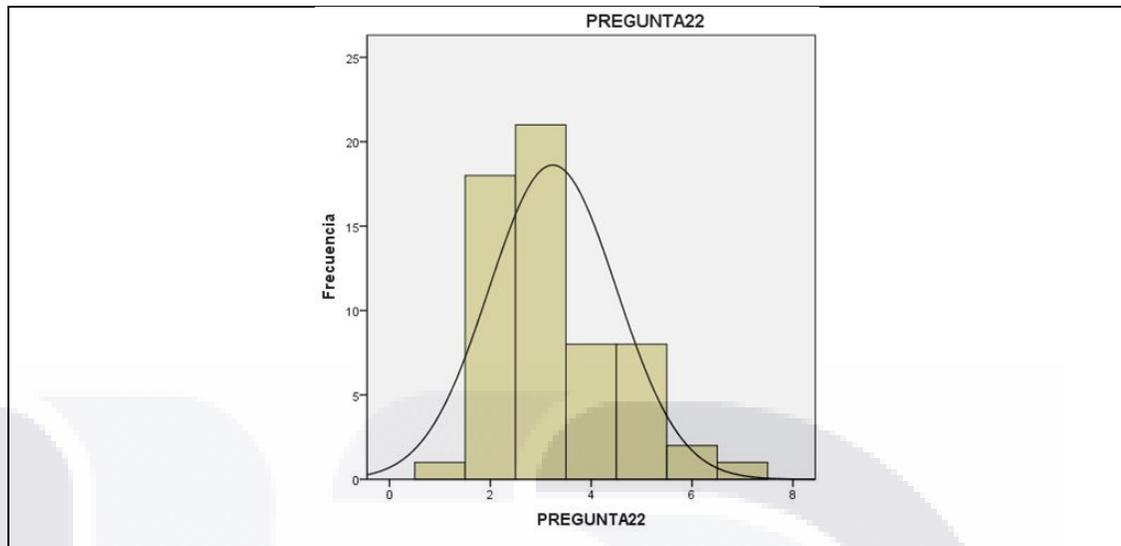


Figura 21. Histogramas - ítems factor *Exactitud*.

La calidad promedio para el factor *Exactitud* fue de 3.06, lo cual indica según a la escala utilizada que la *Exactitud* para la primera versión del sistema posee una mala calidad.

Con respecto al factor *Madurez* que es la capacidad del software para evitar errores, se evaluó con 3 ítems del cuestionario, las cuales arrojaron los siguientes resultados (ver tabla 23).

Estadísticas	ITEM01	ITEM02	ITEM03
Casos validos	59	59	59
Casos perdidos	0	0	0
Media	3.4576	2.2542	2.9661
Mediana	4.0000	2.0000	3.0000
Moda	4.00	1.00	4.00
Desviación Estándar	1.35598	1.43351	1.57531
Varianza	1.839	2.055	2.482
Asimetría	.350	1.206	.633
Error estándar de asimetría	.311	.311	.311
Curtosis	.237	1.050	.135
Error estándar de curtosis	.613	.613	.613
Rango	6.00	6.00	6.00
Mínimo	1.00	1.00	1.00
Máximo	7.00	7.00	7.00

a. Existen múltiples modas. Se muestra el valor más pequeño.

Tabla 23. Estadísticas del factor *Madurez*.

Los histogramas para el factor *Madurez* por ítem son como se muestra a continuación (figura 22).

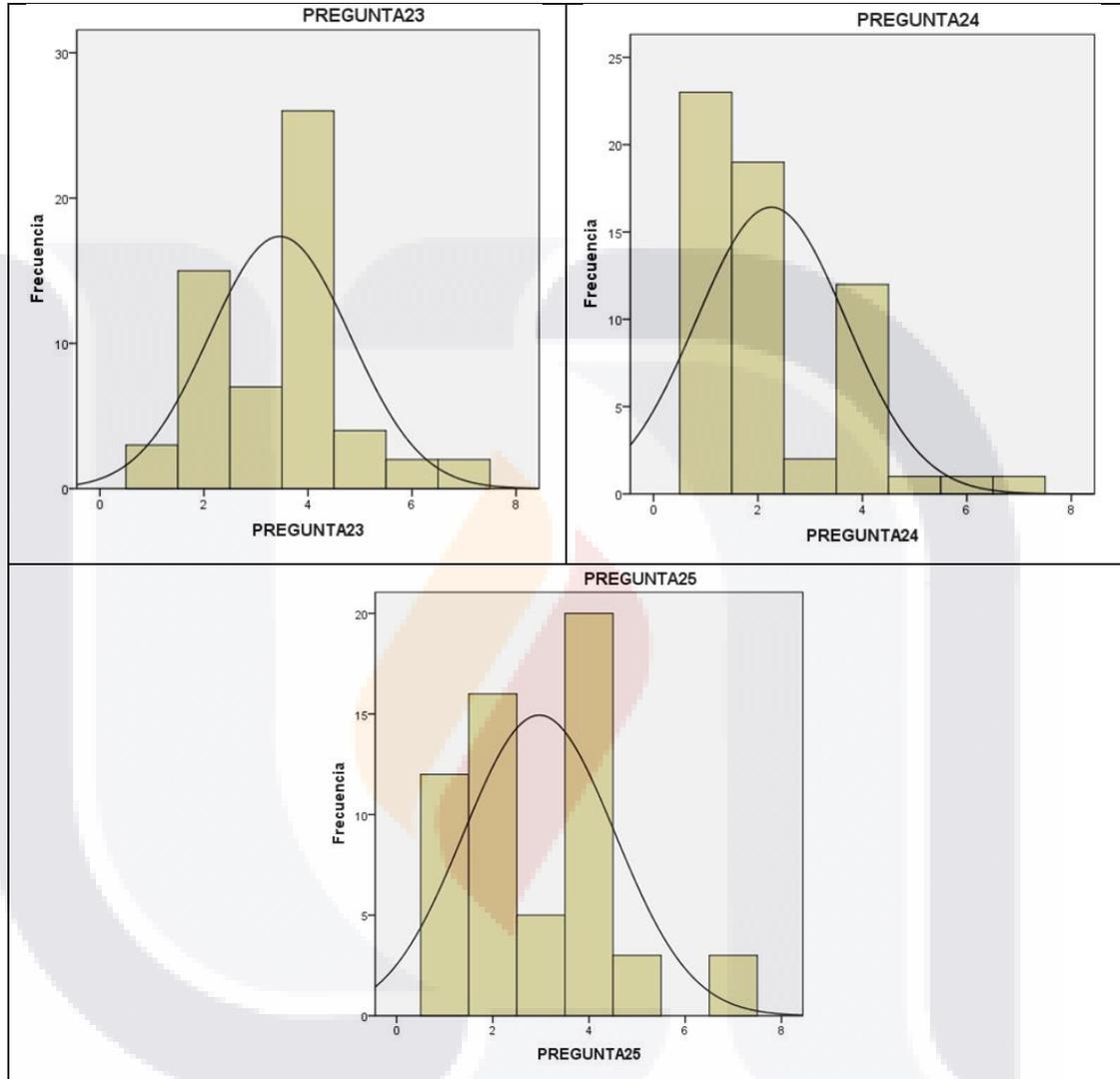


Figura 22. Histogramas - ítems factor *Madurez*.

La calidad promedio para el factor *Madurez* fue de 2.88, lo cual indica según a la escala utilizada que la *Madurez* para la primera versión del sistema posee una mala calidad.

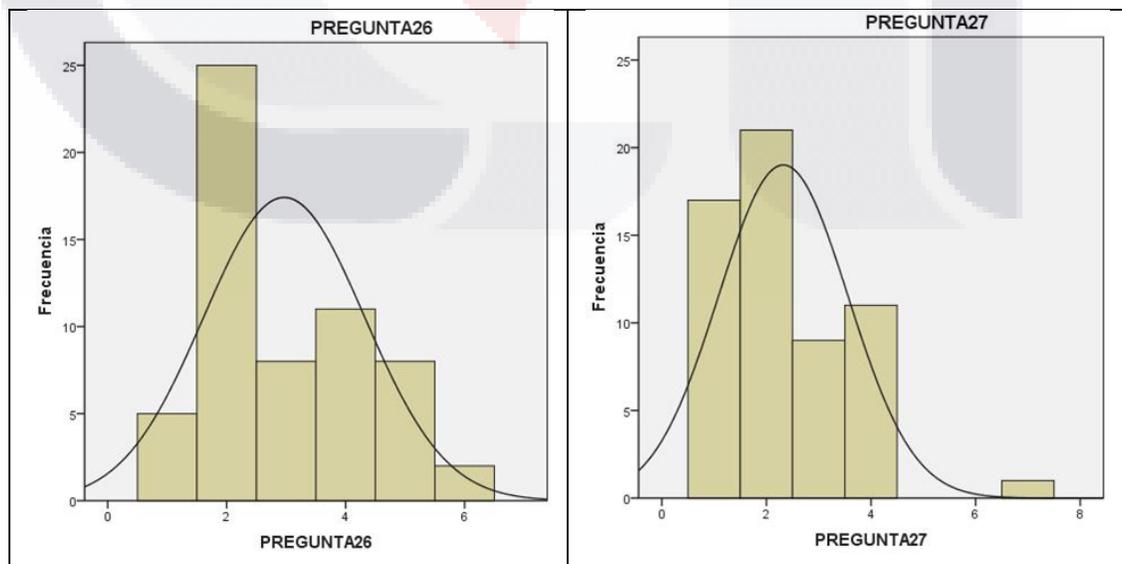
Con respecto al factor *Tolerancia a Fallos* que es la característica del producto de Software para mantener un desempeño adecuado en caso de fallas o errores, se evaluó con 3 ítems del cuestionario los cuales arrojaron los siguientes resultados (ver tabla 24).

Estadísticas	ITEM01	ITEM02	ITEM03
Casos validos	59	59	59
Casos perdidos	0	0	0
Media	2.9661	2.3220	3.3051
Mediana	2.0000	2.0000	4.0000
Moda	2.00	2.00	4.00
Desviación Estándar	1.35145	1.23792	1.41731
Varianza	1.826	1.532	2.009
Asimetría	.541	1.102	.487
Error estándar de asimetría	.311	.311	.311
Curtosis	-.780	1.888	.296
Error estándar de curtosis	.613	.613	.613
Rango	5.00	6.00	6.00
Mínimo	1.00	1.00	1.00
Máximo	6.00	7.00	7.00

a. Existen múltiples modas. Se muestra el valor más pequeño.

Tabla 24. Estadísticas del factor *Tolerancia a Fallos*.

Los histogramas para el factor *Tolerancia a Fallos* por ítem son como se muestra a continuación (figura 23).



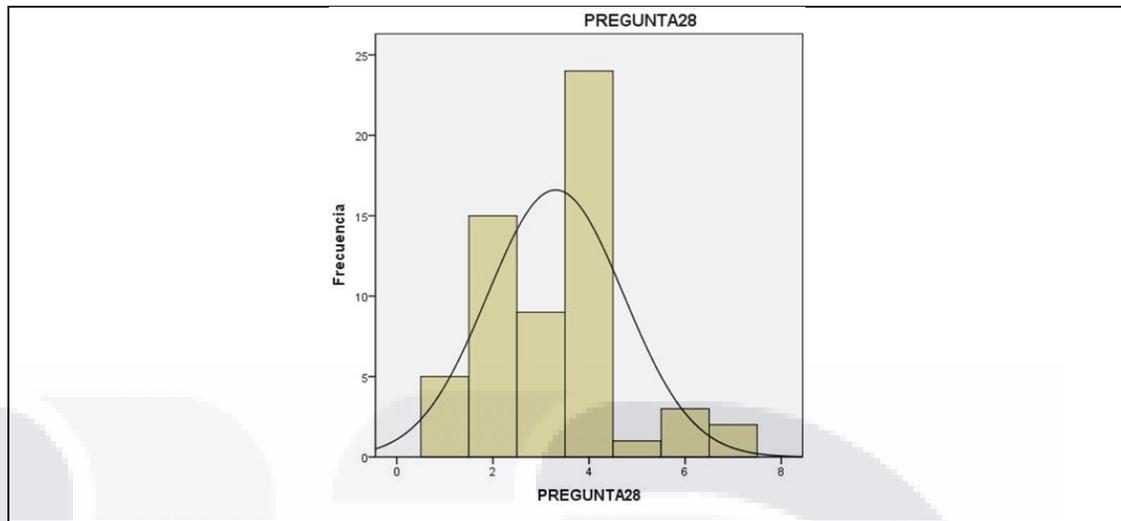


Figura 23. Histogramas - ítems factor *tolerancia a fallas*.

La calidad promedio para el factor *Tolerancia a Fallas* fue de 2.86, lo cual indica según a la escala utilizada que la *Tolerancia a Fallas* para la primera versión del sistema posee una mala calidad.

Con respecto al factor *Recuperabilidad* que es la característica del software que considera las acciones necesarias para recuperarse en caso de fallas de manera fácil y rápida., se evaluó con 3 ítems del cuestionario, los cuales arrojaron los siguientes resultados (ver tabla 25).

Estadísticas	ITEM01	ITEM02	ITEM03
Casos validos	59	59	59
Casos perdidos	0	0	0
Media	2.9153	2.8814	2.8136
Mediana	3.0000	3.0000	3.0000
Moda	4.00	4.00	2.00
Desviación Estándar	1.07139	1.09992	1.19589
Varianza	1.148	1.210	1.430
Asimetría	-.088	-.160	.561
Error estándar de asimetría	.311	.311	.311
Curtosis	-1.390	-1.288	.874
Error estándar de curtosis	.613	.613	.613
Rango	4.00	4.00	6.00
Mínimo	1.00	1.00	1.00
Máximo	5.00	5.00	7.00

a. Existen múltiples modas. Se muestra el valor más pequeño.

Tabla 25. Estadísticas del factor *Recuperabilidad*.

Los histogramas para el factor *Recuperabilidad* por ítem son como se muestra a continuación (figura 24).

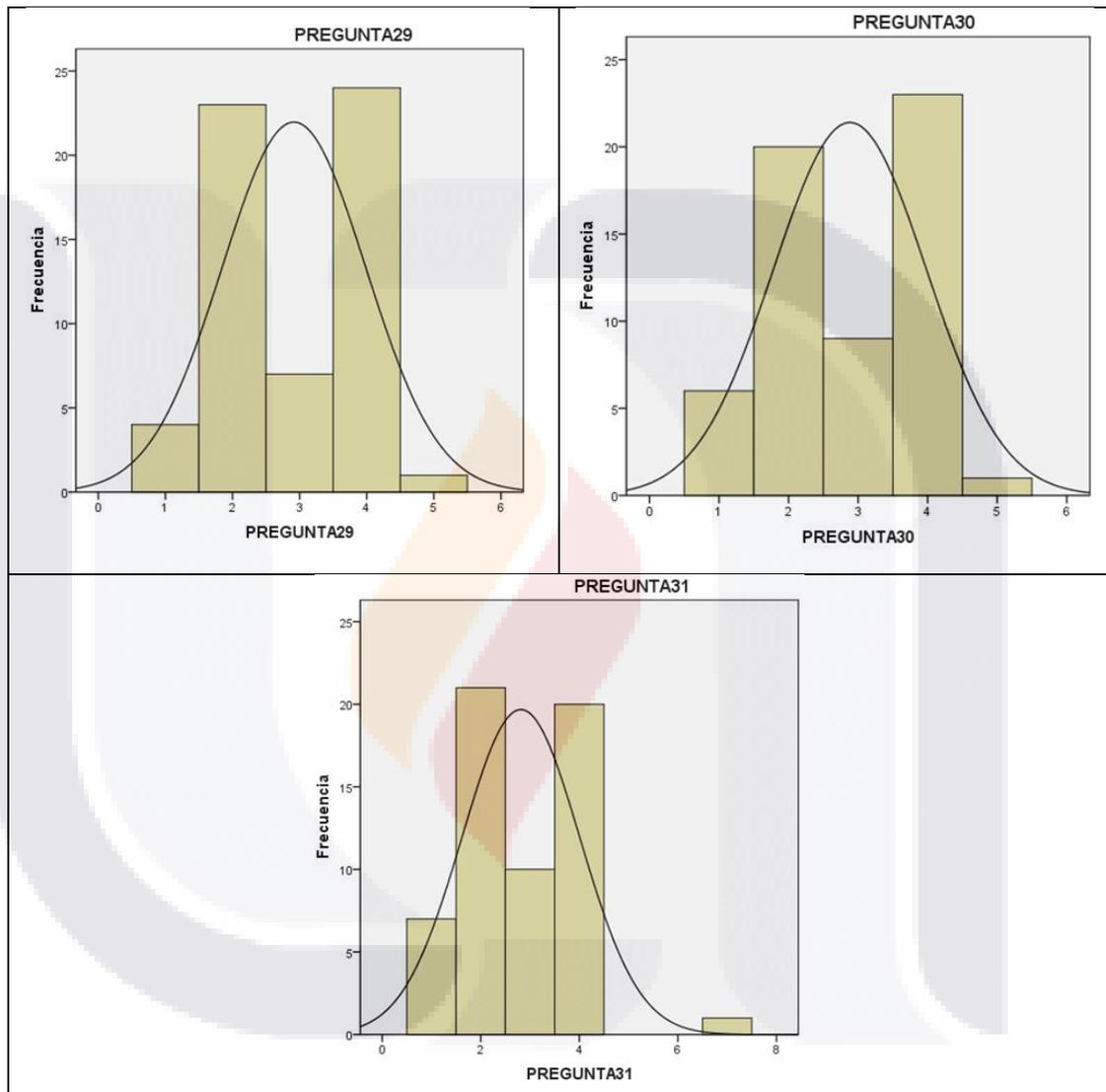


Figura 24. Histogramas - ítems factor *Recuperabilidad*.

La calidad promedio para el factor *Recuperabilidad* fue de 2.86, lo cual indica según a la escala utilizada que la *Recuperabilidad* para la primera versión del sistema posee una mala calidad.

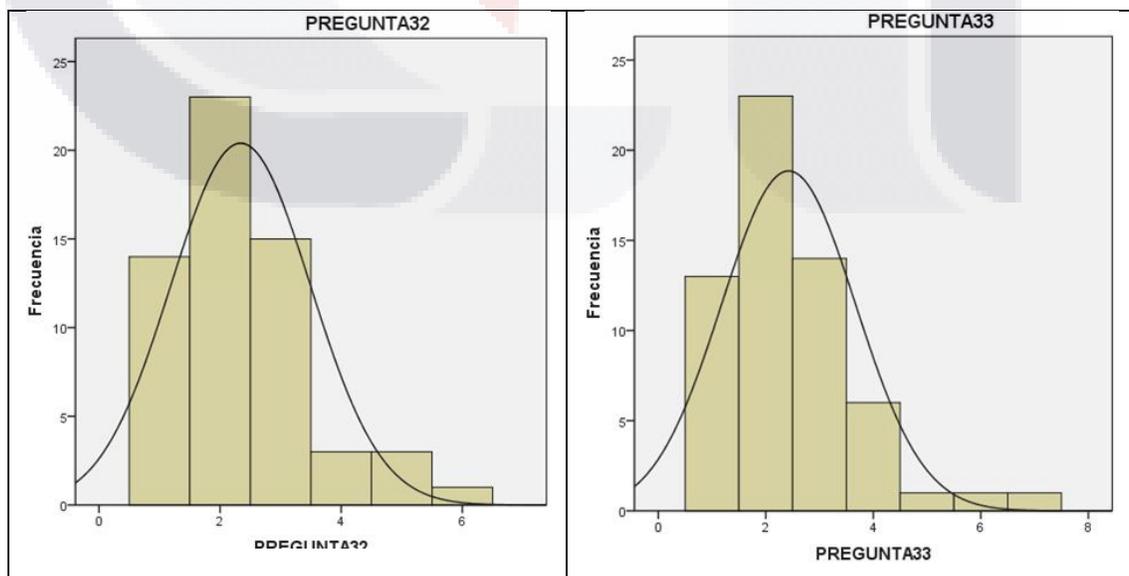
Con respecto al factor *Tiempo de Respuesta* que es la característica del software para proveer tiempos adecuados de respuesta al realizar sus diferentes funciones, se evaluó con 3 ítems del cuestionario, los cuales arrojaron los siguientes resultados (ver tabla 26).

Estadísticas	ITEM01	ITEM02	ITEM03
Casos validos	59	59	59
Casos perdidos	0	0	0
Media	2.3390	2.4237	2.5085
Mediana	2.0000	2.0000	2.0000
Moda	2.00	2.00	2.00
Desviación Estándar	1.15386	1.24849	1.31795
Varianza	1.331	1.559	1.737
Asimetría	1.036	1.335	1.267
Error estándar de asimetría	.311	.311	.311
Curtosis	1.165	2.654	1.839
Error estándar de curtosis	.613	.613	.613
Rango	5.00	6.00	6.00
Mínimo	1.00	1.00	1.00
Máximo	6.00	7.00	7.00

a. Existen múltiples modas. Se muestra el valor más pequeño.

Tabla 26. Estadísticas del factor *Tiempo de Respuesta*.

Los histogramas para el factor *Tiempo de Respuesta* por ítem son como se muestra a continuación (figura 25).



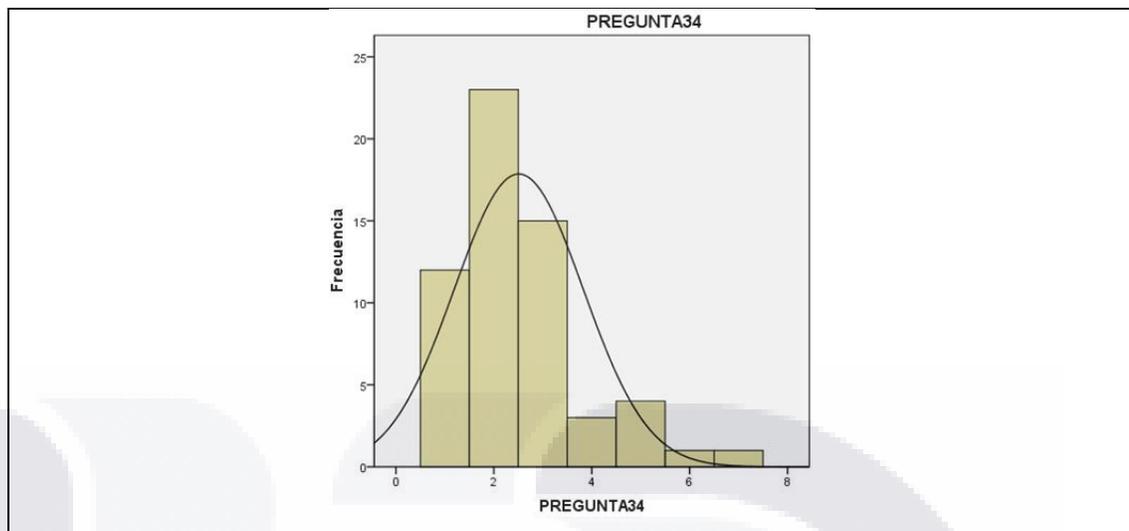


Figura 25. Histogramas - ítems factor *Tiempo de Respuesta*.

La calidad promedio para el factor *Tiempo de Respuesta* fue de 2.41, lo cual indica según a la escala utilizada que el *tiempo de respuesta* para la primera versión del sistema posee una calidad mínima aceptable.

Según los resultados obtenidos de los factores medidos en la primer versión del sistema, se puede decir que hace falta mejorar a cada uno de estos, pues aunque el tiempo de respuesta tiene una calidad mínima aceptable y algunos otros poseen valores cercanos, todos están lejos de tener una calidad óptima, por tanto fue necesario revisar los aspectos que afectan a que la percepción del usuario se acerque a una mejor calidad.

5.1.2 Segunda iteración

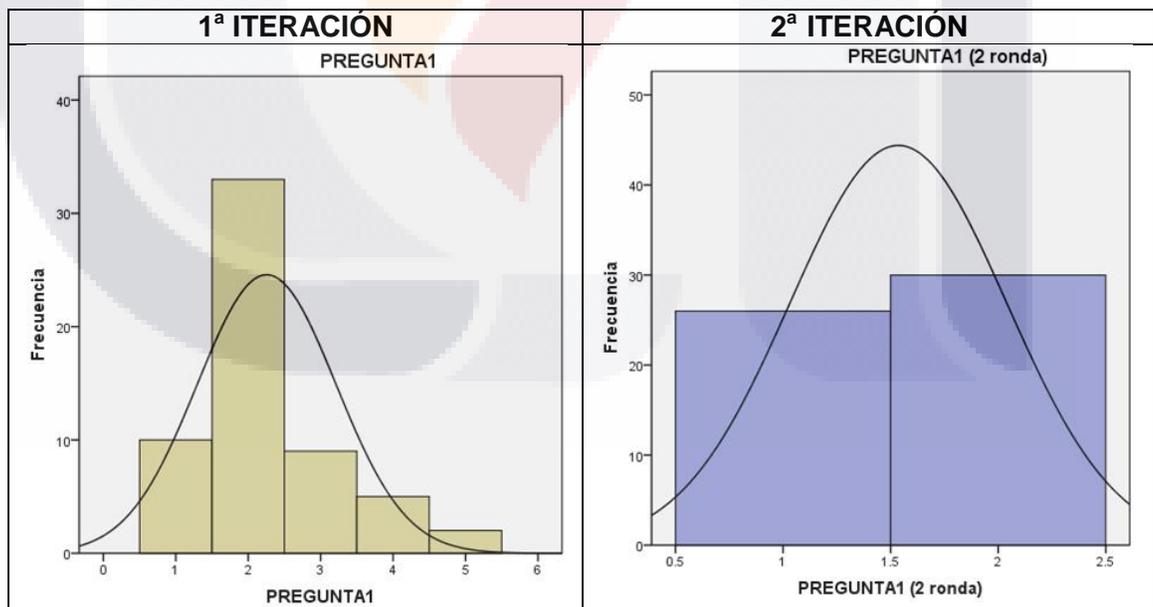
La presente sección muestra los resultados que se obtuvieron luego de hacer las modificaciones pertinentes al sistema en base a los resultados obtenidos en la primera iteración, por lo tanto se muestra un análisis comparativo, así como las pruebas de hipótesis en donde se realizaron por medio de muestras pareadas.

Para el factor de calidad *Operabilidad* se encontraron los siguientes datos durante la segunda iteración, en la tabla 27 se puede observar una comparativa de las estadísticas de cada iteración.

ESTADÍSTICAS	ITERACIONES	
	1ª	2ª
Casos validos	59	56
Casos perdidos	0	0
Media	2.8178	1.5982
Mediana	2.0000	2.0000
Moda	2.00	2.00
Desviación Estándar	1.43981	.49136
Varianza	2.073	.241
Asimetría	.780	-.403
Error estándar de asimetría	.158	.163
Curtosis	-.270	-1.854
Error estándar de curtosis	.316	.324
Rango	6.00	1.00
Mínimo	1.00	1.00
Máximo	7.00	2.00
a. Existen múltiples modas. Se muestra el valor más pequeño.		

Tabla 27. Estadística comparativa entre iteraciones del factor *Operabilidad*.

La comparativa de los histogramas para *Operabilidad* de las iteraciones realizadas para cada pregunta es cómo se muestra a continuación (ver figura 26).



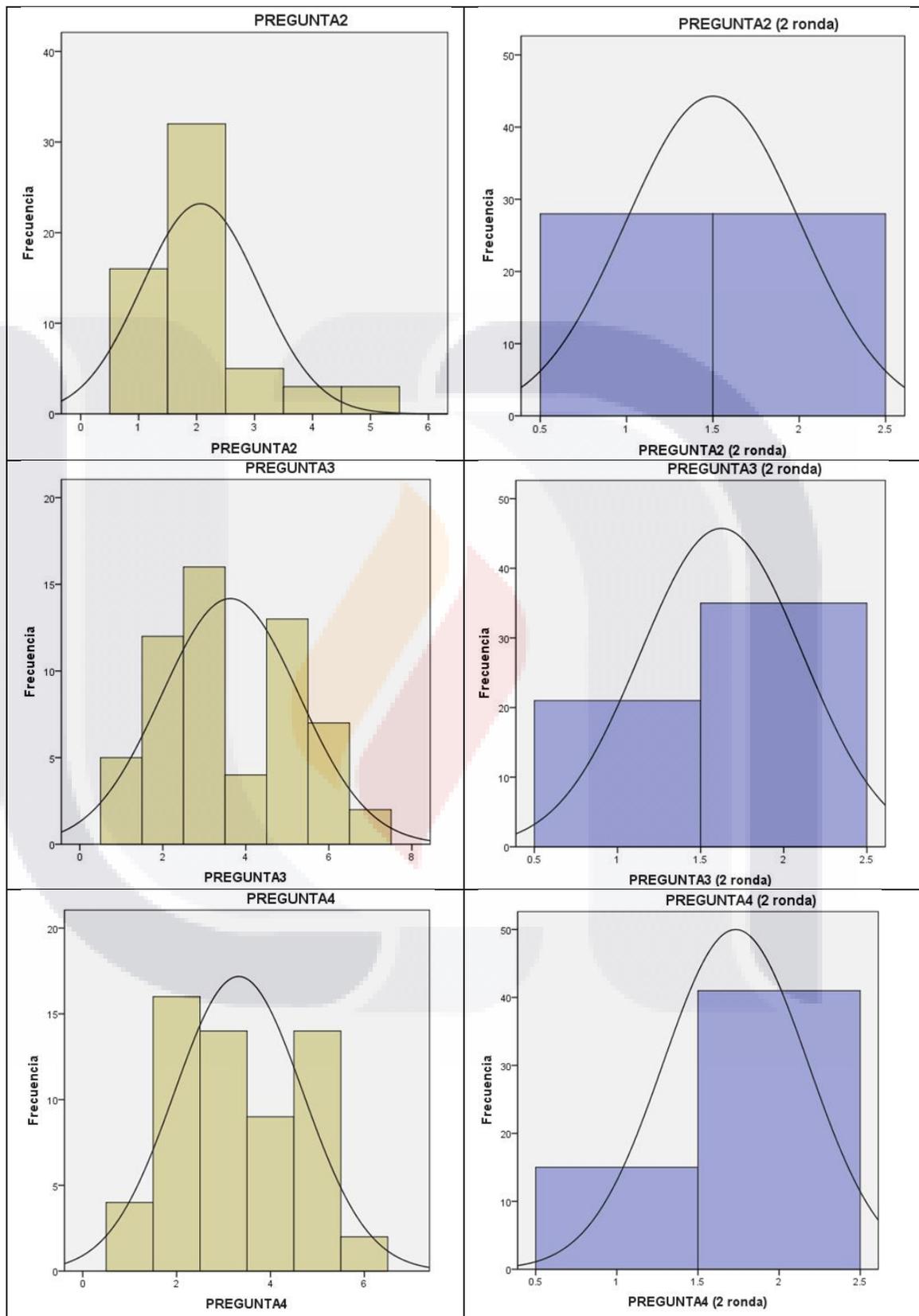


Figura 26. Comparativa de Histogramas entre iteraciones de Operabilidad.

La prueba de muestras pareadas para *Operabilidad* con la cual se aprueba o rechaza la hipótesis dio el siguiente resultado (ver tabla 28).

Items operabilidad	DIFERENCIAS RELACIONADAS					t	gl	Sig. (bilateral)
	Media	Desviación típ.	Error típ. de la media	95% Intervalo de confianza para la diferencia				
				Inferior	Superior			
Par1 Pregunta1 - Pregunta1 (2 Ronda)	.732	1.036	.138	.455	1.009	5.291	55	.000
Par2 Pregunta2 - Pregunta2 (2 Ronda)	.571	1.110	.148	.274	.869	3.854	55	.000
Par3 Pregunta3 - Pregunta3 (2 Ronda)	1.929	1.693	.226	1.475	2.382	8.523	55	.000
Par4 Pregunta4 - Pregunta4 (2 Ronda)	1.589	1.424	.190	1.208	1.971	8.351	55	.000

Tabla 28. Prueba de muestras pareadas factor de calidad *Operabilidad*.

Para la primera versión del sistema la mayoría de los usuarios opinó que estaba algo de acuerdo en la calidad de la *Operabilidad* del sistema, sin embargo para la segunda versión la mayoría de los usuarios cambió su opinión y dijo estar Totalmente de acuerdo.

La calidad encontrada para *Operabilidad* en la segunda iteración fue de 1.5 (ver tabla 27) por lo tanto de acuerdo a la escala utilizada se puede decir que el factor *Operabilidad* en la segunda versión del sistema cuenta con una calidad excelente.

El factor *Operabilidad* tiene una diferencia significativa ($p \leq .001$) (ver Tabla 28), por lo tanto significa que la calidad en este factor tiene una medida muy importante desde el punto de vista del usuario final. Esto confirma la hipótesis H1a de que en la nueva versión de la aplicación, el factor de calidad *Operabilidad* mejoró significativamente.

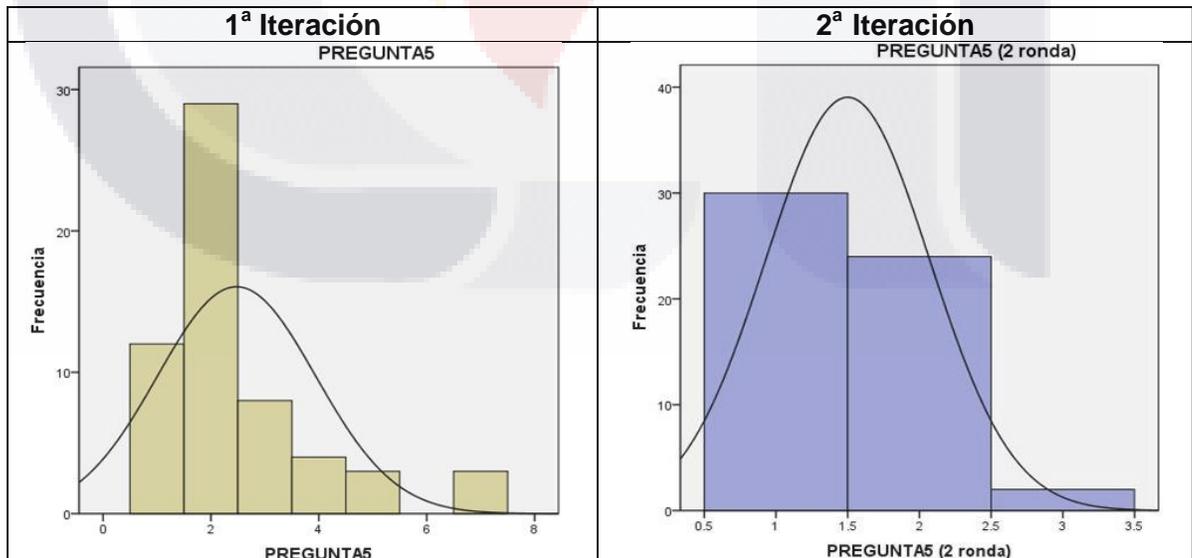
Para el factor *Facilidad de Aprendizaje* se encontraron los siguientes datos en la segunda iteración, en la tabla 29 se pueden observar la comparativa de las estadísticas de cada iteración.

ESTADÍSTICAS	ITERACIONES	
	1ª	2ª
Casos validos	59	56
Casos perdidos	0	0
Media	2.7924	1.6116
Mediana	2.0000	2.0000
Moda	2.00	2.00
Desviación Estándar	1.59897	.58074
Varianza	2.557	.337
Asimetría	.967	.318
Error estándar de asimetría	.158	.163
Curtosis	.108	-.724
Error estándar de curtosis	.316	.324
Rango	1.00	2.00
Mínimo	7.00	1.00
Máximo	236	3.00

a. Existen múltiples modas. Se muestra el valor más pequeño.

Tabla 29. Estadística comparativa entre iteraciones del factor *Facilidad de Aprendizaje*.

La comparativa de histogramas de los ítems para *Facilidad de Aprendizaje* de las iteraciones efectuadas se muestra a continuación.



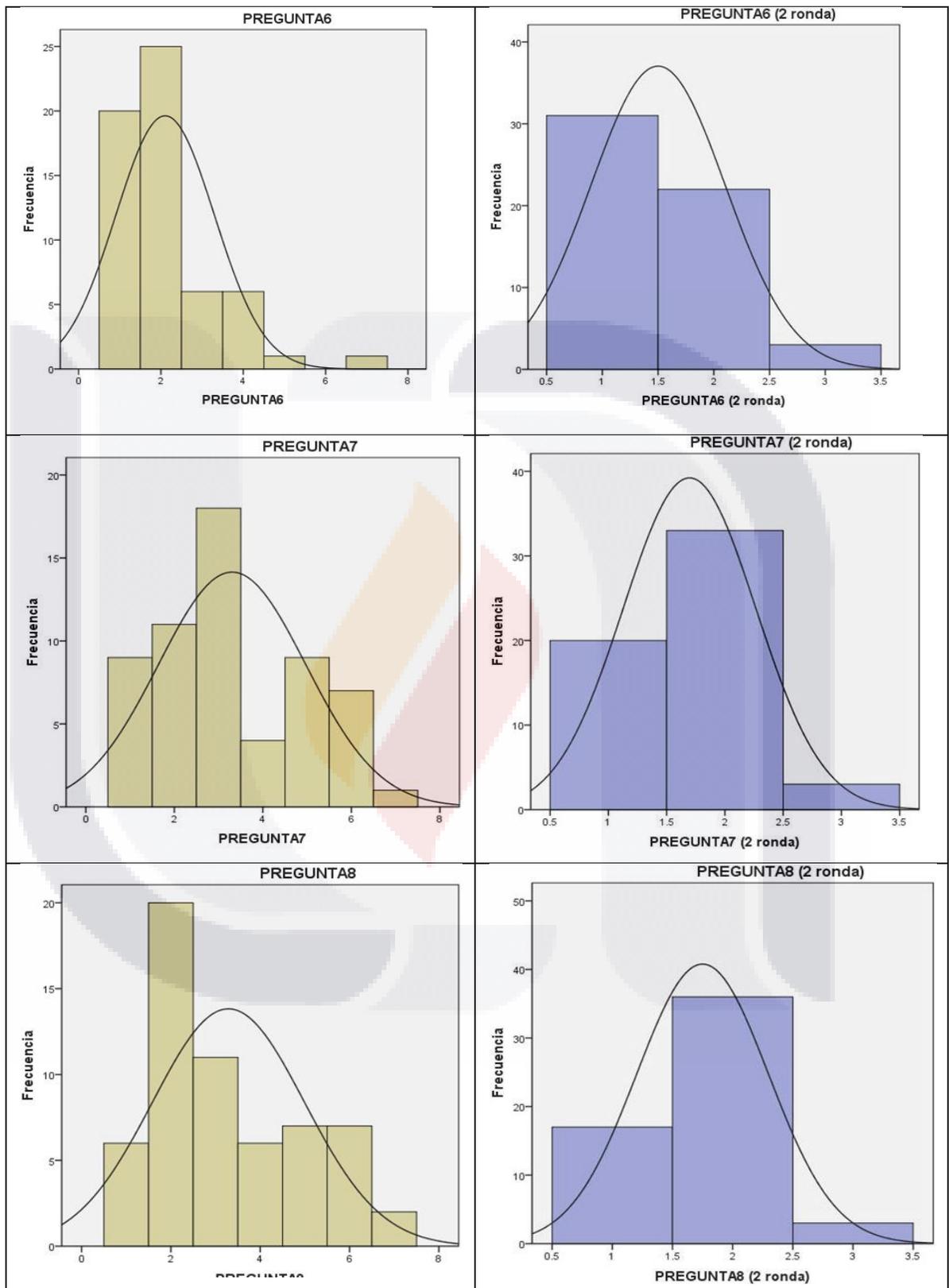


Figura 27. Comparativa de Histogramas entre iteraciones de *Facilidad de Aprendizaje*.

La prueba de muestras pareadas para *Facilidad de Aprendizaje* con la cual se aprueba o rechaza la hipótesis dio el siguiente resultado (ver tabla 30).

Items Facilidad de aprendiza je	DIFERENCIAS RELACIONADAS					t	gl	Sig. (bilat eral)
	Media	Desviación típ.	Error típ. de la medi a	95% Intervalo de confianza para la diferencia				
				Inferior	Superior			
Par5 Pregunta5 - Pregunta5 (2 ronda)	.893	1.485	.198	.495	1.291	4.498	55	.000
Par6 Pregunta6 - Pregunta6 (2 ronda)	.554	1.413	.189	.175	.932	2.931	55	.005
Par7 Pregunta7 - Pregunta7 (2 ronda)	1.589	1.724	.230	1.127	2.051	6.897	55	.000
Par8 Pregunta8 - Pregunta8 (2 ronda)	1.571	1.808	.242	1.087	2.056	6.505	55	.000

Tabla 30. Prueba de muestras pareadas factor *Facilidad de Aprendizaje*.

Para la primera versión del sistema la mayoría de los usuarios opinó que estaba algo de acuerdo con respecto a la calidad en la *Facilidad de Aprendizaje* del sistema evaluado, en la segunda versión la mayoría de los usuarios dijo estar Totalmente de acuerdo.

La calidad para *Facilidad de Aprendizaje* en la segunda evaluación fue de 1.6 (ver tabla 29) por ello de acuerdo a la escala utilizada se conoce que el factor *Facilidad de Aprendizaje* en la segunda versión cuenta con calidad muy buena.

Facilidad de Aprendizaje tiene una diferencia significativa ($p < = .001$) (ver Tabla 30), por lo tanto significa que la calidad en este factor tiene una medida muy importante desde el punto de vista del usuario final.

Esto confirma la hipótesis H1b de que en la nueva versión de la aplicación, el factor *Facilidad de Aprendizaje* mejoró significativamente.

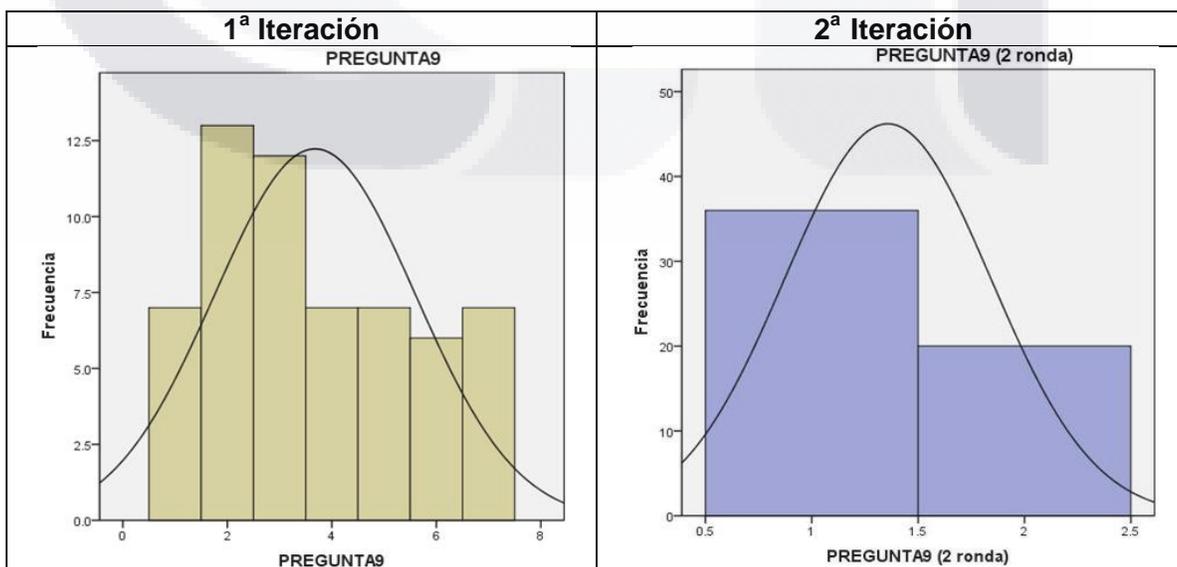
En el factor *Atractivo* se encontraron los siguientes datos en la segunda iteración del software evaluado, en la tabla 31 se pueden observar las comparativas de las estadísticas.

ESTADÍSTICAS	ITERACIONES	
	1 ^a	2 ^a
Casos validos	59	56
Casos perdidos	0	0
Media	3.5763	1.4821
Mediana	3.0000	1.0000
Moda	2.00	1.00
Desviación Estándar	1.96526	.54373
Varianza	3.862	.296
Asimetría	.372	.494
Error estándar de asimetría	.158	.163
Curtosis	-1.111	-.926
Error estándar de curtosis	.316	.324
Rango	6.00	2.00
Mínimo	1.00	1.00
Máximo	7.00	3.00

a. Existen múltiples modas. Se muestra el valor más pequeño.

Tabla 31. Estadística comparativa entre iteraciones del factor *Atractivo*.

La comparativa de los histogramas del factor *Atractivo* de las iteraciones realizadas para cada uno de sus ítems es como se muestra a continuación.



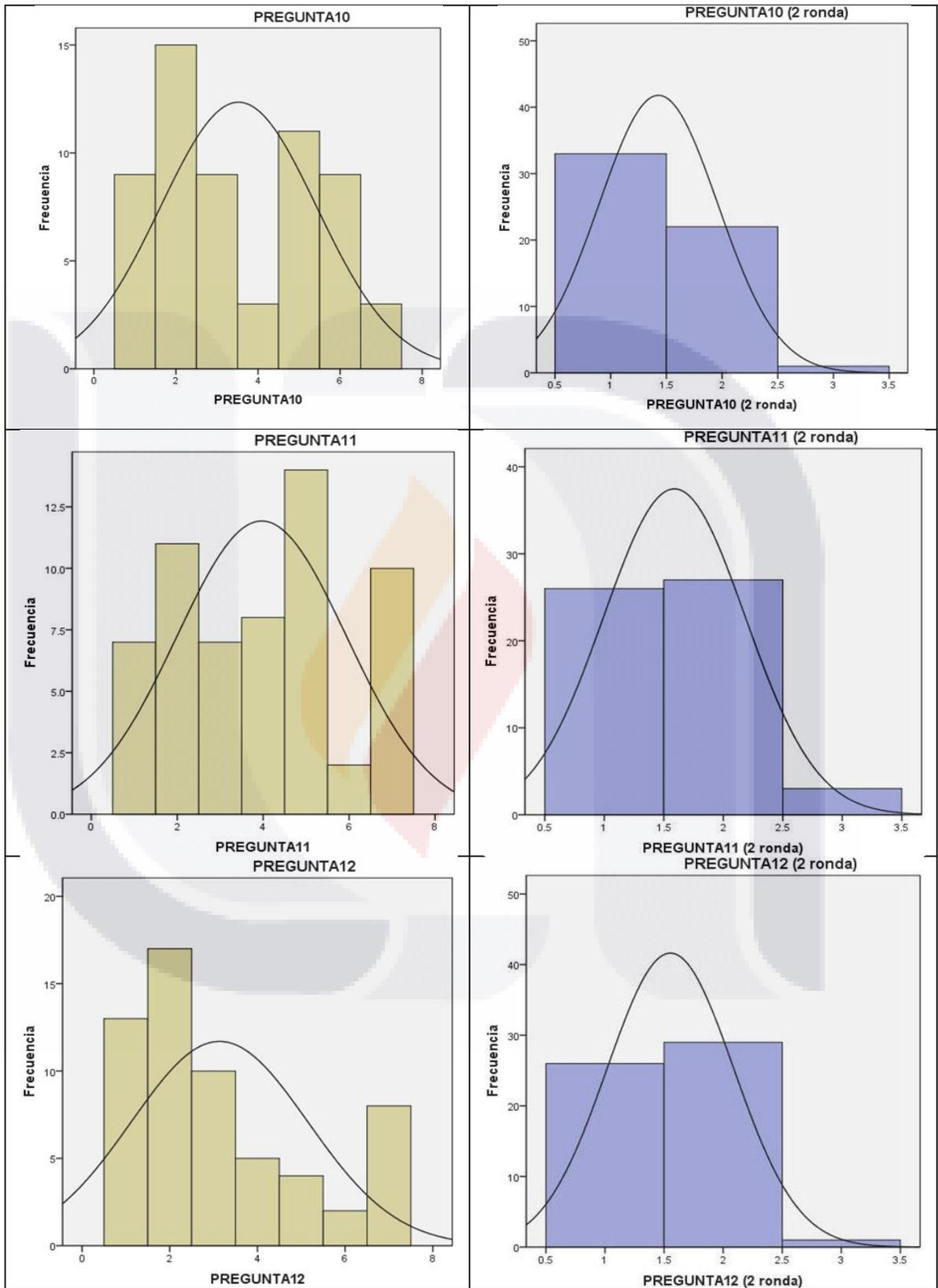


Figura 28. Comparativa de Histogramas entre iteraciones del *Atractivo*.

La prueba de muestras pareadas para *Atractivo* con la cual se aprueba o rechaza la hipótesis dio el siguiente resultado (ver tabla 32).

Items atractivo	DIFERENCIAS RELACIONADAS					t	gl	Sig. (bilateral)
	Media	Desviación típ.	Error típ. de la media	95% Intervalo de confianza para la diferencia				
				Inferior	Superior			
Par9 Pregunta9 - Pregunta9 (2 ronda)	2.268	2.076	.277	1.712	2.824	8.176	55	.000
Par10 Pregunta10 - Pregunta10 (2 ronda)	2.018	1.977	.264	1.488	2.547	7.638	55	.000
Par11 Pregunta11 - Pregunta11 (2 ronda)	2.393	2.077	.278	1.837	2.949	8.620	55	.000
Par12 Pregunta12 - Pregunta12 (2 ronda)	1.482	2.174	.291	.900	2.064	5.101	55	.000

Tabla 32. Prueba de muestras pareadas factor *Atractivo*.

Para la primera versión del sistema la mayoría de los usuarios opinó que estaba en una opinión neutral sobre la calidad del *Atractivo* del sistema, para la segunda versión la mayoría de los usuarios dijo estar Totalmente de acuerdo con la calidad en éste factor.

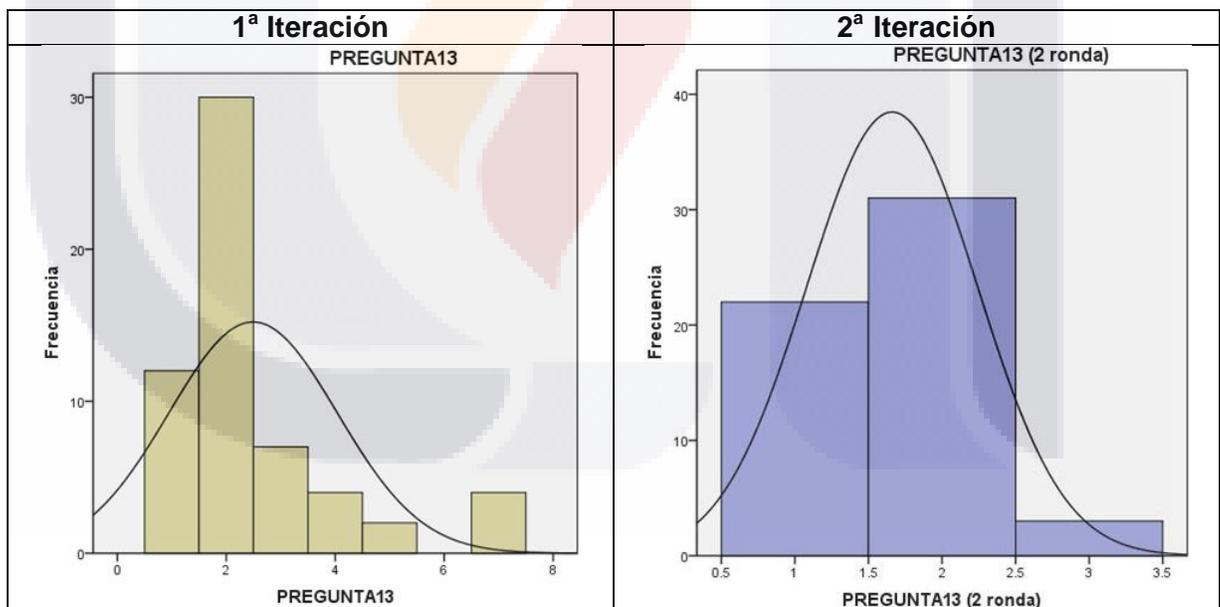
La calidad para el factor *Atractivo* en la segunda iteración fue de 1.4 (ver tabla 31) por lo tanto de acuerdo a la escala utilizada se puede decir que el factor *Atractivo* en la segunda versión del sistema cuenta con una calidad excelente. El factor *Atractivo* tiene una diferencia significativa ($p \leq .001$) (ver Tabla 32), por lo tanto significa que la calidad en este factor tiene una medida muy importante desde el punto de vista del usuario final. Esto confirma la hipótesis H1c de que en la nueva versión de la aplicación, el factor de calidad *Atractivo* mejoró significativamente.

Para el factor *Comprensibilidad* se encontraron los siguientes datos en la segunda iteración, en la Tabla 33 se pueden observar la comparativa de las estadísticas de sus iteraciones.

ESTADÍSTICAS	ITERACIONES	
	1 ^a	2 ^a
Casos validos	59	56
Casos perdidos	0	0
Media	2.7966	1.6027
Mediana	2.0000	2.0000
Moda	2.00	2.00
Desviación Estándar	1.56591	.56679
Varianza	2.452	.321
Asimetría	1.019	.409
Error estándar de asimetría	.158	.163
Curtosis	.461	.159
Error estándar de curtosis	.316	.324
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 33. Estadística comparativa entre iteraciones del factor *Comprensibilidad*.

La comparativa de los histogramas para *Comprensibilidad* de las iteraciones realizadas para cada uno de sus ítems es como se muestra a continuación.



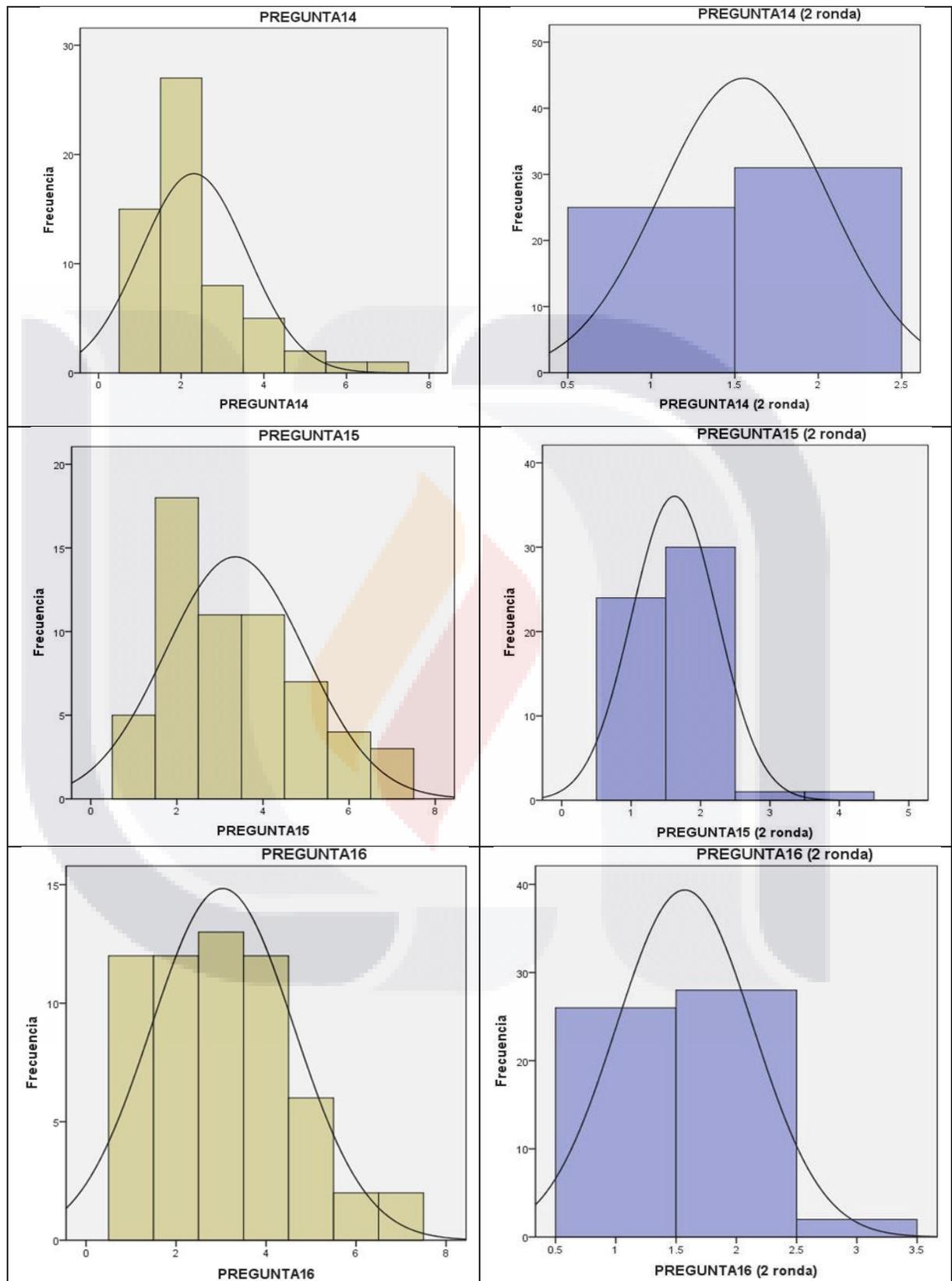


Figura 29. Comparativa de Histogramas entre iteraciones de *comprensibilidad*.

La prueba de muestras pareadas para *Comprensibilidad* con la cual se aprueba o rechaza la hipótesis dio el siguiente resultado (ver tabla 34).

Items comprensibilidad	DIFERENCIAS RELACIONADAS					t	gl	Sig. (bilateral)
	Media	Desviación típ.	Error típ. de la media	95% Intervalo de confianza para la diferencia				
				Inferior	Superior			
Par13 Pregunta13 - Pregunta13 (2 ronda)	.732	1.601	.214	.303	1.161	3.422	55	.001
Par14 Pregunta14 - Pregunta14 (2 ronda)	.643	1.197	.160	.322	.964	4.018	55	.000
Par15 Pregunta15 - Pregunta15 (2 ronda)	1.625	1.602	.214	1.196	2.054	7.591	55	.000
Par16 Pregunta16 - PregUNTA 16 (2 ronda)	1.429	1.704	.228	.972	1.885	6.273	55	.000

Tabla 34. Prueba de muestras pareadas factor *Comprensibilidad*.

Durante la primera evaluación del sistema la mayoría de los usuarios opinó que estaba algo de acuerdo sobre la calidad de la *Comprensibilidad* del sistema, para la segunda ronda la mayoría dijo estar De acuerdo con la calidad en éste factor.

La calidad para el factor *Comprensibilidad* en la segunda iteración fue de 1.6 (ver Tabla 33) por lo tanto de acuerdo a la escala utilizada se puede decir que el factor *Comprensibilidad* en la segunda iteración cuenta con una calidad muy buena.

El factor *Comprensibilidad* tiene una diferencia significativa ($p \leq .001$) (ver Tabla 34), por lo tanto significa que la calidad en este factor tiene una medida muy importante desde el punto de vista del usuario final. Esto confirma la hipótesis H1d de que en la nueva versión de la aplicación, el factor de calidad *Comprensibilidad* mejoró significativamente.

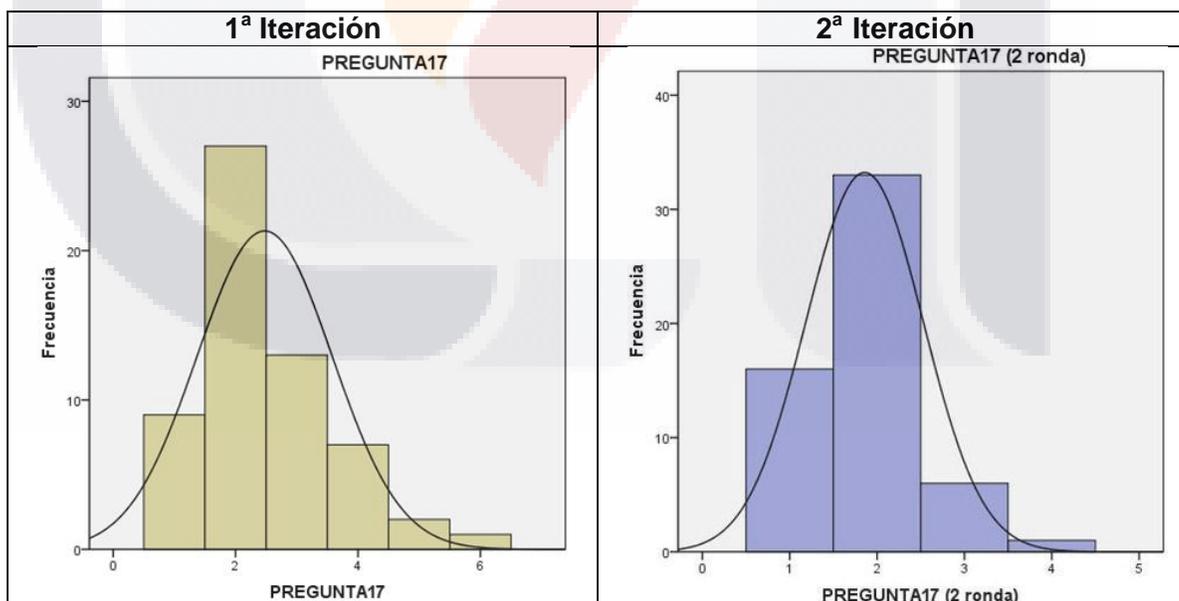
En la medición de *Conformidad* se encontraron los siguientes datos en la segunda evaluación, en la tabla 35 se pueden observar la comparativa de las estadísticas de cada iteración.

ESTADÍSTICAS	ITERACIONES	
	1ª	2ª
Casos validos	59	56
Casos perdidos	0	0
Media	3.0226	1.6964
Mediana	3.0000	2.0000
Moda	2.00	2.00
Desviación Estándar	1.31838	.69004
Varianza	1.738	.476
Asimetría	6.00	.814
Error estándar de asimetría	1.00	.187
Curtosis	7.00	.756
Error estándar de curtosis	177	.373
Rango	0	3.00
Mínimo	3.0226	1.00
Máximo	3.0000	4.00

a. Existen múltiples modas. Se muestra el valor más pequeño.

Tabla 35. Estadística comparativa entre iteraciones del factor *Conformidad*.

La comparativa de los histogramas para *Conformidad* de las iteraciones realizadas se muestra a continuación para cada uno de sus ítems.



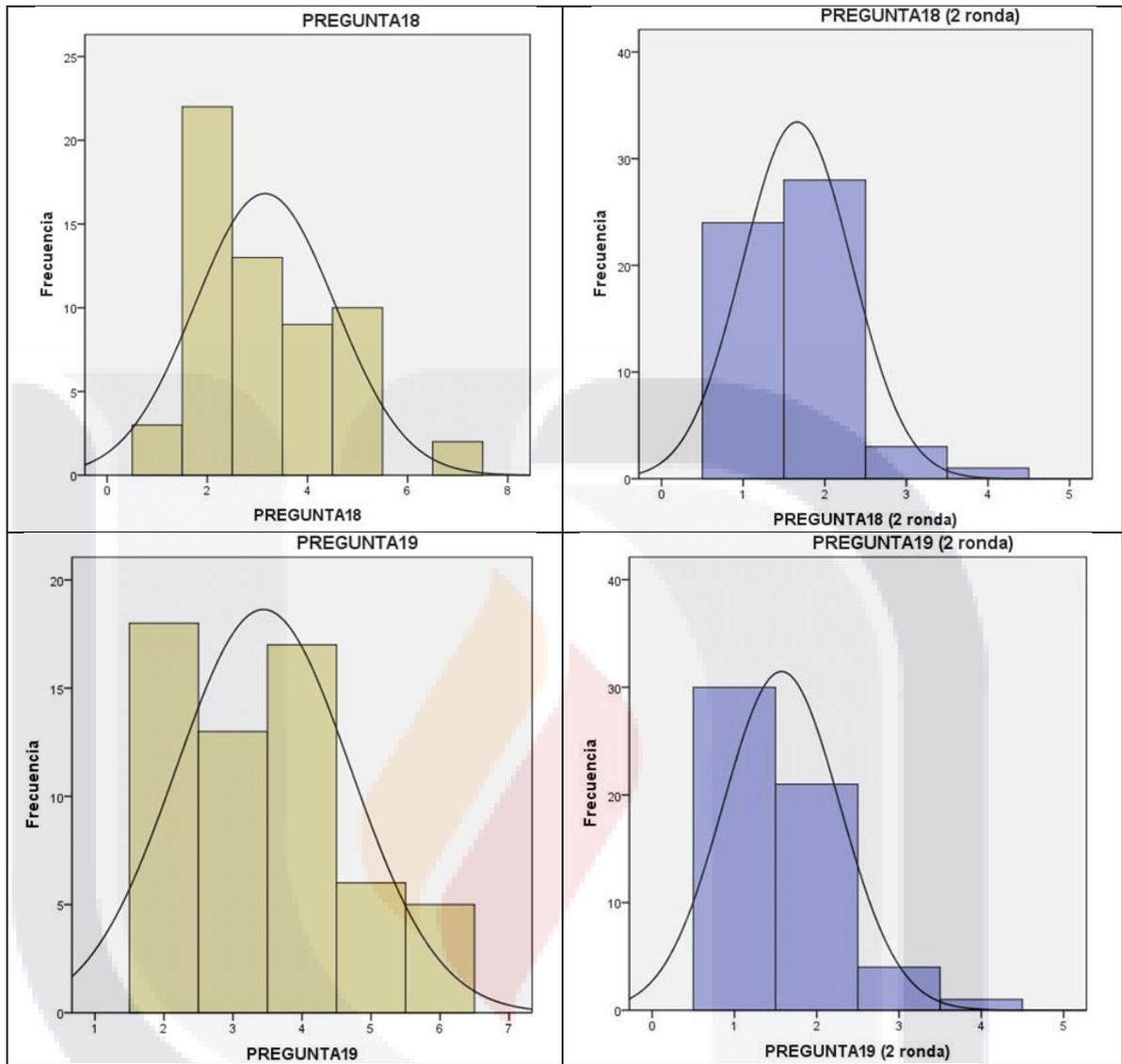


Figura 30. Comparativa de Histogramas entre iteraciones de *Conformidad*.

La prueba de muestras pareadas para *Conformidad* con la cual se aprueba o rechaza la hipótesis dio el siguiente resultado (ver Tabla 36).

Items conformidad	DIFERENCIAS RELACIONADAS					t	gl	Sig. (bilateral)
	Media	Desviación típ.	Error típ. de la media	95% Intervalo de confianza para la diferencia				
				Inferior	Superior			
Par17 PREGunta17 – Pregunta17 (2 ronda)	.589	1.304	.174	.240	.939	3.381	55	.001
Par18 Pregunta18 – Pregunta18 (2 ronda)	1.500	1.549	.207	1.085	1.915	7.246	55	.000
Par19 Pregunta19 – Pregunta19 (2 ronda)	1.857	1.470	.196	1.463	2.251	9.454	55	.000

Tabla 36. Prueba de muestras pareadas factor *Conformidad*.

Para la primera versión del sistema los usuarios opinaron que estaban Algo de acuerdo sobre la calidad de la *Conformidad* del sistema, para la segunda versión la mayoría dijo estar De acuerdo con la calidad en éste.

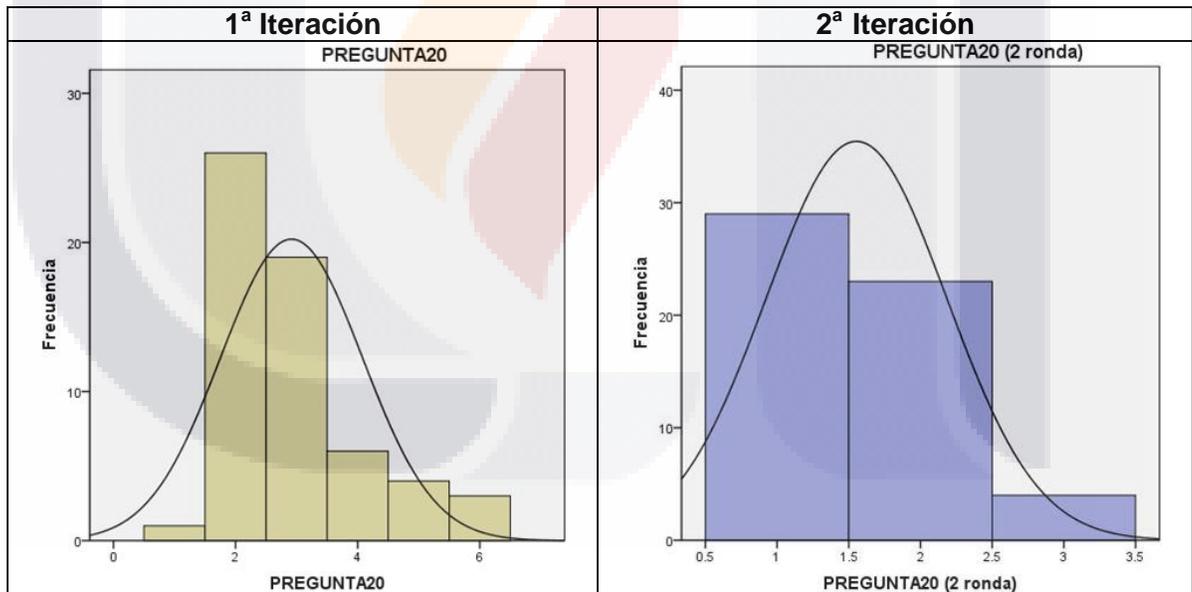
La calidad para el factor *Conformidad* en la segunda iteración fue de 1.7 (ver Tabla 35) por lo tanto de acuerdo a la escala utilizada se puede decir que el factor *Conformidad* en la segunda iteración cuenta con una Muy buena calidad. El factor *Conformidad* tiene una diferencia significativa ($p \leq .001$) (ver Tabla 36), por lo tanto, significa que la calidad en este factor tiene una medida muy importante desde el punto de vista del usuario final. Esto confirma la hipótesis H2a de que en la nueva versión de la aplicación, el factor de calidad *Conformidad* mejoró significativamente.

Para el factor *Exactitud* se encontraron los siguientes datos en la segunda iteración, en la tabla 37 se pueden observar las comparativas de las estadísticas de cada iteración.

ESTADÍSTICAS	ITERACIONES	
	1ª	2ª
Casos validos	59	56
Casos perdidos	0	0
Media	3.0734	1.6190
Mediana	3.0000	2.0000
Moda	2.00	2.00
Desviación Estándar	1.24327	.62684
Varianza	1.546	.393
Asimetría	.882	.647
Error estándar de asimetría	.183	.187
Curtosis	.131	.217
Error estándar de curtosis	.363	.373
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 37. Estadística comparativa entre iteraciones del factor *Exactitud*.

La comparativa de los histogramas para *Exactitud* de las iteraciones realizadas se muestra a continuación para cada uno de sus ítems.



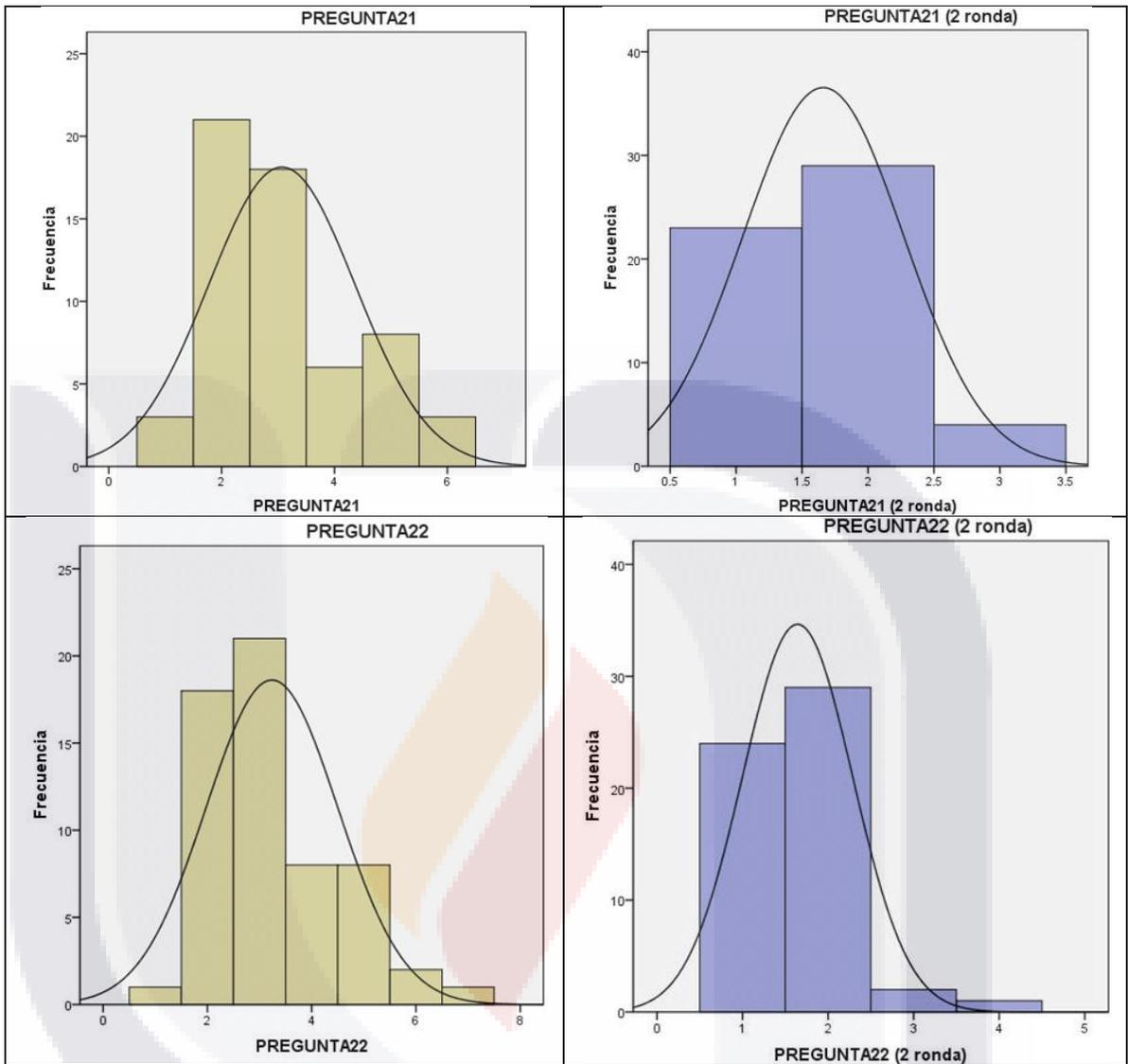


Figura 31. Comparativa de Histogramas entre iteraciones de *Exactitud*.

La prueba de muestras pareadas para *Exactitud* con la cual se aprueba o rechaza la hipótesis dio el siguiente resultado (ver Tabla 38).

Items exactitud	DIFERENCIAS RELACIONADAS					t	gl	Sig. (bilateral)
	Media	Desviación típ.	Error típ. de la media	95% Intervalo de confianza para la diferencia				
				Inferior	Superior			
Par20 Pregunta20 - Pregunta20 (2 ronda)	1.357	1.271	.170	1.017	1.698	7.990	55	.000
Par21 Pregunta21 - Pregunta21 (2 ronda)	1.429	1.386	.185	1.057	1.800	7.711	55	.000
Par22 Pregunta22 - Pregunta22 (2 ronda)	1.607	1.371	.183	1.240	1.974	8.773	55	.000

Tabla 38. Prueba de muestras pareadas factor *Exactitud*.

Para la primera versión del sistema la mayoría de los usuarios opinó que estaba algo de acuerdo sobre la calidad de la *Exactitud* del sistema, para la segunda versión la mayoría de los usuarios dijo estar De acuerdo con la calidad en *Exactitud*. La calidad para el factor *Exactitud* en la segunda iteración fue de 1.6 (ver Tabla 37) por lo tanto de acuerdo a la escala utilizada se puede decir que el factor *Exactitud* en la segunda iteración cuenta con una calidad muy buena.

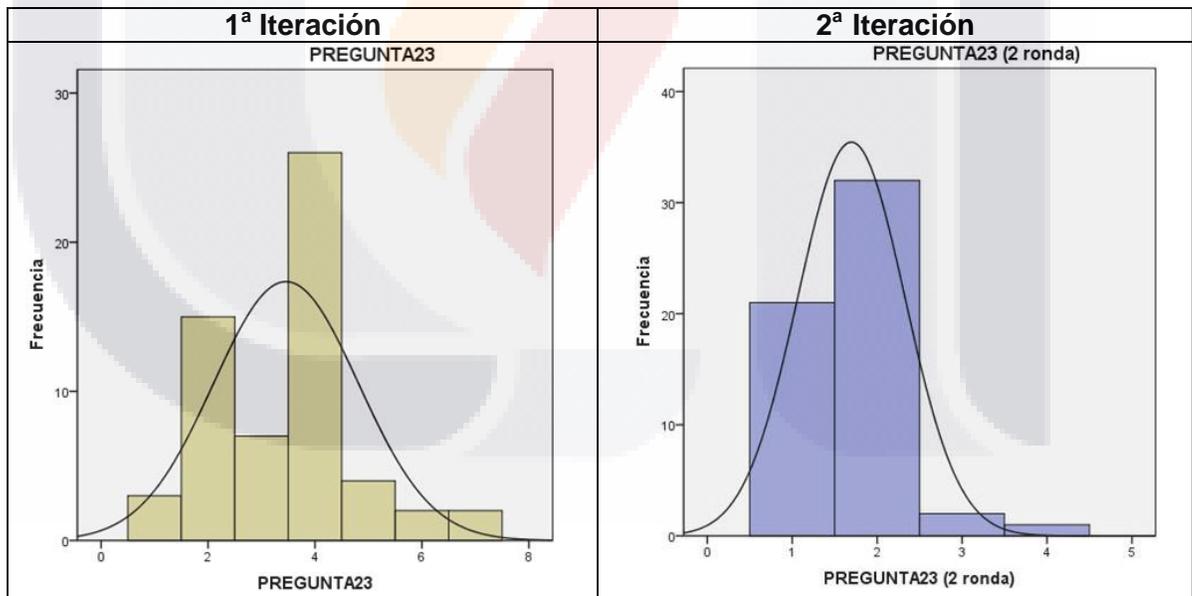
El factor *Exactitud* tiene una diferencia significativa ($p \leq .001$) (ver Tabla 38), por lo tanto significa que la calidad en *Exactitud* tiene una medida muy importante desde el punto de vista del usuario final. Esto confirma la hipótesis H2b de que en la nueva versión de la aplicación, el factor de calidad *Exactitud* mejoró significativamente.

Para *Madurez* se encontraron los siguientes datos en la segunda iteración, en la tabla 39 se pueden observar las comparativas de las estadísticas de cada iteración.

ESTADÍSTICAS	ITERACIONES	
	1 ^a	2 ^a
Casos validos	59	56
Casos perdidos	0	0
Media	3.0734	1.6190
Mediana	3.0000	2.0000
Moda	2.00	2.00
Desviación Estándar	1.24327	.62684
Varianza	1.546	.393
Asimetría	.882	.647
Error estándar de asimetría	.183	.187
Curtosis	.131	.217
Error estándar de curtosis	.363	.373
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 39. Estadística comparativa entre iteraciones del factor *Madurez*.

La comparativa de los histogramas para *Madurez* de las iteraciones que se realizaron en el sistema se muestra a continuación para cada uno de sus ítems.



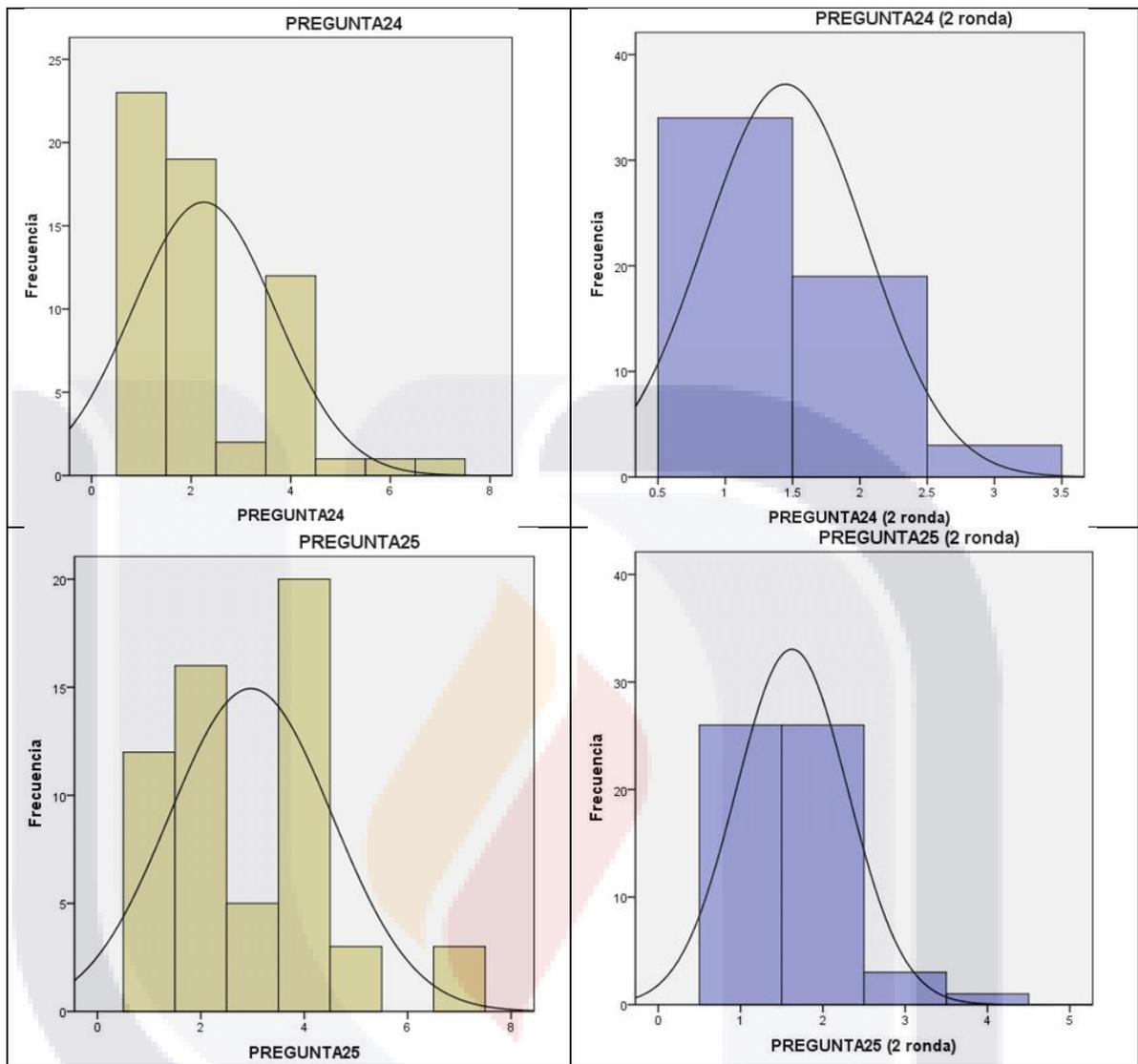


Figura 32. Comparativa de Histogramas entre iteraciones de *Madurez*.

La prueba de muestras pareadas para *Madurez* con la cual se aprueba o rechaza la hipótesis dio el siguiente resultado (ver Tabla 40).

Items madurez	DIFERENCIAS RELACIONADAS					t	gl	Sig. (bilateral)
	Media	Desviación típ.	Error típ. de la media	95% Intervalo de confianza para la diferencia				
				Inferior	Superior			
Par23 Pregunta23 - Pregunta23 (2 ronda)	1.750	1.665	.223	1.304	2.196	7.865	55	.000
Par24 Pregunta24 - Pregunta24 (2 ronda)	.804	1.577	.211	.381	1.226	3.812	55	.000
Par25 Pregunta25 - Pregunta25 (2 ronda)	1.339	1.643	.220	.899	1.779	6.098	55	.000

Tabla 40. Prueba de muestras pareadas factor *Madurez*.

Para la primera versión del sistema la mayoría de los usuarios opinó que estaba algo de acuerdo sobre la calidad del factor *Madurez* del sistema evaluado, para la segunda iteración la mayoría dijo estar De acuerdo con la calidad encontrada.

La calidad para el factor *Madurez* en la segunda iteración fue de 1.6 (ver Tabla 39) por lo tanto acorde a la escala utilizada el factor *Madurez* en la segunda ronda cuenta con una calidad muy buena.

El factor *Madurez* tiene una diferencia significativa ($p \leq .001$) (ver Tabla 40), por lo tanto significa que la calidad en *tal factor* posee una medida importante desde el punto de vista del usuario final. Esto confirma la hipótesis H3a de que en la nueva versión del sistema, el factor *Madurez* mejoró significativamente.

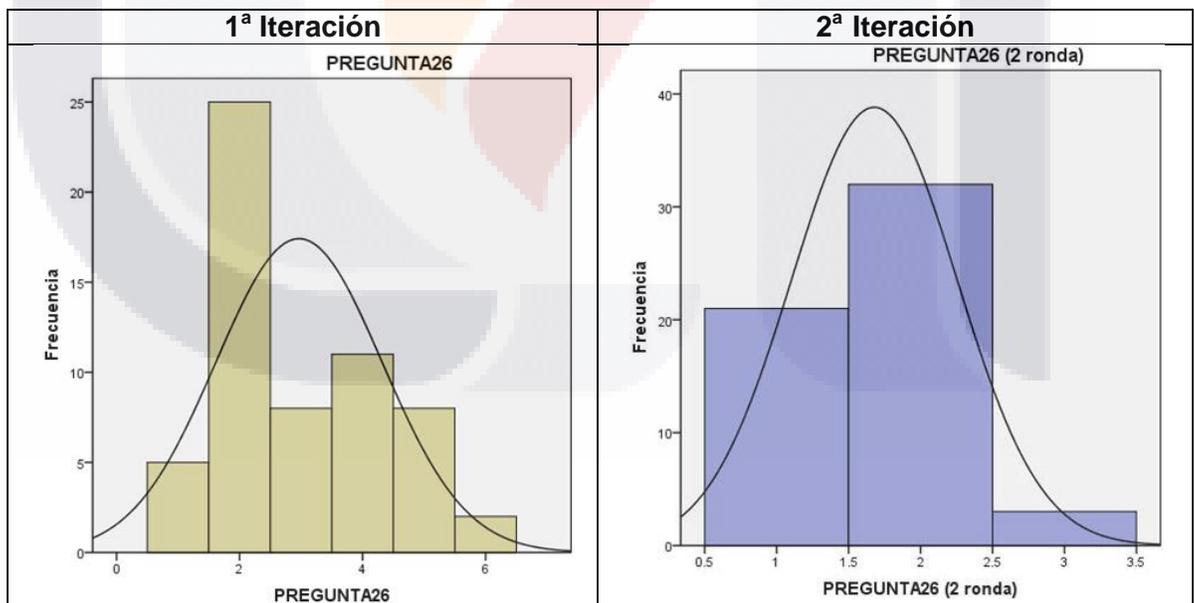
Para *Tolerancia a Fallas* se encontraron los siguientes datos en la segunda ronda, en la tabla 41 se observan las comparativas de las estadísticas de cada iteración.

ESTADÍSTICAS	ITERACIONES	
	1ª	2ª
Casos validos	59	56
Casos perdidos	0	0
Media	2.8644	1.5714
Mediana	3.0000	2.0000
Moda	2.00	2.00
Desviación Estándar	1.39142	.58496
Varianza	1.936	.342
Asimetría	.655	.627
Error estándar de asimetría	.183	.187
Curtosis	6.00	.489
Error estándar de curtosis	1.00	.373
Rango	7.00	3.00
Mínimo	177	1.00
Máximo	0	4.00

a. Existen múltiples modas. Se muestra el valor más pequeño.

Tabla 41. Estadística comparativa entre iteraciones del factor *Tolerancia a Fallas*.

La comparativa de los histogramas de los ítems del factor *Tolerancia a Fallas* de las iteraciones realizadas se muestran a continuación para cada uno de sus ítems.



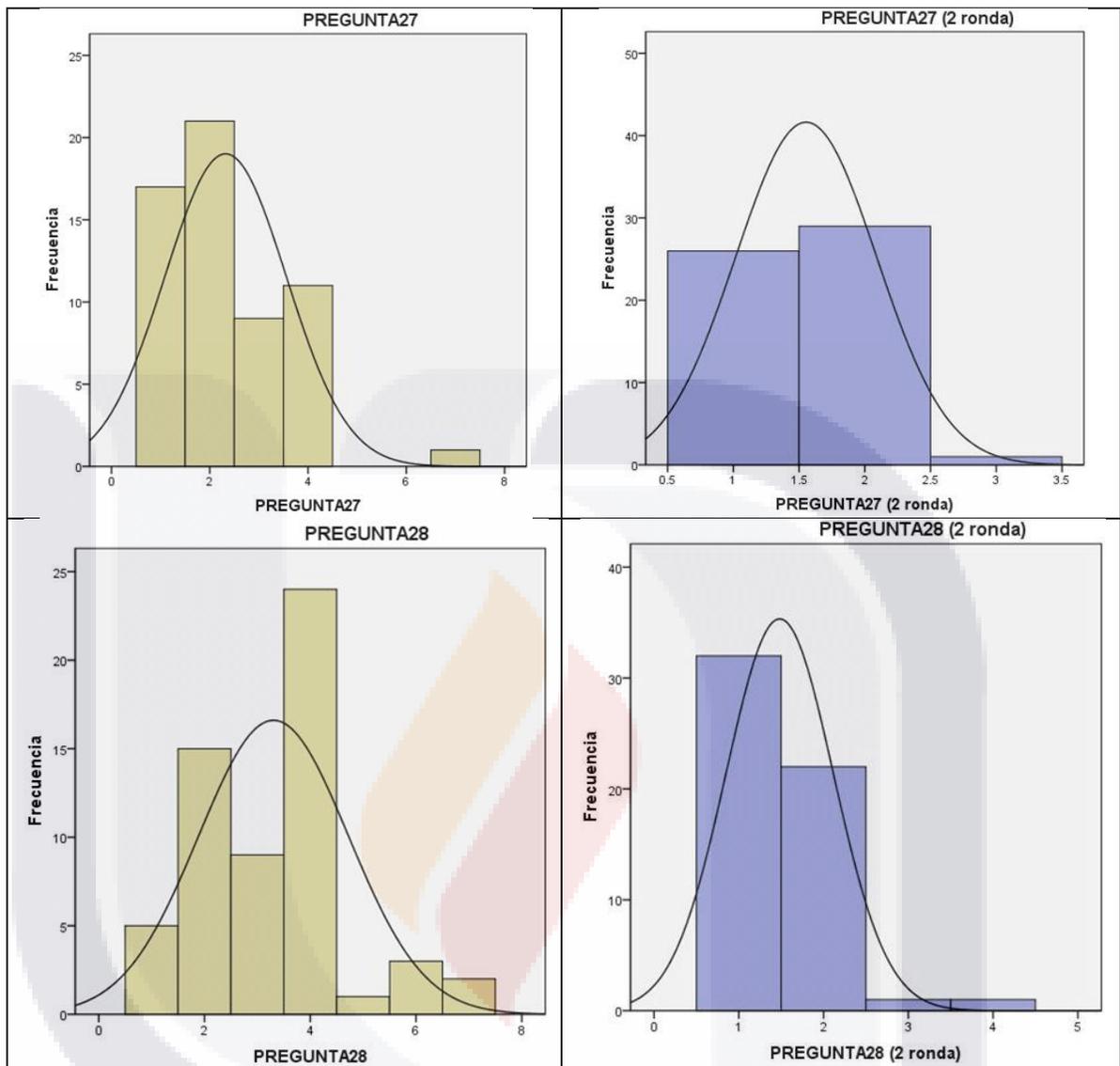


Figura 33. Comparativa de Histogramas entre iteraciones de *Tolerancia a Fallas*.

La prueba de muestras pareadas para *Tolerancia a Fallas* con la cual se aprueba o rechaza la hipótesis dio el siguiente resultado (ver Tabla 42).

Items tolerancia a fallas	DIFERENCIAS RELACIONADAS					t	gl	Sig. (bilateral)
	Media	Desviación típ.	Error típ. de la media	95% Intervalo de confianza para la diferencia				
				Inferior	Superior			
Par26 Pregunta26 – Pregunta26 (2 ronda)	1.232	1.465	.196	.840	1.624	6.295	55	.000
Par27 Pregunta27 – Pregunta27 (2 ronda)	.768	1.427	.191	.386	1.150	4.027	55	.000
Par28 Pregunta28 – Pregunta28 (2 ronda)	1.786	1.569	.210	1.365	2.206	8.516	55	.000

Tabla 42. Prueba de muestras pareadas factor *Tolerancia a Fallas*.

Durante la primera iteración los usuarios opinaron que estaban algo de acuerdo con la calidad respecto a la *Tolerancia a Fallas*, para la segunda ronda la mayoría dijo estar Totalmente De acuerdo con la calidad encontrada.

La calidad para el factor *Tolerancia a Fallas* en la segunda iteración fue de 1.5 (ver Tabla 41) por lo tanto acorde a la escala utilizada el factor *Tolerancia a Fallas* en la segunda ronda cuenta con una excelente calidad.

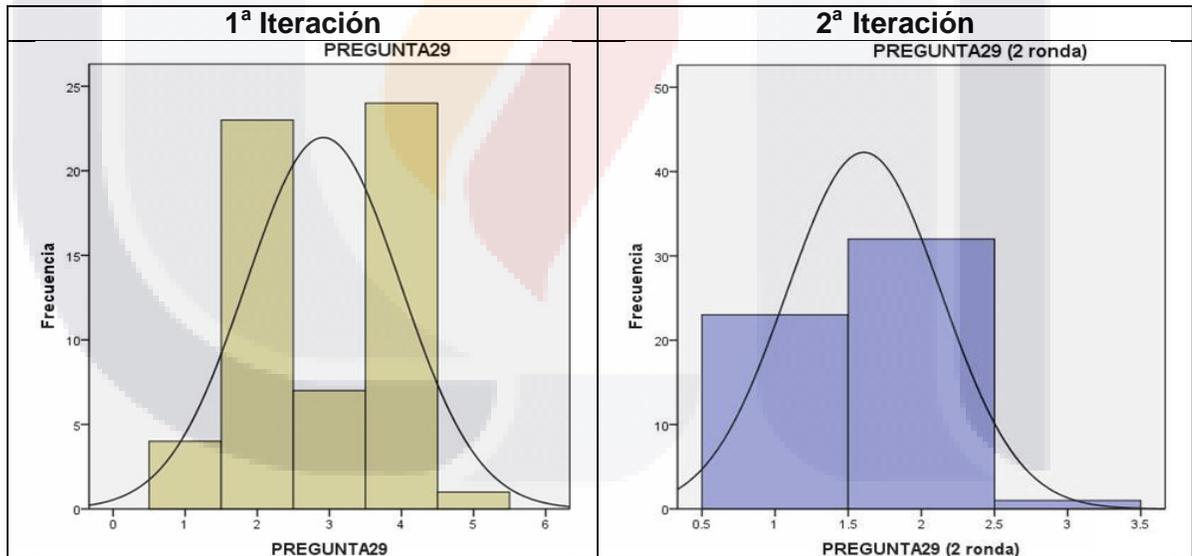
El factor de *Tolerancia a Fallas* tiene una diferencia significativa ($p \leq .001$) (ver Tabla 42), por ello significa que la calidad en éste factor tiene una medida importante desde el punto de vista del usuario final. Esto confirma la hipótesis H3b de que en la nueva versión del sistema, el factor *Tolerancia a Fallas* mejoró significativamente.

Para la *Recuperabilidad* se encontraron los siguientes datos en la segunda ronda, en la tabla 43 se observan las comparativas de las estadísticas de cada iteración.

ESTADÍSTICAS	ITERACIONES	
	1ª	2ª
Casos validos	59	56
Casos perdidos	0	0
Media	2.8701	1.5714
Mediana	3.0000	2.0000
Moda	4.00	2.00
Desviación Estándar	1.11806	.58496
Varianza	1.250	.342
Asimetría	.136	.627
Error estándar de asimetría	.183	.187
Curtosis	-.471	.489
Error estándar de curtosis	.363	.373
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 43. Estadística comparativa entre iteraciones del factor *Recuperabilidad*.

La comparativa de histogramas para los ítems del factor *Recuperabilidad* de las iteraciones realizadas se muestran a continuación.



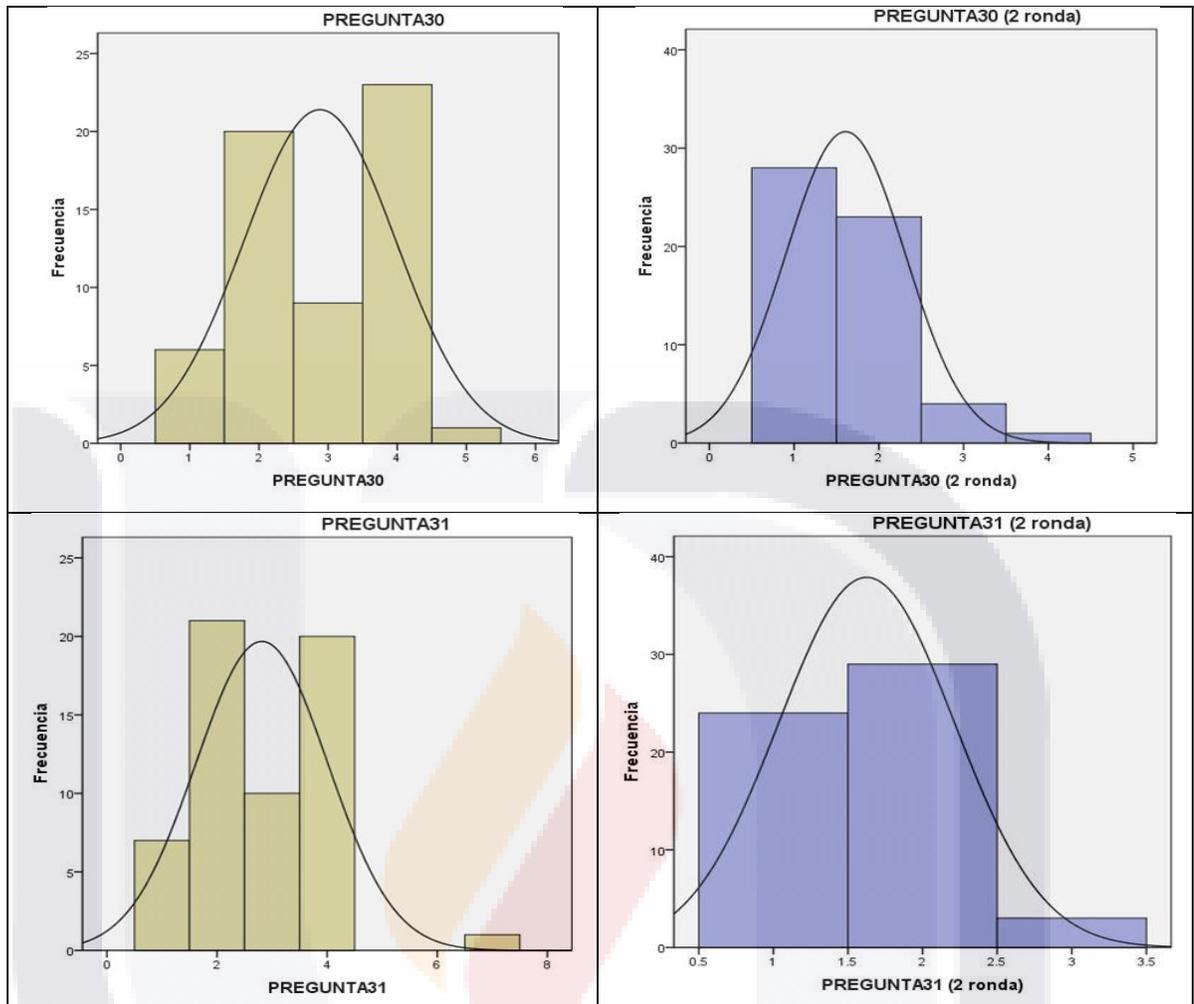


Figura 34. Comparativa de Histogramas entre iteraciones de *Recuperabilidad*.

La prueba de muestras pareadas para *Recuperabilidad* con la cual se aprueba o rechaza la hipótesis dio el siguiente resultado (ver Tabla 44).

Items recuperabilidad	DIFERENCIAS RELACIONADAS					t	gl	Sig. (bilateral)
	Media	Desviación típ.	Error típ. de la media	95% Intervalo de confianza para la diferencia				
				Inferior	Superior			
Par26 Pregunta26 – Pregunta26 (2 ronda)	1.232	1.465	.196	.840	1.624	6.295	55	.000
Par27 Pregunta27 – Pregunta27 (2 ronda)	.768	1.427	.191	.386	1.150	4.027	55	.000
Par28 Pregunta28 – Pregunta28 (2 ronda)	1.786	1.569	.210	1.365	2.206	8.516	55	.000

Tabla 44. Prueba de muestras pareadas factor *Recuperabilidad*.

En la primera iteración los usuarios opinaron que se encontraban Algo de acuerdo con la calidad respecto a la *Recuperabilidad* del sistema, para la segunda ronda la mayoría dijo estar Totalmente de acuerdo con la calidad encontrada.

La calidad para el factor *Recuperabilidad* en la segunda iteración fue de 1.5 (ver Tabla 43) por lo tanto según la escala utilizada éste factor en la segunda evaluación cuenta con excelente calidad.

El factor *Recuperabilidad* tiene una diferencia significativa ($p \leq .001$) (ver Tabla 44), la calidad en éste tiene una medida importante por parte del usuario final. Esto confirma la hipótesis H3c de que en la nueva versión del sistema, el factor de *Recuperabilidad* mejoró de forma significativa.

Para el *Tiempo de Respuesta* se encontraron los siguientes datos en la segunda ronda, en la tabla 45 se observan las comparativas de las estadísticas.

ESTADÍSTICAS	ITERACIONES	
	1 ^a	2 ^a
Casos validos	59	56
Casos perdidos	0	0
Media	2.4237	1.5238
Mediana	2.0000	1.0000
Moda	2.00	1.00
Desviación Estándar	1.23679	.59892
Varianza	1.530	.359
Asimetría	1.225	.838
Error estándar de asimetría	.183	.187
Curtosis	1.877	.634
Error estándar de curtosis	.363	.373
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 45. Estadística comparativa entre iteraciones para el factor de *Tiempo de Respuesta*.

Los histogramas para los ítems del factor *Tiempo de Respuesta* de las iteraciones realizadas se muestran a continuación en una comparativa.

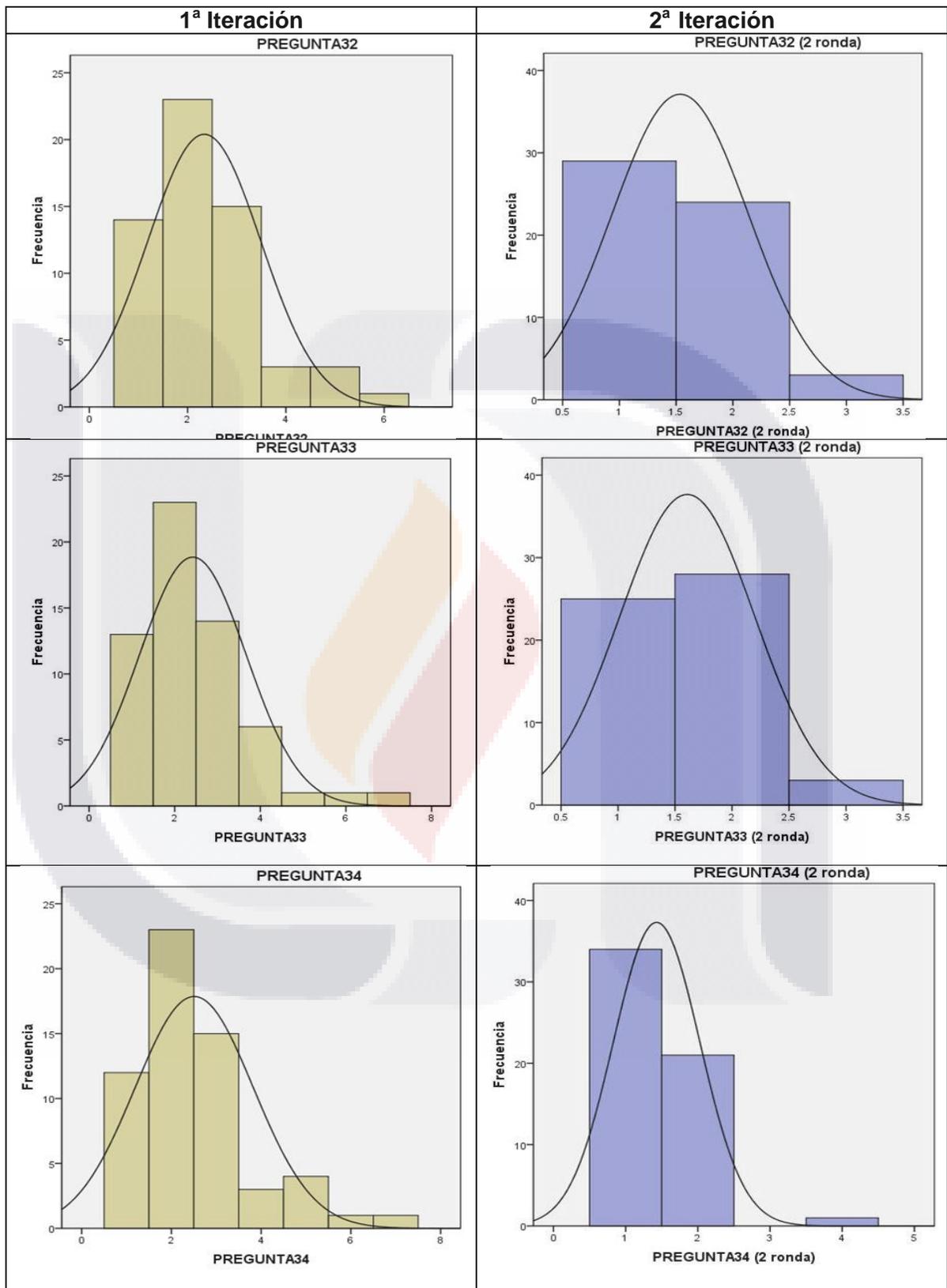


Figura 35. Comparativa de Histogramas entre iteraciones de *Tiempo de Respuesta*.

La prueba de muestras pareadas para *Tiempo de Respuesta* con la cual se aprueba o rechaza la hipótesis dio el siguiente resultado (ver Tabla 46).

Items tiempo de respuesta	DIFERENCIAS RELACIONADAS					t	gl	Sig. (bilateral)
	Media	Desviación típ.	Error típ. de la media	95% Intervalo de confianza para la diferencia				
				Inferior	Superior			
Par26 Pregunta26 – Pregunta26 (2 Ronda)	.804	1.271	.170	.463	1.144	4.731	55	.000
Par27 Pregunta27 – Pregunta27 (2 ronda)	.839	1.424	.190	.458	1.221	4.410	55	.000
Par28 Pregunta28 – Pregunta28 (2 ronda)	1.071	1.450	.194	.683	1.460	5.528	55	.000

Tabla 46. Prueba de muestras pareadas factor *Tiempo de Respuesta*.

En la primera iteración la mayoría de los usuarios se encontraban De acuerdo con la calidad respecto al *Tiempo de Respuesta* del sistema desarrollado, para la segunda evaluación de éste la mayoría de usuarios dijo estar Totalmente de acuerdo con la calidad encontrada.

La calidad para el factor *Tiempo de Respuesta* en la segunda iteración fue de 1.5 (ver Tabla 45) por lo tanto según la escala utilizada, éste factor en la segunda evaluación cuenta con excelente calidad.

El *Tiempo de Respuesta* tiene una diferencia significativa ($p \leq .001$) (ver Tabla 46), por ello significa que la calidad en éste tiene una medida importante desde el punto de vista del usuario final. Esto confirma la hipótesis H4a de que en la nueva versión del sistema, el factor de calidad *Tiempo de Respuesta* mejoró de forma significativa.

5.2 Calculo de Métricas en el Desarrollo

Las métricas en desarrollo son mediciones para el software. Su uso esencial es ayudarnos a planear y predecir el desarrollo del software. La calidad se puede controlar de una mejor forma si el software se mide (Li 2002).

La calidad de un producto de software es determinada por varios atributos de calidad, numerosas medidas cuantitativas son utilizadas en el ciclo de vida del sistema, cada medida es definida para cuantificar uno o varios atributos de calidad (Kafura y Canning 1985).

Los datos recolectados en cada medición facilitan el entendimiento, evaluación y control de cada factor de calidad medido (Honglei,Wei et al. 2010).

A continuación se muestran que factores fueron medidos para el lado del desarrollador, así como la forma en que se midieron.

5.2.1 Facilidad de análisis.

La *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 (ISO 2001).

A pesar de que no existen muchas métricas conocidas en este ámbito, es muy significativo para el mantenimiento de un sistema la facilidad que los desarrolladores tengan para analizar el código del programa a modificar, ello depende en gran medida de lo bien documentado que el sistema se encuentre (Sicilia y De la Morena 2009).

El proceso de software es la secuencia de pasos requeridos para desarrollar o mantener el software. En este sentido, el mantener buenas prácticas durante el proceso de software ayuda de gran manera a realizar este proceso de una manera más rápida (Humphrey 2005).

Existen algunas metodologías que ayudan a definir una serie de pasos para desempeñar buenas prácticas en el desarrollo del software, entre las más utilizadas actualmente se encuentran el Proceso Personal de Software (PSP) y el Proceso en Equipo de Software (TSP).

PSP es un conjunto de prácticas disciplinadas para la gestión del tiempo y mejora de la productividad personal de los programadores o ingenieros de software, en tareas de desarrollo y mantenimiento de sistemas. En esta misma sintonía TSP proporciona un marco de trabajo de procesos definidos que está diseñado para ayudar a equipos de desarrollo a organizar y producir proyectos de software de gran escala (Humphrey 2000).

Tanto PSP como TSP establecen entre sus prácticas: como evitar defectos, como usar revisiones eficientes, como identificar deficiencias del software (Prechelt y Unger 2000). Para lograr esto algunas de las buenas prácticas que se utilizan son: estandarización en comentarios, ideas claras y cortas en comentarios, nombre de variables y métodos acorde a la función de cada uno.

Dentro de la importancia de los comentarios en el código fuente, Sicilia (2008) propone una métrica llamada densidad de comentarios en el código, la cual consiste en hacer un conteo de los comentarios en relación al número de líneas de código, cuanto mejor sean los comentarios mayor es la mantenibilidad del software afirma el autor. Por lo anterior para definir la facilidad de análisis se tomaron en cuenta las siguientes características:

1.- *Nombres de variables y métodos.*- Los nombres de variables y métodos deben tener relación a su uso y dar una idea al desarrollador del empleo de éstas. Se considera bueno si el nombre da una idea de que actividad realiza la variable o método y malo si el nombre no tiene relación alguna al funcionamiento.

2.- *Estructura del Comentario.*- El comentario debe contemplar cierta información, como quien realizó el código, en qué fecha fue la última modificación, o porque se realizó alguna modificación. Se considera bueno si la información proporcionada es

correcta y malo si falta información, o existen modificaciones comentadas que omitan el porqué se realizaron o en qué fecha.

3.- *Calidad del Comentario.*- Los comentarios deben ser ideas entendibles y simples. Se considera bueno si el comentario da a entender al desarrollador lo que se establece en el código, y malo si se utilizan palabras muy técnicas o difíciles de entender, y el desarrollador no tenga una idea clara de lo que se quiere informar.

4.- *Ubicación del comentario.*- Se debe comentar cada método, función, y parte de código que realice ciertas operaciones para conocer por qué y para que se hacen. Se considera bueno si los comentarios son suficientes, malo si son escasos y necesarios para entender ciertos apartados de código que no fueron comentados.

Posteriormente, se dividió el código del sistema en 4 partes y se entregó cada una de ellas a un programador diferente, luego se les pidió que evaluaran cada característica con una escala del 1 al 10, siendo 10 la calificación más alta.

Con el fin de evaluar el resultado obtenido y tomar decisiones se consideraron los siguientes resultados para establecer un resultado de excelente a malo (ver Tabla 47).

Valor	Interpretación Métrica
0.90 – 1.0	Excelente
0.80 - 0.90	Buena
0.70 – 0.80	Minima Aceptable
Menor 0.70	Mala

Tabla 47. Interpretación del resultado de la métrica

5.2.1.1 Primera iteración

Para la primera versión del sistema se encontraron los siguientes resultados para la *Facilidad de Análisis*.

Los resultados para la primera versión del sistema fueron los siguientes (ver Tabla 48):

Característica	P1	P2	P3	P4
<i>n.v.</i>	6	7	8	7
<i>e.c.</i>	6	8	7	8
<i>c.c.</i>	8	9	8	8
<i>u.c.</i>	6	7	7	6

Tabla 48. Evaluación de las características por 4 desarrolladores (1a iteración).

Donde:

n.v.= Nombres de variables y métodos.

e.c.= Estructura del Comentario.

c.c. = Calidad del Comentario.

u.c. = Ubicación de comentarios.

Por lo tanto el cálculo de la *Facilidad de Análisis* es de la siguiente forma:

$$\alpha = \frac{\sum n.v.}{T.p.}$$

$$\beta = \frac{\sum e.c.}{T.p.}$$

$$\gamma = \frac{\sum c.c.}{T.p.}$$

$$\delta = \frac{\sum u.c.}{T.p.}$$

$$f.a. = \frac{\alpha + \beta + \gamma + \delta}{4 \times 10}$$

Donde:

α = Promedio Nombres de variables y métodos.

β = Promedio Estructura del Comentario.

γ = Promedio Calidad del Comentario.

δ = Promedio Ubicación de comentarios.

T.p.= Total Programadores.

f.a. = Facilidad de análisis.

Para la primera versión del sistema la *Facilidad de Análisis* es la siguiente:

$$f.a. = \frac{7 + 7.25 + 8.25 + 6.5}{40}$$

$$f.a. = 0.72$$

Por lo tanto se puede decir que la *Facilidad de Análisis* para la primera versión del programa es regular, por ello es necesario mejorarla en este sentido.

5.2.1.2 Segunda Iteración

Para la segunda versión del sistema se encontraron los siguientes datos:

Característica	P1	P2	P3	P4
<i>n.v.</i>	9	9	10	9
<i>e.c.</i>	9	10	10	10
<i>c.c.</i>	8	9	10	7
<i>u.c.</i>	10	10	10	9

Tabla 49. Evaluación de las características por 4 desarrolladores (2a iteración).

Para la segunda versión del sistema la *Facilidad de Análisis* es la siguiente:

$$f.a. = \frac{9 + 9.5 + 10 + 8.7}{40}$$

$$f.a. = 0.93$$

Por lo tanto la segunda versión del sistema posee una excelente calidad en el factor *Facilidad de Análisis*, esto confirma la hipótesis H5a, que en la nueva versión del sistema, el factor *Facilidad de Análisis* mejoró de forma significativa.

5.2.2 Facilidad de Cambio

La *Facilidad de Cambio* es la capacidad del software para permitir modificaciones, para eliminar fallas o para cambios de entorno (ISO 2001).

El estándar IEEE 982.1-1988 sugiere un índice de madurez de software (IMS) que proporciona una indicación de la estabilidad de un producto de software, indicándonos la complejidad en su *Facilidad de Cambio* (Pressman 2005).

Para tomar decisiones sobre la *Facilidad de Cambio* se consideró como excelente valores entre 1.0 y 0.9, 0.8 – 0.9 como buena, 0.7 – 0.8 mínima aceptable, y valores inferiores como un valor malo.

El IMS se determina de la siguiente manera:

Mt = número de módulos en la versión actual.

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

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

Fd = número de módulos de la versión anterior que se han borrado en la versión actual.

El IMS se calcula de la siguiente manera:

$$IMS = [Mt - (Fa + Fc + Fd)] / Mt$$

5.2.2.1 Primera Iteración

Para la primera versión del sistema el IMS es:

$$IMS = [11 - (0 + 2 + 0)] / 11$$

$$IMS = 0.81$$

Por lo tanto se puede decir que para la primera versión del sistema, la *Facilidad de Cambio* es buena.

5.2.2.2 Segunda Iteración

Para la segunda versión del sistema el IMS es:

$$IMS = [15 - (8 + 2 + 4)] / 15$$

$$IMS = 0.06$$

Por lo tanto la segunda versión del sistema posee una mala calidad en el factor *Facilidad de Cambio*, esto rechaza la hipótesis H5b, que en la nueva versión del sistema, el factor *Facilidad de Cambio* mejoró de forma significativa.

5.2.3 Facilidad de Prueba

La *Facilidad de Prueba* es la capacidad del software de permitir validar las modificaciones realizadas (ISO 2001).

La complejidad ciclomática es la métrica con mayor aceptación. Se basa en el recuento del número de caminos lógicos individuales contenidos en un programa. La complejidad ciclomática puede usarse para elegir módulos como candidatos para pruebas más profundas. Los módulos con gran complejidad ciclomática tienen más probabilidades de tener errores que los módulos con menor complejidad ciclomática (Gomez 2008).

La *Facilidad de Prueba* de un producto de software estará por ello determinada en base a la complejidad con que los módulos del software fueron construidos.

La Complejidad Ciclomática se calcula de la siguiente manera:

$$V(G) = e - n + 2$$

Donde:

$$V(G) = \text{complejidad ciclomática}$$

$$e = \text{Número de Aristas}$$

$$n = \text{Número de nodos}$$

Si la función posee múltiples puntos de salida, entonces, la complejidad ciclomática sería definida como:

$$V(G) = e - n + c + r$$

Donde:

$$c = \text{número de componentes conectados}$$

$$r = \text{número de declaraciones de salida (returns)}.$$

La Tabla 50 muestra los riesgos de la complejidad ciclométrica:

Complejidad Ciclométrica	Evaluación de Riesgo
1-10	Función simple sin mucho riesgo
11-20	Más complejo, riesgo Moderado
21-50	Complejo, riesgo alto.
Mayor a 50	Programa inestable

Tabla 50. Riesgos medidos por la Complejidad Ciclométrica (Anderson 2004).

De la escala mencionada anteriormente se determina entonces un valor entre 1 y 10 como excelente, 11 - 20 como bueno y mayor a 20 como un valor malo.

5.2.3.1 Primera Iteración

A continuación se muestran los resultados arrojados del cálculo de la complejidad Ciclométrica para evaluar la *Facilidad de Prueba* de la primera versión del producto de software (ver tabla 51):

- | | |
|-----------------------------------|---------------------------------|
| 1. $V(G) = 33 - 26 + 2 = 9$ | 7. $V(G) = 12 - 15 + 4 + 3 = 4$ |
| 2. $V(G) = 44 - 41 + 4 + 2 = 9$ | 8. $V(G) = 15 - 20 + 5 + 5 = 5$ |
| 3. $V(G) = 83 - 81 + 14 + 4 = 20$ | 9. $V(G) = 5 - 6 + 2 = 1$ |
| 4. $V(G) = 3 - 3 + 2 = 2$ | 10. $V(G) = 5 - 6 + 2 = 1$ |
| 5. $V(G) = 10 - 11 + 2 = 1$ | 11. $V(G) = 6 - 6 + 2 = 2$ |
| 6. $V(G) = 4 - 5 + 2 = 1$ | |

Modulo	Complejidad Ciclométrica	Interpretación C.C.
1	9	Sin riesgo
2	9	Sin riesgo
3	20	Riesgo Moderado
4	2	Sin riesgo
5	1	Sin riesgo
6	1	Sin riesgo
7	4	Sin riesgo
8	5	Sin riesgo
9	1	Sin riesgo
10	1	Sin riesgo
11	2	Sin riesgo

Tabla 51. Cálculo de la complejidad Ciclométrica para la primera versión

Para la primera versión del producto de software ningún modulo presenta riesgo, la complejidad ciclomática de la aplicación es de 5, por lo tanto se puede decir que la primer versión del sistema cuenta con excelente *Facilidad de Prueba*.

5.2.3.2 Segunda Iteración

Para la segunda iteración del sistema se encontraron los siguientes valores respecto a la *Facilidad de Prueba* (ver tabla 52).

$$1.V(G) = 33 - 26 + 2 = 9$$

$$2.V(G) = 44 - 41 + 4 + 2 = 9$$

$$3.V(G) = 83 - 81 + 14 + 4 = 24$$

$$4.V(G) = 3 - 3 + 2 = 2$$

$$5.V(G) = 12 - 15 + 4 + 3 = 4$$

$$6.V(G) = 15 - 20 + 5 + 5 = 5$$

$$7.V(G) = 12 - 10 + 2 = 4$$

$$8.V(G) = 8 - 10 + 2 = 2$$

$$9.V(G) = 4 - 5 + 2 = 1$$

$$10.V(G) = 3 - 4 + 2 = 1$$

$$11.V(G) = 3 - 4 + 2 = 1$$

$$12.V(G) = 4 - 5 + 2 = 1$$

$$13.V(G) = 28 - 31 + 5 + 4 = 6$$

$$14.V(G) = 42 - 51 + 10 + 4 = 5$$

$$15.V(G) = 4 - 4 + 2$$

Modulo	Complejidad Ciclomática	Interpretación C.C.
1	9	Sin riesgo
2	9	Sin riesgo
3	24	Riesgo Alto
4	2	Sin riesgo
5	4	Sin riesgo
6	5	Sin riesgo
7	4	Sin riesgo
8	2	Sin riesgo
9	1	Sin riesgo
10	1	Sin riesgo
11	1	Sin riesgo
12	1	Sin riesgo
13	6	Sin riesgo
14	5	Sin riesgo
15	2	Sin riesgo

Tabla 52. Cálculo de la complejidad Ciclomática para la segunda versión.

Para la segunda versión del sistema la mayoría de los módulos no presentan riesgos, la complejidad ciclomática de la aplicación es de 5.06, es decir cuenta con una excelente calidad, sin embargo el resultado de la primera versión de la aplicación poseía un riesgo mucho menor por ello la hipótesis H5c se rechaza.

5.2.4 Estabilidad

Es la capacidad del software para evitar efectos no esperados debido a modificaciones (ISO 2001).

La Eficacia de la Eliminación de Defectos (EED) es una métrica que ayuda a definir la estabilidad en los productos de software ya que da un marco de referencia en proporción al número de defectos encontrados antes y después de la entrega del software. La EED se define de la siguiente manera (Pressman 2005):

$$eed = e / (e + d)$$

Donde:

e = número de errores encontrados antes de la entrega del software

d = número de defectos encontrados después de la entrega

Para tomar decisiones sobre estabilidad se consideró como excelente valores entre 1.0 y 0.9, 0.8 – 0.9 como buena, 0.7 – 0.8 mínima aceptable, y valores inferiores como malos.

5.2.4.1 Primera Iteración

El cálculo de la *Estabilidad* a través de la EED para la primera versión del sistema arrojó los siguientes resultados:

Errores antes de la entrega	Defectos después de la entrega
1. Defazamiento en menú	1.- Los scripts de cambio de tamaño de pantalla, no funcionaron en IE.
2. Scroll de galerías mal ajustado.	
3. Resolución no se adaptaba a pantalla	

Tabla 53. Relación de errores para la primera versión del sistema.

$$eed = 3 / (3 + 1) = 0.75$$

El valor de EED es 0.75, que significa que la estabilidad encontrada en la primera versión del sistema es la mínima aceptable, por lo tanto se debe prestar atención a los errores antes de la entrega definitiva.

5.2.4.2 Segunda Iteración

El cálculo de la *Estabilidad* a través de la EED para la segunda versión del sistema arrojó los siguientes resultados:

Errores antes de la entrega
1. Script de video no corría en I.E.

Tabla 54. Relación de errores para la segunda versión del sistema.

$$eed = 1/(1+0) = 1.0$$

Por lo tanto la segunda versión del sistema posee una excelente calidad en el factor *Estabilidad*, esto confirma la hipótesis H5d de que en la nueva versión del sistema, el factor *Estabilidad* mejoró de forma significativa.

5.2.5 Adaptabilidad

La *Adaptabilidad* es la capacidad del producto de software para ser adecuado a diferentes entornos específicos, sin aplicar acciones o medios diferentes de los previstos para el propósito del software (Macias-Rivero, Guzman et al. 2009).

La *Adaptabilidad* esta dentro del concepto de implementación, ya sea de hardware o de software de una máquina cualquiera, tomada en el sentido más amplio y general (Sabatini 2009).

Debido a que el sistema evaluado es un sistema web, se tomaron como diferentes escenarios de adaptabilidad a los diferentes navegadores que un usuario puede utilizar para acceder al sistema.

Según un estudio realizado en marzo de 2010 sobre los navegadores Web más populares en la actualidad (MarketShare 2010) se encontró los siguientes resultados (ver Tabla 55).

Navegador	Uso
Internet Explorer	59.95%
Firefox	24.59%
Chroome	6.7%
Safari	4.7%
Opera	3.0%
Netscape	0.56%

Tabla 55. Porcentaje navegadores web más populares (MarketShare 2010).

Según estudios realizados por IBOPE (Islas 2009) el 80% de la población en México posee algún dispositivo móvil (celular, PDA, pocket pc), poco menos del 25% de éstos lo utilizan para enviar MMS (mensajes multimedia) por Internet, y menos del 5% lo utiliza para navegar en Internet, debido al poco porcentaje de uso del internet en dispositivos móviles, no se profundizará mucho en este aspecto, sin embargo se reconoce el crecimiento tecnológico en este sentido, por ello se tomó únicamente uno de los navegadores en dispositivos móviles con mas auge en la actualidad: Digi@ Web.

En 2008 la empresa Digia (DIGIA 2010) desarrolló un navegador para dispositivos móviles compatible con dispositivos táctiles, de nombre Digia @Web, el cual ha incrementado su popularidad en los últimos días, por ello se utilizó este navegador en la presente evaluación.

En base a lo anterior el cálculo de la adaptabilidad será en base a la prueba del sistema en los diferentes ambientes, es decir los 6 navegadores web más populares en la actualidad, y un navegador de dispositivos móviles.

El cálculo de la *Adaptabilidad* será la sumatoria del éxito o fracaso de cada uno de los diferentes escenarios, en donde el éxito en todos de ellos dará como resultado la unidad que se toma como el valor idóneo. Por lo tanto el cálculo de la adaptabilidad queda de la siguiente forma:

$$a = N \times E$$

Donde:

N = Valor por navegador

E = numero de escenarios exitosos

El valor de éxito por Navegador es de 0.1429, que corresponde a la división de la unidad entre el número de navegadores a probar, de tal forma un caso excelente dará como resultado 1.

Para tomar decisiones sobre *Adaptabilidad* se consideró como excelente valores entre 1.0 y 0.9, 0.8 – 0.9 como buena, 0.7 – 0.8 mínima aceptable, y valores inferiores como malos.

5.2.5.1 Primera Iteración

Para la primera versión las pruebas realizadas al sistema arrojaron el siguiente resultado (ver tabla 56).

Navegador	Sistema Operativo	Resultado
Netscape	Windows, Mac Os, Linux	Bueno
Internet Explorer	Windows	Malo
Mozilla	Windows, Mac Os, Linux	Bueno
Opera	Windows, Mac Os, Linux	Bueno
Safari	Windows, Mac Os	Bueno
Google Chrome	Windows, Mac Os, Linux	Bueno
Digia @Web (Dispositivos Móviles)	S.60, Maemo, UIQ 3	Malo

Tabla 56. Escenarios de prueba primera Iteración.

El número de casos exitosos fue de 5 por lo cual el valor de la *Adaptabilidad* en la primera versión del sistema fue de la siguiente forma.

$$a = 0.1429 \times 5$$

$$a = 0.71$$

Por lo tanto la calidad de la *Adaptabilidad* para la primera versión del sistema es la mínima aceptable.

5.2.5.2 Segunda Iteración

En la segunda ronda se encontraron los siguientes resultados (ver tabla 57).

Navegador	S.O.	Resultado
Netscape	Windows, Mac Os, Linux	Bueno
Internet Explorer	Windows	Bueno
Mozilla	Windows, Mac Os, Linux	Bueno
Opera	Windows, Mac Os, Linux	Bueno
Safari	Windows, Mac Os	Bueno
Google Chrome	Windows, Mac Os, Linux	Bueno
Digia @Web (Dispositivos Móviles)	S.60, Maemo	Bueno

Tabla 57. Escenarios de prueba segunda Iteración.

Para la segunda ronda, los escenarios exitosos fueron todos, por lo tanto el cálculo de la *Adaptabilidad* fue.

$$a = 0.1429 \times 7$$

$$a = 1.0$$

La calidad de la *Adaptabilidad* para la segunda versión del sistema fue de 1.0, esto confirma la hipótesis H6a, que en la nueva versión del sistema, el factor de *Adaptabilidad* mejoró de forma significativa.

5.2.6 Conformidad

Es la capacidad del software para adherirse a estándares o convenciones relativas a la portabilidad (ISO 2001).

Mooney (2006) propone una métrica que ayuda a estimar la portabilidad en productos de software.

El rol principal de las métricas en relación a portabilidad es ayudar a caracterizar el costo beneficio de la importancia de tomar en cuenta la portabilidad en los proyectos de software (Mooney 2006).

Mooney (2006) asume que los costos más grandes en el desarrollo de software son los costos humanos envueltos en los procesos de desarrollo, por ello toma como primicia los costos de cada fase del proceso de desarrollo de software para derivar

su métrica, para este caso en particular y dado que el sistema realizado para la presente investigación no fue con fines lucrativos, se toma la medida de tiempo, basado en lo que el mismo autor de la métrica dice al mencionar que la medida de estimación son los días-persona, ya que a mayor tiempo, mayor costo en el desarrollo.

En base a esto el cálculo de la *Conformidad* en la portabilidad es de la siguiente forma:

$$Cdev(req,e1) = Cdes(req) + Ccod(req,e1) + Ctd(req,e1) + Cdoc(req,e1).$$

Donde:

$Cdev(req,e1)$ = Costo desarrollar software

$Cdes(req)$ = costo análisis y diseño

$Ccod(req,e1)$ = costo codificación

$Ctd(req,e1)$ = costo pruebas

$Cdoc(req,e1)$ = costo documentación

De lo anterior se puede decir que $Cdev(req,e1)$ es el costo total del proyecto, el cual servirá como referencia para el cálculo de costos de los siguientes elementos.

$$Cdevp(req,e1) = Cdev(req,e1) + Cpa(req)$$

Donde:

$Cdevp(req,e1)$ = costo rediseño.

$Cpa(req)$ = diseño portable

$Cdevp(req,e1)$ es el costo del desarrollo con el agregado Cpa , que quiere decir que en el proceso de desarrollo se hizo además un diseño portable desde el inicio.

$$Crdev(req,e2) = Crdes(req) + Crcod(req,e2) + Crtd(req,e2) + Crdoc(req,e2)$$

Si se hace un rediseño, se puede agregar Cpa Opcionalmente si el rediseño es para agregar portabilidad, el rediseño significa que la aplicación como tal debe rehacerse para agregar la portabilidad deseada.

$$C_{port}(su,e2) = C_{mod}(su,e2) + C_{ptd}(req,e2) + C_{pdoc}(req,e2).$$

Donde:

$C_{port}(su,e2)$ = portabilidad del software en nuevos ambientes

$C_{mod}(su,e2)$ = modificación manual

$C_{ptd}(req,e2)$ = pruebas

$C_{pdoc}(req,e2)$ = documentación

C_{port} es el costo de una modificación a un software existente para agregar la portabilidad que se desea.

$$DP(su) = 1 - (C_{port}(su,e2) / C_{rdev}(req,e2)).$$

Donde:

$DP(su)$ = conformidad en la portabilidad

5.2.6.1 Primera Iteración

Para la primera versión de la aplicación el costo del desarrollo de la aplicación fue de la siguiente manera.

$$C_{dev}(req,e1) = 5 + 15 + 2 + 2 = 24$$

Como el autor de la métrica lo indica tomamos este valor como la unidad.

$$24 = 1.0$$

En base a este cálculo los porcentajes por fase fueron de esta forma

$$C_{dev}(req,e1) = 0.22 + 0.62 + 0.08 + 0.08 = 1.0$$

La aplicación no funcionó correctamente en el navegador Internet Explorer (I.E.) por lo tanto un rediseño de la aplicación tendría prácticamente el mismo costo que el diseño original. Ya que se tendría que realizar todo nuevamente. Sin embargo existe un ahorro debido a que gran parte del análisis ya se desarrollo, como indica el autor de la métrica el valor del rediseño tendría un ahorro del 10% sobre el costo inicial.

$$C_{rdev}(req,e2) = 0.9$$

Se realizaron modificaciones en el software para corregir los problemas de compatibilidad con algunos navegadores, el costo de estas modificaciones fue como se muestra.

$$C_{port}(su,e2) = 25+5+5 = 35$$

Esto se traduce en porcentaje como:

$$C_{port}(su,e2) = 1.0 + 0.2 + 0.2 = 1.4$$

Consecuentemente la *Conformidad* de la evaluación para la primera iteración fue de la siguiente forma.

$$DP(su) = 1 - (1.4/0.9) = -0.5$$

Por lo tanto la portabilidad para la primera versión del sistema es muy mala, esto debido a que desde un inicio no se tomó en cuenta un diseño para portabilidad.

5.2.6.2 Segunda Iteración

Con las modificaciones realizadas para la segunda versión del software, se corrigió en gran medida el problema de portabilidad, sin embargo se planteó un nuevo ambiente, los dispositivos móviles, gracias a las correcciones que se realizaron la nueva modificación fue más sencilla y el costo de esta fue de esta forma.

$$C_{port}(su,e2) = 3+0.5+0.5 = 4$$

Esto se traduce en porcentaje como:

$$C_{port}(su,e2) = 0.1 + 0.02 + 0.02 = 0.14$$

Por lo tanto la *Conformidad* para la segunda versión del sistema es de:

$$DP(su) = 1 - (0.14/0.9) = 0.84$$

Por lo anterior la calidad en el factor *Conformidad* en la segunda versión del sistema es mucho mejor, según el autor de la métrica, este valor irá mejorando en cada nueva iteración del sistema. Por ello se puede decir que la hipótesis H6b se confirma.

5.3 Resultados de las Hipótesis

A continuación se muestra un concentrado de las pruebas de hipótesis realizadas, en donde se puede observar (ver tabla 58) que de las 16 hipótesis se aceptan 14 y se rechazan 2, ello confirma que la calidad de la aplicación en la segunda versión se mejoró en base a la evaluación y mejoras realizadas.

Hipótesis	Resultado
1.H1a	Se Acepta
2.H1b	Se Acepta
3.H1c	Se Acepta
4.H1d	Se Acepta
5.H2a	Se Acepta
6.H2b	Se Acepta
7.H3a	Se Acepta
8.H3b	Se Acepta
9.H3c	Se Acepta
10.H4a	Se Acepta
11.H5a	Se Acepta
12.H5b	Se Rechaza
13.H5c	Se Rechaza
14.H5d	Se Acepta
15.H6a	Se Acepta
16.H6b	Se Acepta

Tabla 58. Resultados concentrados de la prueba de hipótesis

6. CONCLUSIONES

6.1 Conclusiones Generales

Conforme a los datos obtenidos de la evaluación realizada, se pudo comprobar que el integrar al usuario en la evaluación de una aplicación de software, ayuda a incrementar la calidad en una siguiente versión, ya que de los 10 factores medidos desde el punto de vista del usuario final, todos mostraron una significativa mejora.

De acuerdo a los resultados obtenidos, se comprobó que el tomar en cuenta métricas para mediciones en el desarrollo del software, ayudó a incrementar la calidad en una nueva versión, ya que de los 6 factores medidos desde la orientación del desarrollo, 4 mostraron una mejora significativa. A continuación se detallan las conclusiones relacionadas a cada factor.

La *Operabilidad* que forma parte de la característica general de *Usabilidad*, se mejoró significativamente en la segunda versión, en este factor se tomaron en consideración los comentarios de los usuarios sobre la *Operabilidad* del producto de software, ya que dijeron no estar conformes con el menú de la primera versión, por ello se hizo un replanteamiento en la forma de navegación para la segunda versión del sistema.

Para el factor de *Facilidad de Aprendizaje* perteneciente a la característica general de *Usabilidad*, en la primera versión de la aplicación los usuarios dijeron confundirse en algunas secciones del sistema, algunos sugerían el uso de botones de regresar y la mayoría proponía un menú más sencillo, por lo tanto en la segunda versión se buscó tener un menú bastante intuitivo, que facilitara a los usuarios su navegación de forma natural, esto mejoró significativamente la calidad en éste factor.

En el *Atractivo*, también perteneciente a *Usabilidad*, se encontró que para la primera versión del sistema los colores no agradaron a los usuarios, los colores fueron seleccionados en base al estándar de gobierno del estado de Zacatecas para páginas web, en la segunda versión se buscaron colores que permitieran la fácil

lectura y que resaltaran el sitio para hacerlo más agradable, por esto en la segunda versión del sistema la calidad se mejoró de forma significativa.

La *Comprensibilidad* fue el último factor evaluado dentro de la característica de *Usabilidad*, en este apartado se mejoraron algunos textos, cambiando oraciones a algunas más simples, se buscaron términos técnicos difíciles de entender y fueron cambiados por palabras más entendibles, así mismo se busco que la forma de navegación de botones fuera acorde al nuevo menú, para lograr un buen entendimiento por parte del usuario sin mayor esfuerzo, por esto la segunda versión mostro un valor de calidad significativamente mejor que la primer versión del sistema.

El primer factor de la característica general *Funcionalidad* en ser evaluado, fue el de *Conformidad*, éste mejoro significativamente durante la segunda versión del sistema principalmente en las interfaces del sistema, al realizar un reacomodo de los elementos de pantalla, y un nuevo diseño para estos. También se cambió el tipo de letra y su tamaño ello porque los usuarios en la primera evaluación dijeron no estar conformes con los que se tenían.

Para el factor *Exactitud* perteneciente a la *Funcionalidad* se mejoró significativamente la calidad en la segunda versión del sistema, esto se logró corrigiendo algunos scripts encargados de ajustar la resolución de los elementos de pantalla, los cuales no funcionaron de manera correcta en el navegador I.E. en la segunda ronda los elementos funcionaron como se esperaba.

En cuanto a la *Madurez*, *Tolerancia a Fallas*, y *Recuperabilidad* del sistema, que forman parte de la característica general de *Confiabilidad*, se encontró que todas mejoraron en la segunda versión. Durante la primera versión de la aplicación en varias computadoras de las utilizadas para la evaluación, que contaban únicamente con I.E. se mostraba la leyenda "listo pero con errores en la página" esto influyó en el comportamiento de algunos elementos del sistema ya que no era el indicado, por lo tanto el error no se recuperaba. Para la segunda versión se corrigieron estos problemas, sin embargo cabe destacar que en caso de existir un error no es el

sistema mismo quien da el mensaje de error sino que en muchas ocasiones es el navegador, y se utilizan palabras técnicas que los usuarios podrían no entender, sin embargo al no formar parte de la aplicación esto queda fuera de la Investigación.

En cuanto al *Tiempo de Respuesta* perteneciente a la característica general de *Eficiencia*, el *Tiempo de Respuesta* hallado en el primer sistema funcionando bajo condiciones normales (Mozilla) era bueno, sin embargo debido a los problemas de incompatibilidad de algunos scripts con Internet Explorer, el funcionamiento del sistema no fue el adecuado, por ello el tiempo de respuesta era mayor en esta condición. Para la segunda versión donde estos problemas se corrigieron, el tiempo de respuesta encontrado fue significativamente mejor a la primera versión.

Con respecto a la *Facilidad de Análisis* perteneciente a la característica general de *Mantenibilidad*, en la primer versión del sistema, había ciertos apartados de difícil comprensión ya que no se habían comentado, nombres de variables, y métodos que no hacían alusión a su uso, por ello al corregir estos detalles la *Facilidad de Análisis* fue significativamente mejor en la segunda versión del sistema.

Para la *Facilidad de Cambio* factor de la característica general *Mantenibilidad*, para la primera versión del sistema se encontró que tenía una buena calidad, sin embargo al encontrar que algunos módulos del sistema eran incompatibles con Internet Explorer, hubo que descartarlos, y crear nuevos módulos, otros más se modificaron según las exigencias del Usuario, esto provocó que la medida de *Facilidad de Cambio* de la segunda versión del sistema tuviera mala calidad.

La *Facilidad de Prueba* es un factor de la característica general *Mantenibilidad*, en la primer versión del sistema el cálculo de la complejidad ciclomática arrojó como resultado una buena *Facilidad de Prueba*, sin embargo para satisfacer el gusto del usuario, hubo que cambiar interfaces, elementos, colores, colocar nuevos módulos y descartar módulos existentes, ello dio como resultado una complejidad ciclomática más elevada que en la primer versión, por lo tanto la calidad no mejoró en la segunda versión del sistema.

El último factor evaluado de la característica general *Mantenibilidad*, fue la *Estabilidad*, en donde a través del cálculo de la EED, se encontró que la primer versión tuvo una *Estabilidad* no tan buena ya que se encontraron algunos errores en la primer iteración, sin embargo ello ayudó a mejorar para la segunda versión, por ello la calidad encontrada en la segunda ronda fue significativamente mayor.

Con respecto a la *Portabilidad* del sistema se evaluaron los factores de *Adaptabilidad* y *Conformidad*, para la *Adaptabilidad* se establecieron los diferentes escenarios donde el sistema podría ser utilizado, y en base al funcionamiento de éste bajo esas condiciones se determino la calidad, en la primer versión el sistema no funcionaba correctamente en el navegador Internet Explorer, ni se había tomado en cuenta hacer el sitio compatible con dispositivos móviles. Para la segunda versión estos errores se corrigieron y por lo tanto la *Adaptabilidad* de la aplicación fue excelente. Por esto la *Conformidad* en la primera versión dio un mal resultado, y en la segunda versión gracias a las modificaciones realizadas la *Conformidad* mejoró significativamente.

Finalmente con 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, ayuda a obtener productos con mejor calidad y cumplir con la satisfacción del usuario.

6.2 Limitaciones

Las limitantes que se encontraron a lo largo de la presente investigación fueron las siguientes:

- La Investigación se atraso un tiempo signficante, ya que cuando todos los elementos estaban listos para la primera evaluación (incluyendo centro de cómputo), nos percatamos de que el sistema web a evaluar no se podía visualizar dentro de la Universidad Autónoma de Aguascalientes, ello debido a las exageradas políticas de seguridad del Departamento de Redes, por este motivo se hizo una petición este departamento para comunicar el inconveniente, posteriormente se fijó una nueva fecha para la evaluación, la

cual fue postergada nuevamente debido a que, la solicitud verbal no era suficiente, por tal motivo fue necesario enviar un oficio con la petición a las instancias apropiadas, luego de muchos procesos burocráticos se pudo llevar a cabo la evaluación.

- Debido al tiempo reducido para realizar la investigación no se pudo crear un software de mayor tamaño y más complejo para realizar la investigación, ya que un sistema de mayor magnitud no sólo requeriría un mayor tiempo en desarrollo, sino que también significaba diferentes estrategias para su evaluación, se desarrolló un sistema web porque este tipos de sistemas tienen una orientación general, además de que facilitan la evaluación al permitir crear herramientas online para esta tarea y se ajustaba a los tiempos perfectamente.
- A causa de que se trata de un sistema web de carácter turístico, ninguna información del usuario es registrada, ello es una limitante ya que no se hicieron mediciones con respecto a la seguridad de datos y manejo de información, que en la actualidad son de carácter importantísimo para algunos productos de software.
- La segunda evaluación no se pudo llevar a cabo en los tiempos que se había establecido ya que en varias ocasiones, se había planeado la fecha de evaluación, los usuarios habían sido informados, pero no se encontraban centros de cómputo disponibles para la evaluación.
- Al tratarse de un sistema web, el tiempo de respuesta se ve afectado por los recursos del sistema en donde la página se utiliza, es decir que aunque el código se optimice para mejorar este sentido, si se utiliza un equipo muy antiguo, el funcionamiento del sistema estará muy alejado de ser el adecuado.
- El tiempo de carga será acorde a la tasa de transferencia del ISP (proveedor de servicios de Internet) con que se cuente, en las evaluaciones

desarrolladas en la Universidad Autónoma de Aguascalientes, en algunas computadoras el tiempo de carga fue exageradamente lento.

- La computadora donde se utilice el sistema debe contar con el códec de Adobe Shockwave, y la máquina de Java, ya que sin estos los videos no se pueden visualizar, ni se reproducen los sonidos como en el apartado de leyendas, en la evaluación realizada algunas computadoras no contaban con codecs de Adobe Shockwave o tenían codecs antiguos, ello provocó que los videos no se pudieran visualizar.

6.3 Trabajos Futuros

Pese a que los resultados obtenidos en la segunda versión de la aplicación fueron significativamente mejores a la primera evaluación, en algunos factores tales como *Facilidad de Prueba* y *Facilidad de Cambio* los datos obtenidos fueron mejores en la primera versión del sistema esto es debido a las modificaciones que se realizaron a los módulos, eliminación de otros y creación de nuevos módulos de software. Por ello para la correcta estabilidad del sistema y valores óptimos de calidad en los factores medidos para mejorar la *Mantenibilidad* y *Portabilidad* es necesario realizar más iteraciones del software ya que cada iteración hace más estable el software en los factores mencionados, por ello es recomendable realizar una tercera iteración para obtener resultados de excelente en todos los factores de calidad medidos.

Las métricas que se utilizaron para medir la *Facilidad de Análisis*, y la *Adaptabilidad* fueron propuestas de esta investigación, por lo tanto en trabajos futuros sería recomendable utilizarlas en otras investigaciones para mejorarlas y validarlas de forma más profunda.

Realizar un plan estratégico para las métricas del lado del desarrollador desde el comienzo del desarrollo del sistema, ya que para esta investigación muchas de las métricas fueron definidas cuando ya se había desarrollado la primera versión del sistema, esto repercutió en la calidad de algunas métricas como *Facilidad de Análisis* la cual no contó con un buen nivel de calidad en la primer evaluación.

7. BIBLIOGRAFÍA

1. Abran, A..A. Khelifi, et al. "Usability Meanings and Interpretations in ISO Standards." *Software Quality Journal*.(2003).
2. AMIPCI. "ESTUDIO AMIPCI 2009 Sobre hábitos de los Usuarios de Internet en México."(2009).
3. Anderson, J. L. "Using Software Tools and Metrics to Produce Better Quality Test Software." *AUTOTESTCON. Proceedings* (2004).
4. Bertoa, M. F..J. M. Troya, et al. Aspectos de Calidad en el Desarrollo de Software Basado en Componentes. Depto. Lenguajes y Ciencias de la Computación. Universidad de Málaga.(2002).
5. Bevan, N. y M. Macleod. "Usability measurement in context, Behavior and Information Technology." pp. 132–145.(1994).
6. Boehm, B. W..J. R. Brown, et al. Quantitative evaluation of software quality. *International Conference on Software Engineering. San Francisco, California, United States IEEE Computer Society Press*.(1976).
7. Calero, M. y Coral. Modelos de Calidad. WQM, PQM, e-commerce, portlets. Departamento de Informática, Universidad de Castilla-La Mancha.(2005).
8. Carvallo, J. P. y X. Franch. "Extending the ISO/IEC 9126-1 Quality Model with Non-Technical Factors for COTS Components Selection."Cuenca, Ecuador.(2006).
9. Castañeda, A..J. M. Gómez-Reynoso, et al. Método de Evaluación de Software Para Mejorar la Calidad del Producto Final. Ingeniería de Software. Aguascalientes, Ags., Universidad Autónoma de Aguascalientes.(2009).
10. Cavano, J. P. y J. A. McCall. "A framework for the measurement of software quality." *ACM New York, NY, USA* (1978).
11. Clery, A. G. "La calidad total como una estrategia competitiva aplicada a una empresa de servicios de instalaciones eléctricas y telecomunicaciones." Consultada en 10/08/09, 2009, fuente <http://temas-gerenciales-modernos.lacoctelera.net/post/2009/10/02/calidad-total-como-ventaja-competitiva>.(2009).
12. DIGIA. "Digia Technology." Consultada en septiembre 2010, fuente <http://www.digia.com/C22571C5003CAD30/0/405001848>.(2010).
13. Ejiogu, L. "Software Engineering with formal metrics."QED Publishing.(1991).
14. Firesmith, D. Achieving Quality Requirements with Reused Software Components: Challenges to Successful Reuse. C. M. University. Pittsburgh, PA 15213, Software Engineering Institute.(2005).
15. Gómez-Pantoja, J. L. "El código de Hammurabi."Universidad de Alcala.(1998).
16. Gomez-Reynoso, J. M. y M. R. Brizuela-Sandoval. Participación de los usuarios en sistemas de información. Americas Conference on Information System. Toronto, ON, Canada, Association of Information Systems.(2008).
17. Gomez-Reynoso, J. M..E. L. Muñoz-Andrade. Utilizando el Modelo de Calidad McCall y el Estándar ISO-9126 para la Evaluación de la Calidad de Sistemas por los Usuarios. Proceedings of the sixteenth Americas Conference of Informations Systems Lima, Peru.(2010).
18. Gomez, D. "Complejidad Ciclomática." Consultada en septiembre 2010, fuente <http://www.dosideas.com/noticias/metodologias/320-como-entender-la-complejidad-ciclomatica.html>.(2008).
19. Hernandez-Sampieri, R. Metodología de la Investigación, McGraw-Hill.(1997).
20. Honglei.Wei, et al. The Research on Software Metrics and Software Complexity Metrics. Computer Science-Technology and Applications. IFCSTA '09. International Forum on Chongqing, China, IEEE.(2010).
21. Humphrey, W. S. "The Personal Softwareren Process (PSP)."Pittsburgh, PA.(2000).
22. Humphrey, W. S. PSP: A Self-Improvement Process for Software Engineers Addison-Wesley.(2005).

23. Hyatt, L. E. y Rosenberg. "A software quality model and metrics for identifying project risks and assessing software quality." Noordwijk, The Netherlands Software Product Assurance Workshop.(1996).
24. INEGI. "DISPONIBILIDAD Y USO DE LAS TECNOLOGÍAS DE LA INFORMACIÓN EN LOS HOGARES ".(2008).
25. INEGI. "ESTADÍSTICAS A PROPÓSITO DEL DÍA MUNDIAL DE INTERNET."(2009).
26. Islas, O. "RESULTADOS DEL ESTUDIO DE IBOPE EN MATERIA DE HÁBITOS DE INTERNET, 2009 SEGUNDA PARTE Y ÚLTIMA." Consultada en octubre 2010, fuente <http://www.razonypalabra.org.mx/espejo/2009/mar20.html>.(2009).
27. ISO. ISO 9126-1:2001 Software engineering-Product quality. Part 1: Quality Model, International Organization for standardization.(2001).
28. ISO. Information Technology – Process assessment. Part 5: An exemplar process assessment mod-el. Technical Report.(2004).
29. ISO/IEC. Information technology – software product evaluation – quality characteristics and guidelines for their use. Technical Report 9126, International Organization for Standardization.(1990).
30. Kafura, D. y J. Canning. A validation of software metrics using many metrics and two resources. International Conference on Software Engineering. Los Alamitos, CA, USA, IEEE Computer Society Press.(1985).
31. Khan, R. A..K. Mustafa, et al. Software Quality Concepts and Practices, Alpha Science.(2006).
32. Lewis, W. E. Software Testing and Continuous Quality Improvement, CRC Press.(2009).
33. Li, W. "Software Product Metrics." Potentials, IEEE.(2002).
34. Macias-Rivero, Y..M. Guzman, et al. "Evaluation model for the software using metric indicators to science and technology surveillance " ACIMED **Vol. 20**.(2009).
35. Macracken, D. y R. J. Wolfe. User-Centered Website Development, Prentice Hall.(2004).
36. MarketShare. "Navegadores más populares." Consultada en septiembre 2010, fuente <http://arturotena.com/2010/04/06/navegadores-mas-populares-marzo-2010-browser-share-march-2010/>.(2010).
37. McCall, J. A..P. Richards, et al. Factors in software quality, US Rome Air Development Centre Reports.(1977).
38. Mooney, J. D. Issues in the Specification and Measurement of Software Portability. 15th International Conference on Software Engineering. Los Angeles, California., Dept. of Statistics and Computer Science.(2006).
39. Murine, G. E. "Integrating software quality metrics with software QA." Quality Progress **11**: 38 - 43.(1988).
40. Nigel, B. "Meeting User Needs for Quality." Journal of System and Software.(2004).
41. Padron, J. "Bases del Concepto de Investigación Aplicada."Caracas.(2006).
42. Pérez, G. "Calidad Total." Consultada en 25/09/09, fuente http://www.wikilearning.com/curso_gratis/teorias_de_calidad-origenes_y_tendencias_de_la_calidad_total/11500-1.(2009).
43. Pfleeger y S. Lawrence. Ingeniería de software: teoría y práctica.(2002).
44. Piattini, M. G. y F. O. García. Calidad en el desarrollo y mantenimiento del software, Rama.(2003).
45. Prechelt, L. y B. Unger. "An Experiment Measuring the Effects of Personal Software Process (PSP) Training." IEEE TRANSACTIONS ON SOFTWARE ENGINEERING.(2000).
46. Pressman, R. Ingeniería de Software, McGraw Hill.(2005).
47. Sabatini, A. (septiembre 2010). "Adaptabilidad (Interaction-Interface)." Consultada en septiembre 2010, fuente <http://interfacemindbraincomputer.wetpaint.com/page/2.A.1.1.1.-Adaptabilidad+%28Interaction-Interface%29>.(2009).
48. Seffah, A..M. Donyaee, et al. "Usability measurement and metrics: A consolidated model." Software Qual J.(2006).

49. Sicilia, M. y V. De la Morena. "Métricas de Mantenibilidad Orientadas al Producto." Consultada en septiembre 2010, fuente <http://cnx.org/content/m17466/latest/>.(2009).
50. Sivamuni, K. y S. Rengaramanujam. "A Retrospective on Software Component Quality Models."(2008).
51. Sommerville, I. Ingeniería de Software, Addison Wesley.(2006).
52. Tullis, T. S. y J. S. Stetson. A Comparison of Questionnaires for Assessing Website Usability, Human Interface Design Department, Fidelity Center for Applied Technology.(2004).
53. Ward, W. A. y B. Venkataraman. "Some Observations on Software Quality."School of Computer and Information Sciences, University of South Alabama.(1999).
54. Zarazua, M. "Calidad Total."Nort Miami, Florida.(2005).
55. Zeiss, B. y D. Vega. "Applying the ISO 9126 Quality Model to Test Specifications. Institute for Informatics."G"ottingen, Germany.(2007).

