UNIVERSIDAD CATÓLICA ANDRÉS BELLO
VICERRECTORADO ACADÉMICO
ESTUDIOS DE POSTGRADO
FACULTAD DE INGENIERÍA
POSTGRADO EN SISTEMAS DE INFORMACIÓN

TRABAJO DE GRADO DE MAESTRÍA

MODELO DE EVALUACIÓN DE LA CALIDAD DEL PRODUCTO DE SOFTWARE PARA MEDICINA PREPAGADA

Presentado por:

Ing. Carrasco Sosa, Belkis Elena

Para optar al título de
Magíster en Sistema de Información

Tutor
Bonillo Ramos, Pedro Nolasco

Caracas, julio de 2016

UNIVERSIDAD CATÓLICA ANDRÉS BELLO
VICERRECTORADO ACADÉMICO
ESTUDIOS DE POSTGRADO
FACULTAD DE INGENIERÍA
POSTGRADO EN SISTEMAS DE INFORMACIÓN

TRABAJO DE GRADO DE MAESTRÍA

MODELO DE EVALUACIÓN DE LA CALIDAD DEL PRODUCTO DE SOFTWARE PARA MEDICINA PREPAGADA

Presentado por:

Ing. Carrasco Sosa, Belkis Elena

Para optar al título de
Magíster en Sistema de Información

Tutor
Bonillo Ramos, Pedro Nolasco

Caracas, julio de 2016

Caracas, 24 de febrero de 2016

UNIVERSIDAD CATÓLICA ANDRES BELLO

Postgrado en Sistemas de Información

Lic. María Esther Remedios

Presente.-

CARTA DE ACEPTACIÓN DEL TUTOR

Por la presente me permito comunicar que he sido el tutor del Trabajo de Grado de Maestría del estudiante Belkis Elena Carrasco Sosa, cédula de identidad nro. 19.453.820, quien opta por el título de Magíster en Sistemas de Información, intitulado “Modelo de evaluación de la calidad del producto de software para medicina prepagada”.

Asimismo, hago constar que como tutor estoy conforme con el contenido presentado, por lo que cuenta con mi aprobación para ser inscrito como Trabajo de Grado de Maestría.

Sin otro particular al cual hacer referencia, se despide cordialmente,

_____

Dr. Pedro Nolasco Bonillo Ramos

C.I. 10.868.538

DEDICATORIA

A mi madre,

Porque cada uno de mis logros te los debo a ti, por tu apoyo incondicional, dedicación y amor.

Gracias por confiar en mí.

Te amo

Abuelita,

Desde el cielo me mandabas tu bendición que me guiaba y cuidaba desde el primer día de carrera. Recordando lo orgullosa que te sentiste siempre por lo mucho que estudiaba, este logro también te lo dedico a ti.

Te amaré siempre

AGRADECIMIENTOS

A Dios y San Miguel Arcángel por guiarme y bendecir mis pasos durante esta etapa.

A mis padres, mi hermana y mi cuñado, por ser parte importante en el cumplimiento de esta meta, su apoyo, amor y compañía me ayudaron a seguir adelante.

A Carlos, por recordarme siempre de que estamos hechos y lo que somos capaces de lograr, tu confianza en mí siempre ha sido mi fuerza, celebro contigo una victoria más.

A mis compañeros y amigos por compartir momentos hermosos, son grandes personas y excelentes profesionales.

A mi tutor, profesor Pedro Bonillo, por guiarme, apoyarme incondicionalmente y ser un ejemplo a seguir.

A la Universidad Católica Andrés Bello (UCAB) y todos los profesores de la carrera, por abrirme las puertas, guiarme, su dedicación y por brindarme tanto crecimiento personal como profesional.

A todas aquellas personas que formaron parte de este sueño, por su apoyo y fe en que este esfuerzo daría grandes frutos.

A todos muchas gracias.

Belkis Carrasco

UNIVERSIDAD CATÓLICA ANDRÉS BELLO
VICERRECTORADO ACADÉMICO
ESTUDIOS DE POSTGRADO
FACULTAD DE INGENIERÍA
POSTGRADO EN SISTEMAS DE INFORMACIÓN

MODELO DE EVALUACIÓN DE LA CALIDAD DEL PRODUCTO DE SOFTWARE PARA MEDICINA PREPAGADA

Autor: Carrasco Sosa, Belkis Elena
Tutor: Bonillo Ramos, Pedro Nolasco
Año: 2016

RESUMEN

El presente trabajo de investigación se llevó a cabo con el objetivo de proponer al sector de medicina prepagada un modelo de evaluación de la calidad del producto de software, que solvente los inconvenientes que presentan los estándares existentes en el mercado, entre los cuales se tiene la detección y corrección de errores en etapas tardías con grados de implicación máximos o inaceptables. El diseño de este patrón de la calidad se basó en la norma ISO/IEC 9126-1:2000, y en él se busca otorgar como beneficio la adaptabilidad al desarrollo basado en componentes, bajo costo en asesorías costosas para su aplicación, disminuir los tiempos de construcción de sistemas de información, reutilización de código, y dilucidar las preguntas frecuentes o dudas acerca de cómo adentrarse en el mundo de la calidad del software. Esta investigación es de tipo descriptiva, lo que implica el uso de la observación para especificar el entorno del problema a tratar. A su vez, se caracteriza por ser de tipo aplicada, debido a la validación del modelo a diseñar y proponer, a través de pruebas pilotos, aplicadas en dos empresas pertenecientes al sector de medicina prepagada. Con el cumplimiento de los objetivos planteados se busca disminuir la brecha entre los sistemas de información en Venezuela y el aseguramiento de la calidad del producto de software, otorgando la orientación necesaria para evitar el incremento del presupuesto de los proyectos en consultorías costosas. Para ello se diseñó un modelo de evaluación de la calidad del producto de software que consiste en 11 características a evaluar a través de 33 subcaracterísticas y sus respectivas métricas, el cual fue validado y aplicado en dos empresas de medicina prepagada tomadas como muestra.

Palabras Clave: Adaptabilidad, calidad, evaluación, modelo, características, subcaracterísticas, reutilización de código.

Línea de Investigación: Ingeniería de software (Calidad de producto de software)

INDICE GENERAL

DEDICATORIA ii
AGRADECIMIENTOS iii
RESUMEN iv
INDICE GENERAL 1
INDICE DE FIGURAS 4
INDICE DE TABLAS 6
INTRODUCCIÓN 8
CAPITULO I: EL PROBLEMA 11
1.1 Planteamiento del Problema 11
1.1.1 Formulación del Problema 14
1.1.2 Sistematización del Problema 14
1.2. Objetivos 15
1.2.1 Objetivo General 15
1.2.2 Objetivos Específicos 15
1.3. Justificación de la Investigación 15
1.4. Alcance y limitaciones de la Investigación 17
CAPÍTULO II: MARCO TEÓRICO 18
2.1 Antecedentes 18
Trabajos de Grado de Maestría 18
Artículos Técnicos 19
2.2. Fundamentos Teóricos 22
2.2.1. Ingeniería de software 22
2.2.2. Calidad de software 23
2.2.2.1 Tipos de calidad de software 25
2.2.2.2 Garantía de calidad 26
2.2.2.3 Planificación de la calidad 27
2.2.2.4. Control de la calidad 28
2.2.2.5. Costos de la calidad 28
2.2.3. Estándares para la evaluación de la calidad de producto de software . 29
2.2.3.1. ISO/IEC 9126-1:2000: 29

2.2.3.2. McCall 26
2.2.3.3. Dromey 27
2.3 Marco Referencial 28
Misión 28
Visión 28
Objetivos 29
Atribuciones 29
Estructura organizacional 30
2.4. Bases Legales 34
CAPITULO III: MARCO METODOLÓGICO 36
3.1 Tipo de Investigación 36
3.2 Diseño de la Investigación 37
3.3 Población y Muestra 38
3.4 Técnicas e instrumentos de recolección de datos 39
3.5 Fases de la Investigación 40
3.6 Procedimiento por Objetivos 40
3.7 Variables, Definición Conceptual y Operacional e Indicadores 41
3.8 Estructura Desagregada de Trabajo 43
3.9 Aspectos Éticos 45
3.10 Cronograma 45
3.11 Recursos 45
CAPÍTULO IV: ANÁLISIS E INTERPRETACIÓN DE RESULTADOS 49
Desarrollo de los Instrumentos de Recolección de Datos 49
1.- Encuesta 49
2.- Guía de Observación 49
Validación de los Instrumentos de Recolección de Datos 50
Resultados de la Encuesta y Análisis de su Aplicación 50
Resultados de la Guía de Observación y Análisis de su Aplicación 61
CAPÍTULO V: DESARROLLO DE LA PROPUESTA 68
Características 68
Subcaracterísticas 70

Métricas 78
CAPÍTULO VI: EVALUACIÓN DE LA PROPUESTA..... 94
Matriz de evaluación de la calidad para la página web de la empresa Sanitas de Venezuela 94
Matriz de evaluación de la calidad para la página web de la empresa PlanSanitas de Venezuela 103
Análisis de la evaluación de la calidad..... 111
CAPÍTULO VII: CONCLUSIONES Y RECOMENDACIONES 118
REFERENCIAS BIBLIOGRÁFICAS 121
ANEXOS 125

INDICE DE FIGURAS

Figura **Pág.**

1. Características Modelo ISO/IEC 9126-1:2000 medición de calidad interna y externa. 23
2. Características Modelo ISO/IEC 9126-1:2000 medición de calidad de uso 26
3. Estructura Organizativa SUDEASEG 31
4. Sujetos regulados por la SUDEASEG 33
5.Estructura Desagregada de Trabajo 44
6.Cronograma del Proyecto Parte I 47
7.Cronograma del Proyecto Parte II 48
8.Resultado pregunta número 1 de la encuesta. 51
9.Resultado pregunta número 2 de la encuesta. 51
10.Resultado pregunta número 3 de la encuesta. 52
11.Resultado pregunta número 4 de la encuesta. 53
12.Resultado pregunta número 5 de la encuesta. 53
13.Resultado pregunta número 6 de la encuesta. 54
14.Resultado pregunta número 7 de la encuesta. 55
15.Resultado pregunta número 8 de la encuesta. 55
16.Resultado pregunta número 9 de la encuesta. 56
17.Resultado pregunta número 10 de la encuesta. 57
18.Resultado pregunta número 11 de la encuesta. 58
19.Resultado pregunta número 12 de la encuesta. 58
20.Resultado pregunta número 13 de la encuesta. 59
21.Resultado pregunta número 14 de la encuesta. 60
22.Resultado de la característica Funcionalidad de la guía de observación. 62
23.Resultado de la característica Fiabilidad de la guía de observación. 63
24.Resultado de la característica Eficiencia de la guía de observación. 65
25.Resultado de la característica Mantenibilidad de la guía de observación. 66
26.Resultado de la característica Portabilidad de la guía de observación. 67
27.Descripción del modelo propuesto 68

28.Características del modelo de evaluación de la calidad de producto de software propuesto. 69
29.Subcaracterísticas de la Arquitectura. 71
30.Subcaracterísticas del Control. 71
31.Subcaracterísticas de la Depuración. 72
32.Subcaracterísticas de la Formalidad. 73
33.Subcaracterísticas de la Eficacia. 73
34.Subcaracterísticas de la Integridad. 74
35.Subcaracterísticas de la Reusabilidad. 75
36.Subcaracterísticas de la Comprensibilidad. 75
37.Subcaracterísticas de la Eficiencia. 76
38.Subcaracterísticas de la Mantenibilidad. 77
39.Subcaracterísticas de la Portabilidad. 77
40.Gráfico de los resultados de la aplicación del modelo propuesto en la página web de la empresa Sanitas de Venezuela. 116
41.Gráfico de los resultados de la aplicación del modelo propuesto en la página web de la empresa PlanSanitas de Venezuela. 117

INDICE DE TABLAS

Tabla **Pág.**

1. Características y Subcaracterísticas Modelo ISO/IEC 9126-1:2000 medición de calidad interna y externa. 24
2. Medicina Prepagada 32
3. Técnicas de Recolección de Datos 39
4. Operacionalización de las variables 42
5. Recursos necesarios para llevar a cabo la investigación 46
6. Tipos de métricas del modelo propuesto 78
7. Métricas de la subcaracterística incrementalidad del modelo propuesto 79
8. Métricas de la subcaracterística abstracción del modelo propuesto 79
9. Métricas de la subcaracterística modularidad del modelo propuesto 80
10. Métricas de la subcaracterística revisiones técnicas del modelo propuesto 80
11. Métricas de la subcaracterística Seguimiento funcional del modelo propuesto 81
12. Métricas de la subcaracterística funcional del modelo propuesto 81
13. Métricas de la subcaracterística unitaria del modelo propuesto 82
14. Métricas de la subcaracterística rendimiento del modelo propuesto 82
15. Métricas de la subcaracterística seguridad del modelo propuesto 82
16. Métricas de la subcaracterística documentación mantenimiento del modelo propuesto 83
17. Métricas de la subcaracterística documentación técnica del modelo propuesto 83
18. Métricas de la subcaracterística resultados de pruebas del modelo propuesto 84
19. Métricas de la subcaracterística estabilidad del modelo propuesto 84
20. Métricas de la subcaracterística adecuación del modelo propuesto 85
21. Métricas de la subcaracterística conformidad del modelo propuesto 85
22. Métricas de la subcaracterística confidencialidad del modelo propuesto 86
23. Métricas de la subcaracterística disponibilidad del modelo propuesto 86

24. Métricas de la subcaracterística auditoría de acceso del modelo propuesto .. 86
25. Métricas de la subcaracterística generalidad del modelo propuesto 87
26. Métricas de la subcaracterística independencia de módulos del modelo propuesto 87
27. Métricas de la subcaracterística Atractividad del modelo propuesto 88
28. Métricas de la subcaracterística aprendizaje del modelo propuesto 88
29. Métricas de la subcaracterística operatividad del modelo propuesto 88
30. Métricas de la subcaracterística validaciones para errores de usuarios del modelo propuesto..... 89
31. Métricas de la subcaracterística documentación de usuario del modelo propuesto 89
32. Métricas de la subcaracterística uso de recursos del modelo propuesto 90
33. Métricas de la subcaracterística tiempo de respuesta del modelo propuesto . 90
34. Métricas de la subcaracterística anticipación al cambio del modelo propuesto 91
35. Métricas de la subcaracterística facilidad de cambio del modelo propuesto ... 91
36. Métricas de la subcaracterística estabilidad del modelo propuesto..... 92
37. Métricas de la subcaracterística facilidad de diagnóstico del modelo propuesto 92
38. Métricas de la subcaracterística adaptabilidad del modelo propuesto 92
39. Métricas de la subcaracterística interoperabilidad del modelo propuesto 93
40. Resultados aplicación del modelo propuesto a la página web de la empresa Sanitas de Venezuela 112
41. Resultados aplicación del modelo propuesto a la página web de la empresa PlanSanitas de Venezuela 113

INTRODUCCIÓN

La ingeniería de software, como área de estudio de la tecnología de la información y bajo su perspectiva disciplinada y sistematizada, involucra la calidad del producto de software como la capacidad que poseen los sistemas para satisfacer las necesidades de los usuarios y cumplir los requerimientos para los cuales fue creado.

Debido a las grandes ventajas que esta proporciona, tales como disminución de costos, optimización del tiempo de desarrollo, producto confiable y un gran valor agregado a las características del mismo, los profesionales dedicados al desarrollo de aplicaciones se han interesado en indagar en el mundo de la calidad.

Los modelos de evaluación aparecen con el objetivo de definir estándares y características que los sistemas de información deben cumplir para garantizar su calidad. Durante los últimos años se han propuesto diversos tipos de ellos.

Los primeros avances en esta materia fueron elaborados por Watts Humphrey, en el *Carnegie Mellon Software Engineering Institute (SEI)*, motivo por el cual es conocido como el padre de la calidad del software (Carnegie Mellon University, 2016).

Los patrones definidos, algunos de ellos internacionalmente, enfocan la aplicación de sus métricas en las fases finales de la construcción del software, impidiendo en muchos casos que se realicen modificaciones sobre el producto, debido a sus grandes implicaciones, las cuales afectan el tiempo de culminación del proyecto, y por ende aumentan sus costos, alejándose de esta forma de los beneficios que la calidad por si sola otorga.

En este sentido, organizaciones dedicadas al desarrollo de sistemas tecnológicos se abstienen de incluir en la planificación de sus proyectos el aseguramiento de la calidad del producto final, debido a las complicaciones que se presentan al intentar adaptar las características de evaluación de los estándares al software desarrollado, así como afrontar las diversas interpretaciones particulares de los profesionales dedicados a esta área, basados en los modelos existentes.

Guilarte (2013), en su artículo referente a la celebración de la Jornada de Calidad de Producto de Software cita unas palabras de Jesús Hernando

Corrochano (responsable del Grupo de Ingeniería de Software en El Corte Inglés), haciendo referencia a este hecho:

> El 80 % del software que se produce tiene deficiencias de calidad, lo que supone importantes perjuicios clientes y usuarios y menoscaba la confianza que se tiene en los proveedores. Debemos ir a un modelo orientado al producto no al proyecto, basado en una calidad concertada, que contemple pruebas de calidad en todo el ciclo de desarrollo. No se puede esperar a que el producto esté acabado, para verificar que funciona.

Por tales motivos, se planteó como objetivo general de esta investigación proponer un modelo de evaluación de la calidad del producto de software, que solvente estas complicaciones. El mismo se diseñó y validó en el sector de medicina prepagada en Venezuela.

El presente documento está estructurado de la siguiente forma:

A. Capítulo I. El Problema: Caracterizado por describir la problemática abordada en esta investigación, su importancia y los objetivos que se desean cumplir así como su alcance.
B. Capítulo II. Marco Teórico: Establece leyes, antecedentes y todos los fundamentos teóricos que avalan y sustentan científicamente esta investigación. Finalmente se hace mención al marco referencial en el cual está orienta.
C. Capítulo III. Marco Metodológico: Conformado por los componentes que orientan el trabajo de investigación al cumplimiento de los objetivos planteados en el capítulo I. Incluye el diseño, tipo y fases de la investigación, población, muestra, definición de variables junto a su operacionalización e indicadores, estructura desagregada de trabajo, los aspectos éticos, recursos requeridos y finalmente el cronograma de desarrollo de las actividades a llevar a cabo.
D. Capítulo IV. Análisis e Interpretación de resultados: Contiene la información referente al proceso de recolección de datos a través de los instrumentos diseñados, su validación, aplicación y análisis de los resultados obtenidos, los cuales son parte del levantamiento de

información requerido para el desarrollo de la propuesta de este proyecto de investigación.

E. Capítulo V. Desarrollo de la propuesta: Se presenta el modelo de evaluación de la calidad de producto de software para el sector de medicina prepagada, diseñado en base a la necesidad encontrada en la información recaudada durante la investigación.
F. Capítulo VI. Evaluación de la propuesta: Muestra los resultados obtenidos de la evaluación de la calidad del producto de software, a través del modelo diseñado y propuesto.
G. Capítulo VII. Conclusiones y Recomendaciones: Comprende el conjunto de lecciones aprendidas, conocimientos adquiridos, resultados y sugerencias, obtenidos a partir de llevar a cabo el proyecto de investigación

Por último se presentan las referencias bibliográficas consultadas para elaborar el presente proyecto.

CAPITULO I: EL PROBLEMA

En el presente capítulo se describe la situación que da origen al problema tratado en esta investigación. Se definen los logros que se desean alcanzar a través de los objetivos, además señalar la importancia y motivos por los cuales se llevó a cabo, expresado en la sección denominada justificación. Finalmente se denotan las limitaciones que se presentaron durante el trabajo de grado así como las delimitaciones y hasta donde abarcó, aspectos que definieron en el alcance

1.1 Planteamiento del Problema

En la actualidad, las empresas dedicadas al desarrollo de aplicaciones (incluso aquellas donde la tecnología no representa su modelo de negocio) han aumentado su interés en evaluar y asegurar la calidad de sus productos de software. Acerca de esto, Mendoza, Pérez, Grimán, & Rojas expresan:

> La necesidad de garantizar Sistemas de Software de calidad en una competencia abierta y global ha motivado que a nivel mundial se propongan un conjunto de modelos para evaluar su calidad. Los Sistemas de Software pueden ser considerados como productos o servicios que responden a las características propias de cada organización y/o negocio y a las necesidades de sus consumidores. Un modelo de calidad representa estos requerimientos y necesidades. Junto a su formulación se deben también desarrollar los pasos a seguir para su aplicación. (p.1)

Algunos beneficios de la evaluación de la calidad de producto de software son:

- Incremento del precio final en caso que los productos desarrollados sean para la venta.
- Alta reputación y confiabilidad en el uso de software que cumplan con normativas de calidad.
- Reconocimiento para la organización que llevó a cabo los proyectos.

- En los casos de aplicaciones desarrolladas para satisfacer una necesidad interna de la empresa, asegura, entre otros aspectos, el correcto desempeño funcional del sistema.

Sin embargo los estándares existentes que otorgan estas mejoras están diseñados para ser aplicados en fases finales del desarrollo, teniendo como consecuencia una detección y corrección de errores tardías con grados de implicación máximos o inaceptables, obteniendo en muchos casos el fracaso del proyecto, o bien desistir su medición de calidad, es decir, las características evalúan el producto final, y no los entregables en las fases de diseño o construcción de software.

Esto se debe a que los modelos existentes se enfocan en la medición de la mantenibilidad como característica principal de un sistema, además las métricas y métodos de evaluación propuestos se orientan a la idea más general de software, visualizándolo como un todo y no el conjunto de sus partes, es decir, no permite ofrecer soluciones a casos concretos o particulares, denominados módulos, lo cual impide la evaluación y aplicación a grandes cantidades de casos de estudio.

Acerca de este tema, Moreno, Bolaños, & Navia (2010) afirman lo siguiente:

> La importancia de la calidad de software en la disciplina de la Ingeniería del software es ampliamente reconocida en la actualidad, sin embargo desde el punto de vista de los modelos y estándares hacia el producto, el desarrollo de estos durante décadas, la sobreabundancia de información, el alto costo y el acceso limitado a esta información, impiden un acercamiento de estos a los ingenieros de software en pro de la calidad del producto software al interior de la organización. (p.1).

Cuando de calidad de software se trata, éstos no son los únicos problemas con los que se puede topar una organización al querer evaluar la calidad de un proyecto, las interpretaciones particulares de las características y métricas de evaluación de los modelos existentes, como ISO/IEC 9126-1:2000, McCall, entre otros, limitan la definición de un criterio único de medición con el que se pueda cumplir el objetivo de evaluar la calidad en aplicaciones o sistemas.

Debido a las consideraciones mencionadas anteriormente, la situacion global en el tema de calidad de software se ve afectada, provocando que las organizaciones pierdan interés en este ámbito tan importante para la ingeniería de software. “Una de las principales debilidades de esta industria radica en la falta de orientación hacia el mercado extranjero, especialmente en lo referente a las exportaciones y a la carencia de estándares de calidad en los productos” (Figarella & Zamora, 2007, p.12).

La medicina prepagada, en Venezuela y en cualquier parte del mundo, requiere de los sistemas de información gerenciales para inteligencia de negocios, manejo de los datos, gestión de inventarios, recursos humanos, entre otros. Existen software que cumplen con los estándares de calidad internacionales y pueden satisfacer las necesidades de las organizaciones dedicadas a este sector (ejemplo de ello es SAP), sin embargo su adquisición implica un alto costo que no toda empresa puede permitirse. “En Venezuela no existían desarrollos informáticos expresamente elaborados para el mercado de empresas de salud que atendieran consumidores de bajos ingresos, de manera que confeccionar una aplicación propia se presentaba como una opción de peso.” (Viana, Gomes, & Jaén, 2008, p.9).

De acuerdo a lo anteriormente expuesto se plantea el siguiente conflicto ¿Cómo desarrollar un software en el sector de medicina prepagada que cumpla con estándares de calidad internacionales? En este ámbito la limitante del costo de las asesorías en esta materia pueden llegar a ser muy elevado, y adentrarse sin consulta profesional se convierte una tarea difícil debido a la falta de orientación y pocas herramientas en el mercado que faciliten este trabajo. Algunas organizaciones, en busca de soluciones, optaron por asesoría extranjera. Figarella y Zamora (2007) narran lo siguiente: “Las empresas que estaban interesadas en obtener una certificación avalada internacionalmente tenían que dirigirse a instituciones especializadas fuera de las fronteras venezolanas y hacer una inversión bastante alejada de sus posibilidades económicas” (p.8).

Todos estos escenarios traen las siguientes consecuencias:

- Grandes inversiones monetarias.

- Retrasos en la culminación de los proyectos tecnológicos.
- Modelos y normativas aplicados a media, ya que estos no se adaptan a la realidad del proceso de desarrollo de software o bien a diversas metodologías.
- Interpretaciones particulares debido al vacío de orientación, lo que conlleva a una evaluación errada, evitando certificaciones internacionales.
- Las empresas desisten de continuar con la evaluación de calidad de producto de software de sus proyectos.

Finalmente, con el modelo diseñado en esta investigación pretende dar solución a gran parte de los obstáculos que enfrentan las empresas al evaluar la calidad de sus productos de software, entre los cuales se encuentran:

- Evaluación de las características de la calidad en etapas tardías.
- Interpretaciones particulares en la aplicación de las métricas de los estándares internacionales.
- Características que abarquen todo el ciclo de desarrollo.
- Presentar una solución para la evaluación de calidad de producto de software de fácil aplicación y sin invertir en asesorías costosas.

1.1.1 Formulación del Problema

De acuerdo a la descripción del problema, se plantea la siguiente pregunta de investigación:

- ¿Cómo contribuirá el modelo planteado en esta investigación a solucionar los problemas de evaluación de calidad de producto de software en el sector de medicina prepagada?

1.1.2 Sistematización del Problema

A continuación se plantean las preguntas que constituyen el problema de esta investigación.

- ¿Cuáles son las características de los modelos para evaluar la calidad del producto de software orientado al sector de medicina prepagada?
- ¿Cómo estaría caracterizado un modelo de evaluación de calidad de producto de software para el sector de la medicina prepagada?
- ¿Cómo sería el diseño de las etapas del modelo de evaluación de calidad de producto de software para el sector de la medicina prepagada?
- ¿Qué pasos se debe seguir para validar un modelo de evaluación de la calidad de producto de software dentro del sector de medicina prepagada?

1.2. Objetivos

Los objetivos que guiarán el alcance de este trabajo de grado son los siguientes:

1.2.1 Objetivo General

Diseñar un modelo para la evaluación de la calidad del producto de software para medicina prepagada

1.2.2 Objetivos Específicos

- Describir los modelos existentes en el mercado para la evaluación de la calidad de software.
- Caracterizar los elementos del modelo de la calidad de software para medicina prepagada
- Diseñar las características, subcaracterísticas y métricas del modelo de evaluación de la calidad del producto de software adaptado a las necesidades del mercado.
- Validar el modelo de evaluación diseñado.

1.3. Justificación de la Investigación

El diseño del modelo propuesto para la evaluación de la calidad del producto de software busca otorgar una solución a aquellas empresas dedicadas a la medicina prepagada que deseen incursionar en el ámbito de la calidad del software (así como proveedores encargados de desarrollar sistemas para este sector), permitiendo cumplir con estándares internacionales de calidad sin requerir la contratación de asesorías costosas, evitando generar retrasos en la culminación de proyectos, y llevar a cabo mediciones erradas a través de las normativas existentes, debido a mala interpretación o baja adaptabilidad a los desarrollos.

Aplicar este modelo en los sistemas o aplicaciones garantiza los siguientes aspectos:

- Adaptabilidad al desarrollo basado en componentes, por lo que el tiempo de construcción del software disminuye, ya que apoya la reutilización de código.
- Evaluaciones en etapas donde las implicaciones o modificaciones a realizar generen poco impacto en los tiempos del proyecto.
- Aclarar las dudas más frecuentes que presentan los líderes de proyectos de tecnología acerca de cómo abordar las características de medición de la calidad del producto, evitando las inferencias particulares y mal manejo de las métricas.
- Es un modelo basado en los estándares de calidad existentes y aprobados internacionalmente, como por ejemplo ISO/IEC 9126-1:2000.
- Mediciones de calidad en productos desarrollados dentro de las organizaciones o adquiridos a proveedores.
- Disminución de esfuerzos, tiempo y dinero en lograr un repositorio de aplicaciones con evaluaciones de calidad aprobadas

Adicionalmente proporciona patrones que permiten llevar a cabo con mayor facilidad las tareas de evaluación, y representa un aporte a la ingeniería de software (pues el modelo a diseñar cumple con los principios que la rigen), así como fomentar la inclusión de organizaciones ubicadas en la base de la pirámide empresarial, cuyo acercamiento al tema de calidad de software no es común

1.4. Alcance y limitaciones de la Investigación

La presente investigación está orientada a un tipo de estudio de alcance descriptivo, cuyo objetivo principal de acuerdo a Hernández, Fernández, & Baptista (2010) es:

> Especificar las propiedades, las características y los perfiles de personas, grupos, comunidades, procesos, objetos o cualquier otro fenómeno que se someta a un análisis. Es decir, únicamente pretenden medir o recoger información de manera independiente o conjunta sobre los conceptos o las variables a las que se refieren, esto es, su objetivo no es indicar cómo se relacionan estas (p.80).

En este margen de ideas, el modelo de evaluación a diseñar pretende únicamente medir la variable calidad en los software, basado en normativas internacionales como ISO/IEC 9126-1:2000, esto dentro del marco de la comunidad de desarrolladores de aplicaciones o sistemas. Será aplicado y validado en la empresa Sanitas Venezuela, organización perteneciente al sector de medicina prepagada.

El tiempo estimado para el diseño de este modelo de evaluación es de seis meses.

Se debe contar con un conjunto de colaboradores para poder llevar a cabo la validación del modelo, entre los cuales se encuentran: arquitecto de software, auditor de calidad de software, líder de proyecto de desarrollo.

En cuanto a las limitaciones para lograr los objetivos del estudio se encuentran las siguientes:

- Acceso a la información referente a los sistemas de información y su calidad en empresas de medicina prepagada diferentes a Sanitas y PlanSanitas.
- Falta de empresas que apliquen los estándares de calidad de producto de software en el país que puedan aportar experiencias a la investigación.

CAPÍTULO II: MARCO TEÓRICO

El siguiente capítulo constituye las bases teóricas de la investigación. Su estructura va desde orientaciones y soportes que anteceden a este trabajo de grado hasta teorías, sustentos, conocimientos que la avalan, divididos en: fundamentos teóricos, marco referencial, bases legales y definición de términos.

2.1 Antecedentes

A continuación se detallan los antecedentes de esta investigación.

Trabajos de Grado de Maestría

1.- Mendoza (2010). *Modelo de referencia para la selección de herramientas de pruebas como soporte al proceso de desarrollo de software en PYMES venezolanas*. El objetivo de esta investigación de tipo aplicada fue proponer un modelo de selección de herramientas que permitan hacer efectivo la fase de pruebas en el proceso de desarrollo de software, permitiendo que al menos el 75% de las actividades se puedan automatizar, esto con la finalidad de disminuir costos y tiempo de los proyectos y adicionalmente garantizar calidad en el producto final. Este trabajo de grado se basa principalmente en las PYMES desarrolladoras de software en Venezuela, y fue elaborado bajo los pasos y actividades definidos por el framework metodológico LISI.

Este modelo no solo define un patrón o referencia que debe ser tomado en cuenta para elegir una herramienta robusta y efectiva, también puede ser utilizada como guía para construir un software adaptado a las necesidades de la empresa que así lo requiera.

Aporte: Proporciona información acerca de la importancia de la fase de pruebas en el proceso de desarrollo de software y herramientas que permiten agilizarlas, adicionalmente tiene un fin en común con este trabajo de grado, la calidad del

producto de software. Además representa un gran aporte bibliográfico ya que el estudio se realizó bajo el mismo mercado tecnológico que es Venezuela.
Palabras clave: calidad en el producto, software, PYMES, LISI, framework.

Artículos Técnicos

1.- Moreno, Bolaños y Navia (2010). *Exploración de Modelos y Estándares de calidad para el producto software*. Los autores plantean un artículo de tipo bibliográfico, donde hacen mención de los principales modelos propuesto en el mercado para la evaluación de la calidad del producto de software, los aportes de cada uno y aquellos aspectos que todos tienen en común y según su criterio se deben consolidar.
Aporte: Proporciona un acercamiento a los principales modelos de evaluación de calidad de software (McCall, Boehm, FURPS, ISO 9126,Dromey, SQAE, Bansiya y Quint2), la interacción entre ellos, métricas, características y subcaracterísticas de cada uno.
Palabras clave: Evaluación, modelo, calidad.

2.- Sánchez (2011). *La usabilidad en Ingeniería de Software: definición y características*. Describe una de las características de la norma ISO/IEC 9126 de gran importancia, que permite medir la capacidad que posee un software de ser intuitivo, eficiente y amigable para sus usuarios, denominada usabilidad.
Aporte: Presenta aportes teóricos de la usabilidad en el software, importancia, beneficios, características y modelos que hacen referencia a la usabilidad como una característica que se debe medir para garantizar calidad.
Palabras clave: intuitivo, eficiente, amigable, usabilidad.

3.- Pentón (2013). *Métrica para evaluar la seguridad de los SGIC*. Hace referencia a un modelo que permite seleccionar de la gama de Sistemas de Gestión de Información Contable (SGIC), el que mejor se adapte a los requerimientos de los usuarios, apoyado por las métricas definidas en las normas como ISO/IEC 9126

así como en algoritmos robustos que determinan el comportamiento de los paquetes.

Aporte: Aplicación de las métricas de evaluación de calidad en modelos reconocidos como ISO/IEC 9126-1:2000 en el ámbito de implementación de Sistemas de Gestión de Información Contable – Financiera.

Palabras clave: SGIC, ISO/IEC 9126-1:2000, algoritmos, modelo.

4.- Ortega, Villavicencio. *Medición de la Calidad de Productos de Software en un Ambiente Académico Usando la Norma ISO/IEC 9126*. Este artículo técnico hace referencia a la evaluación de calidad de productos de software con la norma ISO/IEC 9126-1:2000, la particularidad es el ambiente en el que se desenvuelve el proyecto, ya que es del tipo académico. Describen brevemente los software a los cuales se les aplicó el estudio, en total fueron 11, desarrollados para distintos contextos y por 41 estudiantes de la materia ingeniería de software II.

Apegados al estándar ISO/IEC 9126, la evaluación de calidad de los productos se realizó con las siguientes características: Eficiencia, Usabilidad, Facilidad de mantenimiento, Fiabilidad, Funcionalidad, las cuales describen la calidad interna y externa según ISO/IEC 9126-2 e ISO/IEC 9126-3 respectivamente.

Adicionalmente se evaluaron los aspectos descritos en la ISO/IEC 9126-4 como son: satisfacción y eficiencia, referentes a la calidad de uso.

Aporte: Además de contribuir con aportes teóricos al modelo propuesto en este trabajo de grado, brinda un caso exitoso de aplicación de normas ISO/IEC 9126-1:2000 en proyecto de desarrollo de software, del cual se pueden tomar tanto experiencias como datos bibliográficos, sin importar que el ámbito de aplicación sean distintos, ya que en el artículo hacen mención del alcance hacia las empresas.

Palabras clave: Eficiencia, Usabilidad, Facilidad de mantenimiento, Fiabilidad, Funcionalidad, satisfacción, eficiencia.

5.- Bertoa, Troya, Vallecillo. *Aspectos de Calidad en el Desarrollo de Software Basado en Componentes*. Incursiona en la disciplina de Desarrollo de Software

basado en componentes, la cual la define como: “unidades de software ya desarrollados, que son combinados adecuadamente para satisfacer los requisitos del sistema”. El entorno de calidad viene a jugar un papel importante en la selección de estos componentes, y son llamados “extra-funcionales”, y son un aspecto a tomar en cuenta en el proceso de discriminación de componentes candidatos, siendo la principal característica a tomar en cuenta la funcionalidad del mismo.

La propuesta de calidad de los componentes del software toma en cuenta que la calidad de un producto de software no se refiere únicamente a obtener un producto sin errores, por lo que involucra los siguientes apartados:

- Calidad en el producto: Basados en los estándares ISO-9126-1:2000 e ISO-14598, así como el tratamiento que da Bass, Jan Bosch y Preiss.
- Calidad en el proceso: El proceso de construcción de un producto software basado en las características definidas en los trabajos de: COCOTS, QESTA, COTS-Based Requirements Engineering (CRE), Off-The-Shelf Option (OTSO).

-Procurement-Oriented Requirements Engineering (PORE): es una propuesta basada en plantillas (templates). Utiliza un proceso iterativo de captura de requisitos y de selección y evaluación de productos

Aporte: Realiza el aporte teórico más importante de todos los antecedentes ya que hace mención a las métricas de evaluación de calidad relativas a los componentes que van a constituir un nuevo software de manera más eficiente.

Palabras clave: sistema, calidad, componentes.

6.- Mendoza, Pérez, Grimán, Rojas. *Algoritmo para la Evaluación de la Calidad Sistémica del Software*. En este artículo muestra un algoritmo basado en el modelo sistemático de calidad (MOSCA) que permite estimar la calidad del software. Esta herramienta, los autores la definen como “efectiva de análisis y estimación de la Calidad Global Sistémica”, ya que involucra el proceso de construcción del software, el producto final, la relación existente entre ellos y finalmente con el medio ambiente.

Aporte: Además de los aportes teóricos como lo son el modelo MOSCA, calidad global sistemática y el algoritmo propuesto en este artículo, muestra dos casos de estudios en el mismo mercado de este trabajo de grado, empresas venezolanas.
Palabras clave: MOSCA, modelo sistemático, estimar, Calidad Global Sistémica.

2.2. Fundamentos Teóricos

En esta sección del documento se hace referencia a todas aquellas teorías, leyes y fundamentos referentes al problema planteado en el capítulo anterior. “La perspectiva teórica proporciona una visión de dónde se sitúa el planteamiento propuesto dentro conocimiento en el cual nos “moveremos”” (Hernández, Fernández, & Baptista, 2010, pág. 52).

Los aspectos teóricos fundamentales para que sustentaran el desarrollo de los objetivos de esta investigación se presentan a continuación.

2.2.1. Ingeniería de software

Antes de entrar en el tema de la calidad de software, es de gran importancia conocer acerca del área de sistemas encargada de desarrollar y mantener aplicaciones con estas características. Acerca de esto Sommerville,(2005) indica:

> La ingeniería de software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza. En esta definición, existen dos frases claves:
>
> 1. Disciplina de la ingeniería. Los ingenieros hacen que las cosas funcionen. Aplican teorías, métodos y herramientas donde sean convenientes, pero las utilizan de forma selectiva y siempre tratando de descubrir soluciones a los problemas, aun cuando no existan teorías y métodos aplicables para resolverlos. Los ingenieros también saben que deben trabajar con restricciones financieras y organizacionales, por lo que buscan soluciones tomando en cuenta estas restricciones.
> 2. Todos los aspectos de la producción de software. La ingeniería del software no sólo comprende los procesos técnicos del desarrollo de software, sino también con actividades tales como la gestión de proyectos de software y el desarrollo de herramientas, métodos y teorías de apoyo a la producción de software. (p. 23)

La ingeniería de software se involucra en el ciclo de vida del sistema, es decir, participa en el análisis, diseño, elaboración, pruebas, implementación, e incluso el mantenimiento. Su aplicación en cada uno de estos procedimientos se realiza buscando los mejores resultados posibles y un correcto funcionamiento del software, así como optimizar los tiempos de desarrollo.

La relación existente entre calidad del software y el desarrollo de ellos viene dado por los métodos de la ingeniería de software. “Un método de ingeniería del software es un enfoque estructurado para el desarrollo de software cuyo propósito es facilitar la producción de software de alta calidad de una forma costeable” (Sommerville, 2005)

2.2.2. Calidad de software

Esta característica del software ha adquirido gran importancia en el mercado tecnológico. Con respecto a este tema, Garzón, Sampalo, & José, (2003) expresan lo siguiente:

> La calidad del software es un término bastante difícil de definir, es un concepto un poco escurridizo. Cada autor lo enfoca de una forma diferente, en general todos coinciden en que no es sólo un factor el que indica la calidad del mismo, sino que es una mezcla de muchos factores. (p.13)

Ahora bien, una definición para la calidad de software es: “Proceso eficaz de software que se aplica de manera que crea un producto útil que proporciona valor medible a quienes lo producen y a quienes lo utilizan” (Pressman, 2010).

En todo proceso de creación de un producto, incluso en los sistemas, se proyecta su obtención con el nivel de calidad más alto posible, y esto se mide a través de la satisfacción del usuario, es decir, que satisfaga sus necesidades, sea de fácil manejo, confiable, seguro y garantice integridad en su funcionamiento.

Como se puede observar, el concepto de calidad no está asociado a un sistema sin fallas, ya que un desarrollador no podría garantizar esta condición.

De la definición Pressman (2010), amplía tres puntos importantes:

> 1. Un proceso eficaz de software establece la infraestructura que da apoyo a cualquier esfuerzo de elaboración de un producto de software de alta calidad. Los aspectos de administración del proceso generan las

> verificaciones y equilibrios que ayudan a evitar que el proyecto caiga en el caos, contribuyente clave de la mala calidad. Las prácticas de ingeniería de software permiten al desarrollador analizar el problema y diseñar una solución sólida, ambas actividades críticas de la construcción de software de alta calidad. Por último, las actividades sombrilla, tales como administración del cambio y revisiones técnicas, tienen tanto que ver con la calidad como cualquier otra parte de la práctica de la ingeniería de software.
> 2. Un producto útil entrega contenido, funciones y características que el usuario final desea; sin embargo, de igual importancia es que entrega estos activos en forma confiable y libre de errores. Un producto útil siempre satisface los requerimientos establecidos en forma explícita por los participantes. Además, satisface el conjunto de requerimientos (por ejemplo, la facilidad de uso) con los que se espera que cuente el software de alta calidad.
> 3. Al agregar valor para el productor y para el usuario de un producto, el software de alta calidad proporciona beneficios a la organización que lo produce y a la comunidad de usuarios finales. La organización que elabora el software obtiene valor agregado porque el software de alta calidad requiere un menor esfuerzo de mantenimiento, menos errores que corregir y poca asistencia al cliente. Esto permite que los ingenieros de software dediquen más tiempo a crear nuevas aplicaciones y menos a repetir trabajos mal hechos. La comunidad de usuarios obtiene valor agregado porque la aplicación provee una capacidad útil en forma tal que agiliza algún proceso de negocios. El resultado final es 1) mayores utilidades por el producto de software, 2) más rentabilidad cuando una aplicación apoya un proceso de negocios y 3) mejor disponibilidad de información, que es crucial para el negocio. (p. 340-341)

Garzón, Sampalo, & José (2003) indican que un software es considerado de calidad si cumple con los tres objetivos siguientes:

> 1. Concordancia del software con los requerimientos: el cliente desea que el software satisfaga una serie de requisitos o metas iniciales, y si ni si quiera alcanzamos estos objetivos, nuestro software carecerá por completo de calidad.
> 2. Desarrollo coherente, aplicando correctamente los criterios de la ingeniería del software: uno de los objetivos de la ingeniería del software es mejorar la calidad. Luego, está claro que debemos seguir una metodología correcta y apropiada a nuestro proyecto, si queremos aumentar la calidad del resultado final.
> 3. Desarrollo de requerimientos implícitos al proyecto: siempre existen una serie de requerimientos que nuestro cliente no específica, pero que son deseables. Por ejemplo, que nuestro software sea fácil de mantener, que sea fácil de usar, etc. Si no se alcanzan estos requerimientos, nuestro software carecerá de calidad. (p. 13)

La calidad de un producto de software no es un capricho de las empresas o de los líderes de proyectos. El cumplimiento de los estándares de calidad permiten asegurar que el sistema diseñado será creado con planificación de tiempo y costo óptimo, cumplirá con los objetivos para los cuales será creado, tendrá características como: portable, seguro, íntegro, eficiente, y además s mantenimiento no implica una gran inverso de tiempo y dinero para las organizaciones.

2.2.2.1 Tipos de calidad de software

La calidad de software está constituida por tres tipos fundamentales:

a) Calidad del producto.
b) Calidad del proceso.
c) Calidad del servicio.

"Diferenciamos claramente la calidad del Producto Proceso y la del Servicio. Las principales iniciativas, se centran en Proceso (CMMi, ISO/IEC 12207, ISO 15504-SPICE), y las de Servicio (ISO 20000)" (Hernando, 2013).

En esta investigación se tratará la calidad del producto de software; acerca de esto Hernando (2013) explica:

> La calidad es una cualidad esencial de cualquier producto generado por una organización, que va a ser usado por otros. Es por ello que una de las actividades principales de la elaboración de un producto, es el aseguramiento de la calidad: Plan de Aseguramiento de la Calidad del Producto (SQAP). Al igual que el conjunto de requisitos que establecen las funcionalidades y los límites del producto, el SQAP describe los umbrales de calidad asociados al producto, y marca los axiomas/límites de aceptación para su traspaso a Servicio Continuo. Es obvio que éstos dependen, en gran medida, del objetivo final del producto y del software del mismo: software de sistemas, tiempo real, gestión, ingeniería y científico, inteligencia artificial,..etc. El software, debe ser entendido como producto y como servicio; y, por ende, es fundamental el aseguramiento de la generación de Productos de Calidad, previo paso a transición a servicio continuo. (p.2)

Sommerville (2005) define una relación entre la calidad del producto y el proceso de la siguiente forma:

> Hay un vínculo claro entre la calidad de proceso y del producto en producción debido a que el proceso es relativamente fácil de estandarizar y monitorizar.
> Cada sistema de producción calibra, y debe producir una y otra vez productos de alta calidad. Si embargo, el software no se manufactura, sino que se diseña. El desarrollo de software es un proceso más creativo que mecánico, donde la experiencia y habilidades individuales son importantes. La calidad del producto, sea cual fuere el producto utilizado, también se ve afectada por factores externos, como la novedad de una aplicación o la presión comercial para sacar un producto rápidamente.
> En el desarrollo softwre, por lo tanto, la relación entre la calidad del proceso y la calidad del producto es muy complejo. Es difícil de medir los atributos de la calidad del software, como mantenibilidad, incluso después de utilizar el software durante un largo período. En consecuencia, es dificil explicar cómo influyen las características del proceso en estos atributos. Además debido al papel del diseño y la creatividad en el proceso software, no podremos predecir la influencia de los cambios en el proceso en la calidad del producto. (p. 590)

2.2.2.2 Garantía de calidad

Este término hace referencia a todos aquellos procedimientos, estándares y actividades diseñados por la organización con el objetivo de conducir el desarrollo de software a productos de calidad.

Acerca de este proceso, Sommerville (2005) expresa lo siguiente:

> La gestión de la calidad provee una comprobación independiente de los procesos de desarrollo de software. Los procesos de gestión de la calidad comprueban las entregas del proyecto para asegurarse que concuerdan con los estándares y metas organizacionales. El equipo de garantía de calidad debe ser independiente del equipo de desarrollo para que puedan tener una visión objetiva del software. Ellos transmitirán los problemas y las dificultades al gestor principal de la organización.
> Un equipo independiente debe ser responsable de la gestión de la calidad y debe informar al gestor del proyecto. El equipo de calidad no está asociado con ningún grupo de desarrollo, sino que tiene la responsabilidad de la gestión de la calidad en toda la organización. La razón de esto es que los gestores del proyecto deben mantener el presupuesto y la agenda. Si aparecen problemas, éstos pueden verse tentados a comprometer la calidad del producto para mantener su agenda. Un equipo independiente de calidad garantiza que los objetivos organizacionales y la calidad no sean comprometidos por consideraciones de presupuesto o agenda. (p. 589)

La calidad de los sistemas no solo depende de un proceso de construcción que cumpla con ciertos estándares y grandes planes de mantenimiento. El uso eficiente del software también forma parte del aseguramiento de la calidad. Garantizar que los datos que se ingresan y se procesan, asi como la aplicación de los estándares de calidad en los planes de mantenimiento correctivo, son parte de esta gestión.

Amo, Martínez, & Segovia (2005) decriben las actividades mas importantes en la garantía de la calidad del software de la siguiente forma:

> - Revisiones técnicas formales, consistentes en reuniones del personal técnico involucrado en el proyecto con el propósito de descubrir errores, verificar que el software se ajusta a los requisitos funcionales, garantizar que se ha desarrollado siguiendo ciertos estándares predefinidos y conseguir un desarrollo uniforme del mismo.
> - Prueba del software. Es un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación. El ingeniero de software crea una serie de casos de prueba con la intención de descubrir un error en el mismo. La prueba no asegura a ausencia de defectos, sólo puede demostrar que no se han descubierto defectos en el software.
> - Control de cambios. Una de las principales amenazas para la calidad de software proviene de los cambios realizados en el mismo, ya que son una fuente de errores. Es normal que con el tiempo se produzcan cambios en la configuración del software, constituida por los programas fuentes y ejecutables, la documentación tecnica y de usuario y las estructuras de los datos, ya que, a medida que pasa el tiempo, el usuario plantea otras necesidades y el informático tiende por su parte a optimizar el sistema. Todos estos cambios en la configuración del software han de estar controlados por una “actividad de control de cambios” que los aprueba. En los proyectos software es normal que, cuando se dstribuye a los clientes un producto de software, se instituya un sistema de control de cambios de la configuración.

Dependiendo de lo lejos que desee llegar la organización en materia de calidad, esta lista de actividades aumentará, así como su planificación.

2.2.2.3 Planificación de la calidad

La planificación de la calidad consiste en definir los estándares que se manejaran durante el proyecto y los entregables durante la construcción de software, de esta manera el desarrollo se realiza bajo un marco de trabajo.

Project Management Institute (2013), en el PMBOOK indica lo siguiente acerca de la planificación de la calidad:

> El beneficio clave de este proceso es que proporciona guía y dirección sobre cómo se gestionará y validará la calidad a lo largo del proyecto.(p.232).
> La planificación de la calidad debe realizarse en paralelo con los demás procesos de planificación del proyecto. Por ejemplo, los cambios propuestos en los entregables de cara a cumplir con las normas de calidad identificadas, pueden requerir ajustes en el costo o en el cronograma, así como un análisis de riesgo detallado del impacto en los planes.(p. 233)

Una correcta planificación de calidad asegura que el producto final será entregado en el tiempo estipulado, satisface las necesidades para las cuales fue creado, y adicionalmente se apega al diseño elaborado antes de su construcción, y finalmente los planes de mantenimiento no significarán un gran conflicto para la empresa

2.2.2.4. Control de la calidad

El proceso de control de calidad es aquel que se encarga de garantizar que los estándares y patrones establecidos en la organización se cumplan durante la producción del software. Acerca de ello, Cendejas (2014) indica lo siguiente:

> El control de la variación puede equipararse con el control de calidad. Esto involucra la serie de inspecciones, revisiones y pruebas empleadas a lo largo del proceso del software para garantizar que cada producto del trabajo satisfaga los requisitos que se le han asignado. El control de calidad incluye un bucle de retroalimentación con el proceso que creó el producto del trabajo. La combinación de mediciones retroalimentación permite afinar el proceso cuando los productos de trabajo creados fracasan en cuanto a satisfacer sus especificaciones. Un concepto clave del control de calidad es que todos los productos de trabajo tienen especificaciones definidas mensurables con las cuales se puede comparar la salida de cada proceso. Dicho bucle es esencial para minimizar los efectos producidos.

2.2.2.5. Costos de la calidad

Obtener un software de calidad en una organización requiere una gran inversión de tiempo y dinero, sin embargo tiene una alta compensación obtener un producto con esta característica.

Cendejas (2014) lo define de la siguiente manera

> El costo de la calidad incluye todos los costos que se generan o que demandan el desarrollo de las actividades relacionadas con la calidad. Los estudios de costo de la calidad se llevan a cabo para ofrecer una línea base e identificar oportunidades que reduzcan el costo de calidad y proporcionan una base que sirva de comparación. La base de normalización casi siempre es monetaria, ya que se tienen los datos necesarios para evaluar dónde se encuentran las oportunidades para mejorar los procesos, se puede evaluar el efecto de los cambios en términos monetarios

Pressman (2010) habla de los tipos de costo de la calidad indicando lo siguiente:

> El costo de la calidad puede dividirse en los costos que están asociados con la prevención, la evaluación y la falla.
> Los costos de prevención incluyen lo siguiente: 1) el costo de las actividades de administración requeridas para planear y coordinar todas las actividades de control y aseguramiento de la calidad, 2) el costo de las actividades técnicas agregadas para desarrollar modelos completos de los requerimientos y del diseño, 3) los costos de planear las pruebas y 4) el costo de toda la capacitación asociada con estas actividades. Los costos de evaluación incluyen las actividades de investigación de la condición del producto la “primera vez” que pasa por cada proceso.
> Los costos de falla son aquellos que se eliminarían si no hubiera errores antes o después de enviar el producto a los consumidores. Los costos de falla se subdividen en internos y externos.
> Se incurre en costos internos de falla cuando se detecta un error en un producto antes del envío.
> Los costos externos de falla se asocian con defectos encontrados después de que el producto se envió a los consumidores. (p. 346-347)

2.2.3. Estándares para la evaluación de la calidad de producto de software

2.2.3.1. ISO/IEC 9126-1:2000:

Norma internacional que permite evaluar la calidad del producto de software, desarrollado inicialmente en 1991. Está constituido por tres partes encargadas de medir calidad externa, interna y de uso.

Acerca de este modelo, Moreno, Bolaños, & Navia (2010) expresan lo siguiente:

> El estándar ISO 9126 presenta su primera versión en 1991, luego en 2001 es remplazado por ISO 9126:1 que además cuenta con tres ítems adicionales para ayudar a la mejora de la calidad del producto software (Métricas externas, Métricas internas, Métricas de calidad en uso). Además presenta una estrecha relación con el estándar ISO 14598:1. El estándar ISO-9126 define un modelo, basado en modelos ya existentes como McCall, Boehm y US Air Force.
> El estándar ISO 9126 presenta dos partes, el Modelo de calidad para calidad externa e interna, y el Modelo de calidad para calidad en uso. (p.44)

El proceso de medición interna y externa lo realiza a través de seis características: funcionalidad, confiabilidad, usabilidad, eficiencia, facilidad de recibir mantenimiento, portabilidad (Ver Figura #1).

Pressman (2010) las describe de esta forma:

> **Funcionalidad**. Grado en el que el software satisface las necesidades planteadas según las establecen los atributos siguientes: adaptabilidad, exactitud, interoperabilidad, cumplimiento y seguridad.
> **Confiabilidad**. Cantidad de tiempo que el software se encuentra disponible para su uso, según lo indican los siguientes atributos: madurez, tolerancia a fallas y recuperación.
> **Usabilidad**. Grado en el que el software es fácil de usar, según lo indican los siguientes subatributos: entendible, aprendible y operable.
> **Eficiencia**. Grado en el que el software emplea óptimamente los recursos del sistema, según lo indican los subatributos siguientes: comportamiento del tiempo y de los recursos.
> **Facilidad de recibir mantenimiento**. Facilidad con la que pueden efectuarse reparaciones al software, según lo indican los atributos que siguen: analizable, cambiable, estable, susceptible de someterse a pruebas.
> **Portabilidad**. Facilidad con la que el software puede llevarse de un ambiente a otro según lo indican los siguientes atributos: adaptable, instalable, conformidad y sustituible.(p.343)

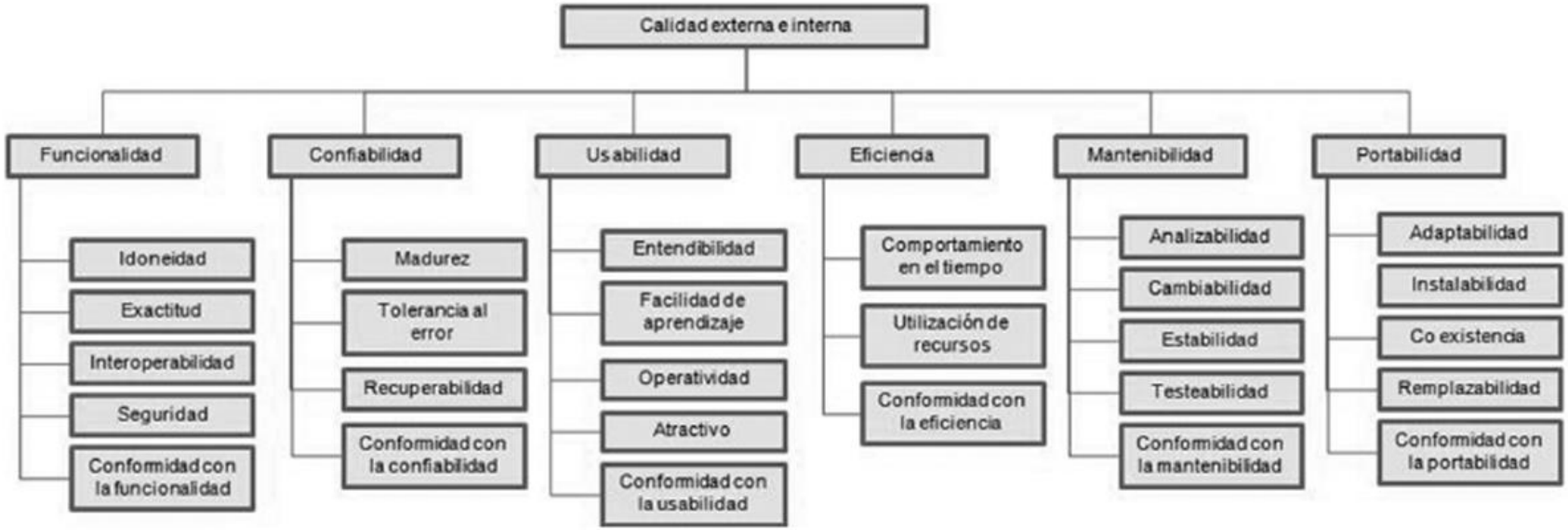

Figura #1. Características Modelo ISO/IEC 9126-1:2000 medición de calidad interna y externa.
Fuente: Moreno, Bolaños, & Navia, (2010).

Borbón (2013) hace enfásis en las subcaracterísticas de cada una de ellas, se resumen en la siguiente tabla:

Tabla #1. Características y Subcaracterísticas Modelo ISO/IEC 9126-1:2000 medición de calidad interna y externa.

Característica	Subcaracterísticas	Descripción
Funcionalidad	Adecuación	La capacidad del software para proveer un adecuado conjunto de funciones que cumplan las tareas y objetivos especificados por el usuario.
	Exactitud	La capacidad del software para hacer procesos y entregar los resultados solicitados con precisión o de forma esperada.
	Interoperabilidad	La capacidad del software de interactuar con uno o más sistemas específicos.
	Seguridad	La capacidad del software para proteger la información y los datos de manera que los usuarios o los sistemas no autorizados no puedan acceder a ellos para realizar operaciones, y la capacidad de aceptar el acceso a los datos de los usuarios o sistemas autorizados
	Conformidad de la funcionalidad	La capacidad del software de cumplir los estándares referentes a la funcionalidad.
Confiabilidad	Madurez	La capacidad que tiene el software para evitar fallas cuando encuentra errores
	Tolerancia a errores	La capacidad que tiene el software para mantener un nivel de funcionamiento en caso de errores.
	Recuperabilidad	La capacidad que tiene el software para restablecer su funcionamiento adecuado y recuperar los datos afectados en el caso de una falla.
	Conformidad de la fiabilidad	La capacidad del software de cumplir a los estándares o normas relacionadas a la fiabilidad.
Usabilidad	Entendimiento	La capacidad que tiene el software para permitir al usuario entender si es adecuado, y de una manera fácil como ser utilizado para las tareas y las condiciones particulares de la aplicación. En este criterio se debe tener en cuenta la documentación y de las ayudas que el software entrega.
	Aprendizaje	La forma como el software permite al usuario aprender su uso. También es importante considerar la documentación
	Operabilidad	La manera como el software permite al usuario operarlo y controlarlo.
	Atracción	La presentación del software debe ser atractiva al usuario. Esto se refiere a las cualidades del software para hacer más agradable al usuario, ejemplo, el diseño gráfico.
	Conformidad de uso	La capacidad del software de cumplir los estándares o normas relacionadas a su usabilidad.

Fuente: Borbón, N. (2013).

Tabla # 1. Características y Subcaracterísticas Modelo ISO/IEC 9126-1:2000 medición de calidad interna y externa. (Cont.)

Característica	Subcaracterísticas	Descripción
Eficiencia	Comportamiento de tiempos	Los tiempos adecuados de respuesta y procesamiento, el rendimiento cuando realiza su función en condiciones específicas
	Utilización de recursos	La capacidad del software para utilizar cantidades y tipos adecuados de recursos cuando este funciona bajo requerimientos o condiciones establecidas
	Conformidad de eficiencia	La capacidad que tiene el software para cumplir con los estándares o convenciones relacionados a la eficiencia.
Capacidad de Mantenimiento	Capacidad de ser analizado	La forma como el software permite diagnósticos de deficiencias o causas de fallas, o la identificación de partes modificadas.
	Cambiabilidad	La capacidad del software para que la implementación de una modificación se pueda realizar, incluye también codificación, diseño y documentación de cambios.
	Estabilidad	La forma como el software evita efectos inesperados para modificaciones del mismo.
	Facilidad de prueba	La forma como el software permite realizar pruebas a las modificaciones sin poner el riesgo los datos.
	Conformidad de facilidad de mantenimiento	La capacidad que tiene el software para cumplir con los estándares de facilidad de mantenimiento.
	Portabilidad	La capacidad que tiene el software para ser trasladado de un entorno a otro
Portabilidad	Adaptabilidad	Es como el software se adapta a diferentes entornos especificados (hardware o sistemas operativos) sin que implique reacciones negativas ante el cambio. Incluye la escalabilidad de capacidad interna
	Facilidad de instalación	La facilidad del software para ser instalado en un entorno específico o por el usuario final.
	Coexistencia	La capacidad que tiene el software para coexistir con otro o varios software, la forma de compartir recursos comunes con otro software o dispositivo.
	Reemplazabilidad	La capacidad que tiene el software para ser remplazado por otro software del mismo tipo, y para el mismo objetivo
	Conformidad de portabilidad	La capacidad que tiene el software para cumplir con los estándares relacionados a la portabilidad.

Fuente: Borbón, N. (2013).

Para el caso de medición en uso maneja cuatro características: efectividad, productividad, satisfacción, seguridad. (Ver Figura #2)

Figura #2. Características Modelo ISO/IEC 9126-1:2000 medición de calidad de uso
Fuente: Moreno, Bolaños, & Navia, (2010).

Borbón (2013) las define como se muestra a continuación:

> **Eficacia**: La capacidad del software para permitir a los usuarios finales realizar los procesos con exactitud e integridad.
> **Productividad**: La forma como el software permite a los usuarios emplear cantidades apropiadas de recursos, en relación a la eficacia lograda en un contexto específico de uso. Para una empresa es muy importante que el software no afecte al productividad del empleado
> **Seguridad**: Se refiere al que el Software no tenga niveles de riesgo para causar daño a las personas, instituciones, software, propiedad intelectual o entorno. Los riesgos son normalmente el resultado de deficiencias en la funcionalidad (Incluyendo seguridad), fiabilidad, usabilidad o facilidad de mantenimiento.
> **Satisfacción**: La satisfacción es la respuesta del usuario a la interacción con el software, e incluye las actitudes hacia el uso del mismo. A continuación se describe un cuadro donde podemos resumir las características y cada uno de sus atributos, este cuadro le ayudara a visualizar el proceso de evaluación.

2.2.3.2. McCall

Este modelo de la evaluación de la calidad es uno de los más antiguos. Fue publicado por McCall, Richards y Walters en 1977, y se enfoca en el producto final.

Moreno, Bolaños, & Navia (2010) indica lo siguiente:

> El modelo refleja perspectivas del desarrollador y del usuario, además presenta una estructura jerárquica para organizar los factores divididos en tres aspectos de calidad de software (revisión, transición y operación).

> Los factores de calidad planteados por McCall se miden a través de 21 criterios o métricas de calidad que él propone; el problema es que dichos criterios se calculan a través de preguntas dicotómicas del tipo “SI”-”NO”, las cuales son contestadas por una o varias personas, lo cual podría implicar subjetividad dado que cada una puede evaluar la calidad de forma diferente. (p. 43)

Los factores son descritos por Pressman (2010) de la siguiente forma:

> **Corrección**. Grado en el que un programa satisface sus especificaciones y en el que cumple con los objetivos de la misión del cliente.
> **Confiabilidad**. Grado en el que se espera que un programa cumpla con su función y con la precisión requerida [debe notarse que se han propuesto otras definiciones más completas de la confiabilidad].
> **Eficiencia**. Cantidad de recursos de cómputo y de código requeridos por un programa para llevar a cabo su función.
> **Integridad**. Grado en el que es posible controlar el acceso de personas no autorizadas al software o a los datos.
> **Usabilidad**. Esfuerzo que se requiere para aprender, operar, preparar las entradas e interpretar las salidas de un programa.
> **Facilidad de recibir mantenimiento**. Esfuerzo requerido para detectar y corregir un error en un programa (ésta es una definición muy limitada).
> **Flexibilidad**. Esfuerzo necesario para modificar un programa que ya opera.
> Susceptibilidad de someterse a pruebas. Esfuerzo que se requiere para probar un programa a fin de garantizar que realiza la función que se pretende.
> **Portabilidad**. Esfuerzo que se necesita para transferir el programa de un ambiente de sistema de hardware o software a otro.
> **Reusabilidad**. Grado en el que un programa (o partes de uno) pueden volverse a utilizar en otras aplicaciones (se relaciona con el empaque y el alcance de las funciones que lleva a cabo el programa).
> **Interoperabilidad**. Esfuerzo requerido para acoplar un sistema con otro. (p.342)

2.2.3.3. Dromey

Este modelo surge en el año 1996, después de McCall, es el segundo modelo más antiguo de la evaluación de la calidad del software. En él propone una técnica que no consiste en características de medición sino en propiedades que generen fiabilidad y mantenibilidad.

Moreno, Bolaños & Navia (2010), definen este modelo de la siguiente forma:

> El modelo de Dromey presenta un esquema de 6 relaciones binarias entre 3 entidades definidas (Conjunto de componentes, propiedades que acarrean calidad de los componentes, atributos de calidad de alto nivel), cuatro de las cuales permiten evaluar la calidad desde la perspectiva del producto o el proceso (aquellas de la punta de flecha sombreada).
> Éste modelo se propone como alternativa al inconveniente que se presenta cuando los atributos de alto nivel no pueden ser medidos directamente sobre el software. En respuesta a esto, los atributos de alto nivel se pueden obtener con la construcción de componentes que representen un conjunto de propiedades del producto, señalando aquellas que afectan los atributos de calidad, como: Funcionalidad, Fiabilidad, Usabilidad, Eficiencia, Mantenibilidad, Portabilidad y Reusabilidad. (p.44).

2.3 Marco Referencial

La medicina prepagada en Venezuela se rige por la Superintendencia de la Actividad Aseguradora (SUDEASEG), cuya misión, visión, objetivos y atribuciones se describen a continuación:

Misión

La SUDEASEG (2015) describe su misión de la siguiente forma:

> Ejercer la potestad regulatoria de la actividad aseguradora nacional, para garantizar la transformación socio-económica que promueve el Estado Venezolano, a través de la participación ciudadana y el compromiso de sus trabajadores y trabajadoras en la defensa de los derechos de los usuarios y usuarias y en el desarrollo sustentable y sostenible del sector asegurador.

Visión

La SUDEASEG (2015) reseña su visión de la siguiente forma:

> Ser un órgano de gestión integral comprometido con el desarrollo social y económico del país, alcanzando el reconocimiento nacional e internacional mediante el alto nivel de profesionalidad de su recurso humano, en consonancia con las políticas del Estado Venezolano, contribuyendo con el ejercicio eficiente en la producción de resultados, a la tutela del interés general representado por los usuarios y usuarias del sector asegurador y a la optimización del sistema público nacional de salud.

Objetivos

La SUDEASEG (2015), en su página web oficial, lista los siguientes objetivos:

- Velar por los intereses de los tomadores, asegurados y beneficiarios de los contratos de seguros, de reaseguros, los contratantes de los servicios de medicina prepagada y de los asociados de las cooperativas, creando condiciones que faciliten el incremento del bienestar social y la estabilidad del sector asegurador.
- Fortalecer las actividades de supervisión, control, vigilancia, autorización, regulación y fiscalización de la actividad aseguradora en el país.
- Adecuar las innovaciones tecnológicas en el accionar de la Superintendencia de la Actividad Aseguradora.
-Implementar normas e instrumentos que optimicen las funciones de regulación, supervisión, fiscalización y control de los sujetos regulados.
- Identificar oportunamente las desviaciones del sector asegurador.
- Velar por el cumplimiento de las normas y procedimientos de la organización.
- Velar por el cumplimiento de las estrategias tendentes al fortalecimiento del recurso humano
- Velar porque las Empresas de Seguros y de Medicina Prepagada realicen los aportes al fondo del sistema público nacional de salud.
- Integrar a las comunidades organizadas en el desarrollo del sector asegurador

Atribuciones

La SUDEASEG (2015), en su página web oficial, enumera las siguientes atribuciones:

1. Ejercer la potestad regulatoria para el control, vigilancia previa, concomitante y posterior, supervisión, autorización, inspección, verificación y fiscalización de la actividad aseguradora, en los términos establecidos en la presente Ley y su Reglamento.
2. Garantizar a las personas el libre acceso a los productos, bienes y servicios objeto de la presente Ley y proteger los derechos e intereses de los tomadores, asegurados, beneficiarios o contratantes respecto de los sujetos regulados.
3. Establecer el sistema de control, vigilancia, supervisión, regulación, inspección y fiscalización de la actividad aseguradora, bajo los criterios de supervisión preventiva e integral y adoptar las medidas necesarias para el cumplimiento de esta Ley, su Reglamento y normas prudenciales.
4. Intervenir y liquidar administrativamente a los sujetos regulados en los términos establecidos en la presente Ley y su Reglamento.

5. Promover la participación ciudadana en defensa de los derechos de los contratantes, asociados, tomadores, asegurados y beneficiarios.
6. Promover la participación ciudadana a través de los consejos comunales u otras formas de organización social.
7. Llevar a cabo procedimientos de conciliación como mecanismo alternativo de solución de conflictos.
8. Efectuar anualmente, en el curso del primer semestre de cada año, las publicaciones que estime necesarias a fin de dar a conocer la situación de la actividad aseguradora y de los sujetos regulados, especialmente en lo relativo a primas, siniestros, reservas técnicas, margen de solvencia, patrimonio propio no comprometido, condiciones patrimoniales y el número de sanciones impuestas a los sujetos regulados, así como de las personas que se haya determinado que han realizado operaciones reguladas por la presente Ley sin estar autorizadas para ello.
9. Establecer vínculos de coordinación y cooperación con otros entes u órganos de la Administración Pública Nacional, así como con autoridades de supervisión de otros países, a los fines de fortalecer los mecanismos de control, actualizar las regulaciones preventivas e intercambiar informaciones; a tal efecto se coordinará con el Ministerio del Poder Popular con competencia en materia de Relaciones exteriores.

Las demás que le atribuyan la presente Ley, otras Leyes y reglamentos.

Estructura organizacional

La estructura organizacional se presenta a través del siguiente organigrama, obtenido de su página web oficial:

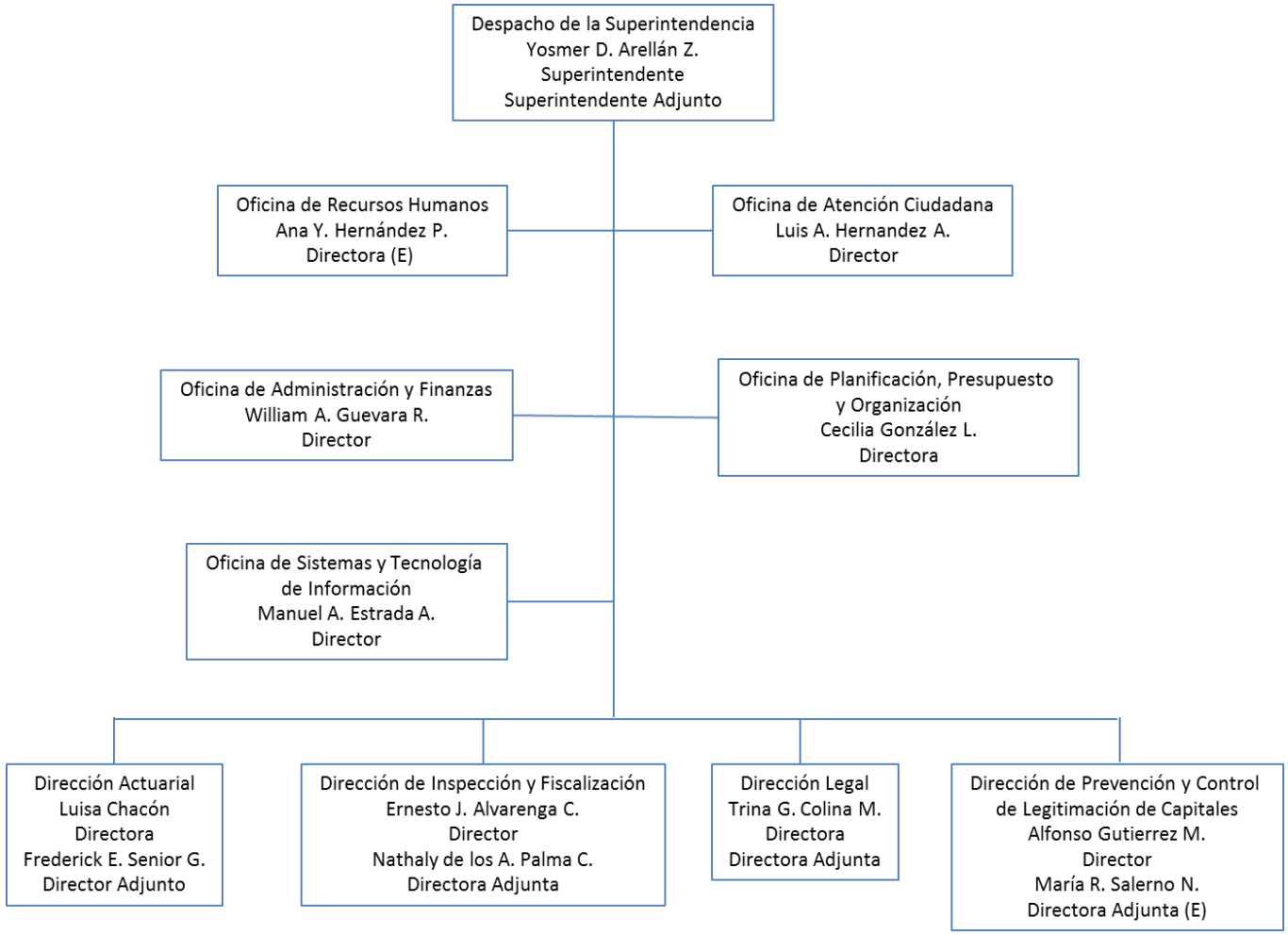

Figura #3. Estructura Organizativa SUDEASEG
Fuente: SUDEASEG (2015).

Los sujetos regulados por la SUDEASEG, se muestran en la figura #4, donde se destaca la medicina prepagada, la cual representa sector industrial de la investigación.

La medicina prepagada, como sujeto regulado por la SUDEASEG dispone siete denominaciones o empresas, tal y como lo describe la siguiente tabla:

Tabla #2. Medicina Prepagada

Nº de Registro	**Denominación**	**Providencia**	**Gaceta Oficial**	**Fecha Gaceta Oficial**
1	Rescarven Medicina Prepagada S.A	FSAA-2-2-002769	39.759	16-sep-11
2	Novosalud Medicina Prepagada,S.A	FSAA-2-5-000749	40.140	04-abr-13
3	Plansanitas S.A. Empresa de Medicina Prepagada	FSAA-2-5-003076	40.212	22-jul-13
4	Planisalud de Medicina Prepagada C.A.	FSAA-2-5-003511	40.261	30-sep-13
5	Sanitas Venezuela S.A., Empresa de Medicina Prepagada	FSAA-2-5-000999	40.450	09-jul-14
6	Salud Vital Medicina Prepagada,S.A	FSAA-2-5-001950	40.497	15-sep-14
7	Cuidasalud Medicina Prepagada,C.A	FSAA-2-5-001952	40.497	15-sep-14

Fuente: sudeaseg (2015).

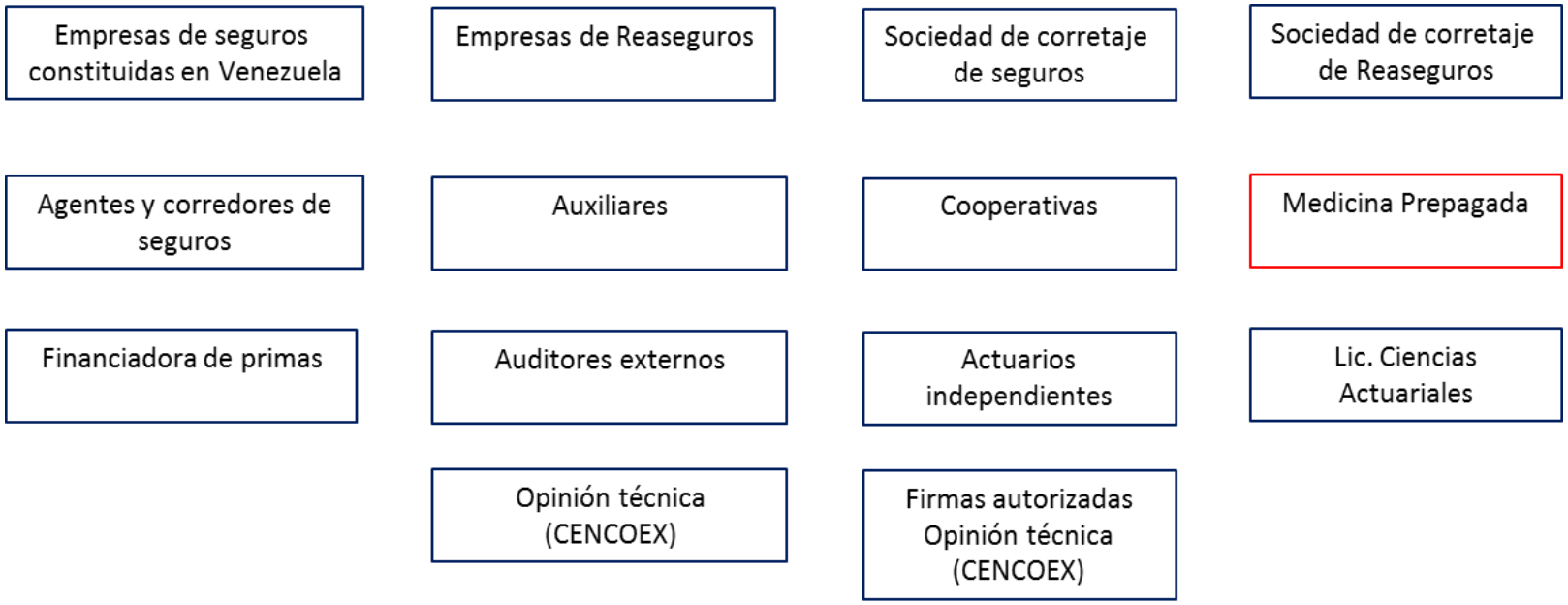

Figura #4. Sujetos regulados por la SUDEASEG
Fuente: SUDEASEG (2015).

2.4. Bases Legales

Esta sección está constituida por aquellos documentos de naturaleza legal que dan soporte a la investigación. Este trabajo de grado de maestría se encuentra bajo el siguiente marco legal:

1.- Marco regulatorio del Fondo Nacional de Ciencia, Tecnología e Innovación (Fonacit), entre las cuales se encuentran Ley Orgánica de Ciencia, Tecnología e Innovación, cuyo objetivo se describe en artículo 1 de la siguiente forma:

> El presente Decreto-Ley tiene por objeto desarrollar los principios orientadores que en materia de ciencia, tecnología e innovación, establece la Constitución de la República Bolivariana de Venezuela, organizar el Sistema Nacional de Ciencia, Tecnología e Innovación, definir los lineamientos que orientarán las políticas y estrategias para la actividad científica, tecnológica y de innovación, con la implantación de mecanismos institucionales y operativos para la promoción, estímulo y fomento de la investigación científica, la apropiación social del conocimiento y la transferencia e innovación tecnológica, a fin de fomentar la capacidad para la generación, uso y circulación del conocimiento y de impulsar el desarrollo nacional.

2.- Marco regulatorio de las empresas de la actividad aseguradora en Venezuela, el cual viene dado por la Ley de la Actividad Aseguradora y las Normas sobre Prevención, Control y Fiscalización de los Delitos de Legitimación de Capitales y el Financiamiento al Terrorismo, en la Actividad Aseguradora.

3.- Ley Orgánica del Sistema Venezolano para la Calidad (2002): Legislación venezolana basada en la Ley de Metrología y la Ley sobre Normas Técnicas y Control de Calidad, cuyo objetivo se plantea en su artículo 1 de la siguiente forma:

> Esta Ley tiene por objeto desarrollar los principios orientadores que en materia de calidad consagra la Constitución de la República Bolivariana de Venezuela, determinar sus bases políticas y diseñar el marco legal que regule el Sistema Venezolano para la Calidad, asimismo establecer los mecanismos necesarios que permitan garantizar los derechos de las personas a disponer de bienes y servicios de calidad en el País, a través de los subsistemas de

> Normalización, Metrología, Acreditación, Certificación, Reglamentaciones Técnicas y Ensayos.(p.2)

4.- Constitución de la República Bolivariana de Venezuela (1999): En el artículo 117 se establece lo siguiente acerca de la calidad de los productos y servicios:

> Todas las personas tendrán derecho a disponer de bienes y servicios de calidad, así como a una información adecuada y no engañosa sobre el contenido y características de los productos y servicios que consumen; a la libertad de elección y a un trato equitativo y digno. La ley establecerá los mecanismos necesarios para garantizar esos derechos, las normas de control de calidad y cantidad de bienes y servicios, los procedimientos de defensa del público consumidor, el resarcimiento de los daños ocasionados y las sanciones correspondientes por la violación de estos derechos.

CAPITULO III: MARCO METODOLÓGICO

El marco metodológico de la investigación define las herramientas y estructura utilizadas para dar respuesta al problema planteado en el primer capítulo. Está constituido por el tipo y el diseño de la investigación, la población y muestra, técnicas de recolección de datos, fases de la investigación, las variables (definición conceptual y operacional e indicadores), estructura desagregada de trabajo, aspectos éticos y finalmente el cronograma.

3.1 Tipo de Investigación

El tipo de investigación determina el objetivo de la investigación, y de esta forma define las herramientas, características y propiedades que ésta deberá adoptar. De acuerdo con el problema planteado en el primer capítulo, esta investigación está enmarcada por el nivel descriptivo, el cual consiste en detallar los hechos, sucesos o fenómenos tal y como son observados en el presente. ”Los estudios descriptivos buscan especificar las propiedades, las características y los perfiles de personas, grupos, comunidades, procesos, objetos o cualquier otro fenómeno que se someta a un análisis.” (Hernández, Fernández, & Baptista, 2010, pág. 80).

A través de este tipo de investigación, se buscará recolectar información acerca de la variable dependiente del proyecto (calidad del producto de software), modelos existentes y aplicados con éxito en una comunidad de desarrolladores de aplicaciones o sistemas, sin pretender indicar su relación con otros conceptos. El objetivo de este levantamiento de información es conocer los avances que se manejan en materia de calidad de producto de software, y de esta forma considerar los aspectos aprobados por los profesionales del área, mejorar aquellos que presentan fallas o inconvenientes, e innovar en los elementos que no se han considerado.

Finalmente el valor de las investigaciones descriptivas, las define Hernández, Fernández, & Baptista (2010), de la siguiente forma:

> …los estudios descriptivos son útiles para mostrar con precisión los ángulos o dimensiones de un fenómeno, suceso, comunidad, contexto o situación.
> En esta clase de estudios el investigador debe ser capaz de definir, o al menos visualizar, qué se medirá (qué conceptos, variables, componentes, etc.) y sobre qué o quiénes se recolectarán los datos (personas, grupos, comunidades, objetos, animales, hechos, etc.).(p.80).

Adicionalmente, de acuerdo al objeto de la investigación se considera que la investigación es de tipo proyectiva, la cual consiste en la presentación de una propuesta de un modelo como respuesta a una necesidad o carencia de un sector específico. Hurtado (2010), hace indica lo siguiente:

> La investigación proyectiva tiene como objetivo diseñar o crear propuestas dirigidas a resolver determinadas situaciones. Los proyectos de arquitectura e ingeniería, el diseño de maquinarias, la creación de programas de intervención social, el diseño de programas de estudio, los inventos, la elaboración de programas informáticos,., entre.. otros,.. siempre que estén sustentados en un proceso de investigación, son ejemplos de investigación proyectiva. Este tipo de investigación potencia el desarrollo tecnológico. (p. 133)

3.2 Diseño de la Investigación

El diseño de la investigación determina cómo el investigador logrará los objetivos planteados. “Plan o estrategia que se desarrolla para obtener la información que se requiere en una investigación” (Hernández, Fernández, & Baptista, 2010, pág. 120).

La importancia de una elección correcta del diseño, es definida por de la siguiente forma:

> Si el diseño está concebido cuidadosamente, el producto final de un estudio (sus resultados) tendrá mayores posibilidades de éxito para generar conocimiento. Puesto que no es lo mismo seleccionar un tipo de diseño que otro: cada uno tiene sus características propias. (p.120).

De acuerdo a estas definiciones, esta investigación corresponde al diseño no experimental, los cuales son “estudios que se realizan sin la manipulación deliberada de variables y en los que sólo se observan los fenómenos en su

ambiente natural para después analizarlos.” (Hernández, Fernández, & Baptista, 2010, pág. 149)

En este trabajo de investigación se definen diversas variables independientes (detalladas en los próximos apartados), las cuales no se pueden manipular ni controlar, son utilizadas para observar y describir hechos de su contexto en un tiempo determinado.

3.3 Población y Muestra

La población es un “conjunto de todos los casos que concuerdan con determinadas especificaciones.” (Hernández, Fernández, & Baptista, 2010, pág. 174).

La población está representada por las empresas de medicina prepagada inscritas en la SUDEASEG hasta el año 2015 en Venezuela, que dispongan una gerencia de sistemas o tecnología, encargada de la compra o desarrollo de software.

Una muestra es un “Subgrupo de la población del cual se recolectan los datos y debe ser representativo de ésta.” (Hernández, Fernández, & Baptista, 2010, pág. 173).

Las muestras se categorizan de la siguiente forma:

a) Muestra probabilística: Todos los elementos que constituyen la población tienen la misma probabilidad de conformar la muestra, a través de una selección aleatoria.

b) Muestra no probabilística: Los elementos son escogidos de acuerdo a las características de la investigación. Acerca de esta categoría, Hernández, Fernández, y Baptista indican lo siguiente:

> El procedimiento no es mecánico ni con base en fórmulas de probabilidad, sino que depende del proceso de toma de decisiones de un investigador o de un grupo de investigadores y, desde luego, las muestras seleccionadas obedecen a otros criterios de investigación. Elegir entre una muestra probabilística o una no probabilística depende de los objetivos del estudio, del esquema de investigación y de la contribución que se piensa hacer con ella. (p.176)

El tipo de muestra seleccionada en este proyecto de investigación es no probabilística, la cual viene dada por dos empresas, de las siete que se encuentras registradas actualmente en la SUDEASEG, ellas son: Sanitas Venezuela S.A (Nro. de registro 5) y Plansanitas S.A. (Nro. de registro 3), estas se utilizarán para validar el modelo de calidad de software a proponer, y son elegidas a criterio del investigador por disponer experiencia trabajando para estas organizaciones y disponer de acceso a la información requerida para la investigación, formando parte de un muestreo intencional u opinático (“selección de los elementos con base en criterios o juicios del investigador” (Arias, 1999, pág. 24)).

3.4 Técnicas e instrumentos de recolección de datos

La recolección de los datos es una etapa de gran importancia para el proyecto de investigación, ya que permite demostrar el problema planteado y comprobar y analizar la solución propuesta.

“Las técnicas de recolección de datos son las distintas formas o maneras de obtener la información” (Arias, 1999, pág. 25).

El recurso necesario para adquirir estos datos es denominado instrumento de investigación, “son los medios materiales que se emplean para recoger y almacenar la información” (Arias, 1999, pág. 25).

La siguiente tabla resume las diferentes técnicas y los instrumentos a implementar en este trabajo de investigación:

Tabla #3. Técnicas de Recolección de Datos

Técnica	Descripción	Instrumento
Encuesta	"Tal vez sea el instrumento más utilizado para recolectar los datos, consiste en un conjunto de preguntas respecto de una o más variables a medir." (Hernández, Fernández, & Baptista, 2010, pág. 217)	Cuestionario
Observación	"Consiste en el registro sistemático, válido y confiable de comportamientos y situaciones observables, a través de un conjunto de categorías y subcategorías." (Hernández, Fernández, & Baptista, 2010, pág. 260)	Guía de observación

Fuente: Elaborado por el autor.

3.5 Fases de la Investigación

De acuerdo al problema expuesto en este trabajo de investigación y los objetivos definidos, se llevará a cabo este proyecto a través de las siguientes fases:

1. Validar la existencia de modelos de calidad de producto de software para el sector de medicina prepagada.

   Se utilizará como herramienta el cuestionario de preguntas cerradas y la guía de observación.
2. Diseñar el modelo de evaluación de calidad de producto de software propuesto, basado en normativas internacionales como ISO/IEC 9126-1:2000.
3. Validar el modelo de evaluación diseñado en el apartado anterior, con la finalidad de comprobar que éste se adapta al sector para el cual fue creado y de esta manera aprobar su implementación.

Se utilizará como herramienta la lista de Cotejo.

3.6 Procedimiento por Objetivos

Los procedimientos que se llevarán a cabo para cumplir con los objetivos específicos planteados en el capítulo I de este trabajo de investigación son los siguientes:

1) Describir los modelos existentes en el mercado para medición o evaluación de la calidad de software:

Levantamiento de información acerca de los modelos exitosos de evaluación de calidad de software existentes en el mercado, los cuales constituyen los antecedentes y bases teóricas de esta investigación.

2) Caracterizar los elementos del modelo de la calidad de software para medicina prepagada:

Investigación acerca de aquellos modelos diseñados especialmente para el sector de medicina prepagada, con el propósito de determinar cuáles componentes pueden ser un gran aporte para el modelo a proponer.

3) Diseñar las características, subcaracterísticas y métricas del modelo de evaluación de la calidad del producto de software adaptado a las necesidades del mercado:

Elaborar el esquema y diseño del modelo de evaluación de calidad de producto de software (incluyendo características y subcaracterísticas de medición), basado en estándares internacionales como ISO/IEC 9126-1:2000 y considerando los requerimientos del mercado.

4) Validar el modelo de evaluación diseñado:

Llevar a cabo una prueba piloto con la cual se apruebe que el modelo de evaluación diseñado cumpla con las necesidades descritas en la recolección de datos, y adicionalmente se adapte al sector al cual está dirigido.

3.7 Variables, Definición Conceptual y Operacional e Indicadores

Una variable en el ámbito de la investigación se puede definir de la siguiente forma: “cualidad susceptible de sufrir cambios. Un sistema de variables consiste, por lo tanto, en una serie de características por estudiar, definidas de manera operacional, es decir, en función de sus indicadores o unidades de medida.” (Arias, 1999, pág. 17)

Ahora bien, existen dos tipos de definición de variables, conceptual y operacional.

“Una definición conceptual trata a la variable con otros términos” (Hernández, Fernández, & Baptista, 2010, pág. 110).

Una definición operacional “especifica qué actividades u operaciones deben realizarse para medir una variable”.

Finalmente, los indicadores representan la forma de medir y precisar resultados una vez aplicados los procesos de la definición operacional.

En la tabla #4 se observa la definición de variables de acuerdo a este trabajo de investigación.

Tabla #4. Operacionalización de las variables

Objetivo General	Diseñar un modelo de evaluación de la calidad del producto de software para medicina prepagada			
Objetivos Específicos	**Variable**	**Definición conceptual**	**Definición operacional**	**Indicadores**
Describir los modelos existentes en el mercado para la evaluación de la calidad de software	Modelos de evaluación de la calidad de producto de software.	Proceso eficaz de software que se aplica de manera que crea un producto útil que proporciona valor medible a quienes lo producen y a quienes lo utilizan. (Pressman, 2010, pág. 340)	Técnicas de recolección de información referente modelos más aplicados en el mercado para la evaluación de la calidad del producto de software.	a) Documentación referente a modelos de calidad de software. b) Documentación referente a las características, subcaracterísticas y métricas empleadas por los modelos de calidad de software existentes.
Caracterizar los elementos del modelo de la calidad de software para medicina prepagada.	Modelos de evaluación de la calidad de producto de software en el sector de medicina prepagada	Conjunto de propiedades mediante las cuales se evalúa y describe su calidad. (Bertoa, Troya, & Vallecillo, pág. 6)	Técnicas de recolección de datos aplicadas en el sector de medicina prepagada, con el objetivo de conocer sus procesos de evaluación de la calidad de sus software.	a) Resultados proceso de observación que contiene los registros de la información recopilada referente a los modelos de calidad de software implementados en dos empresas de medicina prepagada en Venezuela. b) Información recolectada referente a los modelos de calidad de software implementados en las empresas de medicina prepagada en Venezuela.
Diseñar las etapas del modelo de evaluación de la calidad del producto de software adaptado a las necesidades	- Arquitectura - Control - Depuración - Formalidad - Reusabilidad - Integridad - Eficacia - Comprensibilidad - Eficiencia - Mantenibilidad	Propiedad de calidad a la que puede asignársele una métrica, entendiendo por métrica un procedimiento que examina un componente y	Definición de las características, subcaracterísticas y métricas que constituyen el modelo a proponer, las cuales permitirán evaluar la calidad de los productos	a) Características, subcaracterísticas y métricas del modelo de calidad a proponer. b) Adaptabilidad del modelo al sector para el cual fue diseñado.

Objetivo General	Diseñar un modelo de evaluación de la calidad del producto de software para medicina prepagada			
Objetivos Específicos	**Variable**	**Definición conceptual**	**Definición operacional**	**Indicadores**
del mercado.	- Portabilidad	produce un dato simple, un símbolo (p.e. Excelente, Sí, No) o un número. (Bertoa, Troya, & Vallecillo, pág. 6)	de software en las empresas del sector de medicina prepagada.	
Validar el modelo de evaluación diseñado.	Modelo de evaluación de calidad de producto de software propuesto.	Evaluación de conformidad del cumplimiento de los requisitos especificados al realizar el levantamiento de información y diseño del modelo de calidad de producto de software.	Conjunto de acciones que se llevarán a cabo para validar el modelo a proponer en este trabajo de investigación, entre las cuales se tiene la prueba piloto.	Documento de certificación de calidad del sistema evaluado en la prueba piloto.

Fuente: Elaborado por el autor.

3.8 Estructura Desagregada de Trabajo

La Estructura Desagregada de Trabajo (EDT), “permite establecer los trabajos necesarios del proyecto, de forma que nos permita definir claramente y de forma individual los distintos componentes y entregables que formarán parte del proyecto.” (González, 2014).

En la figura #5 se descompone de manera jerárquica los entregables que se deben generar para lograr los objetivos planteados en este trabajo de investigación.

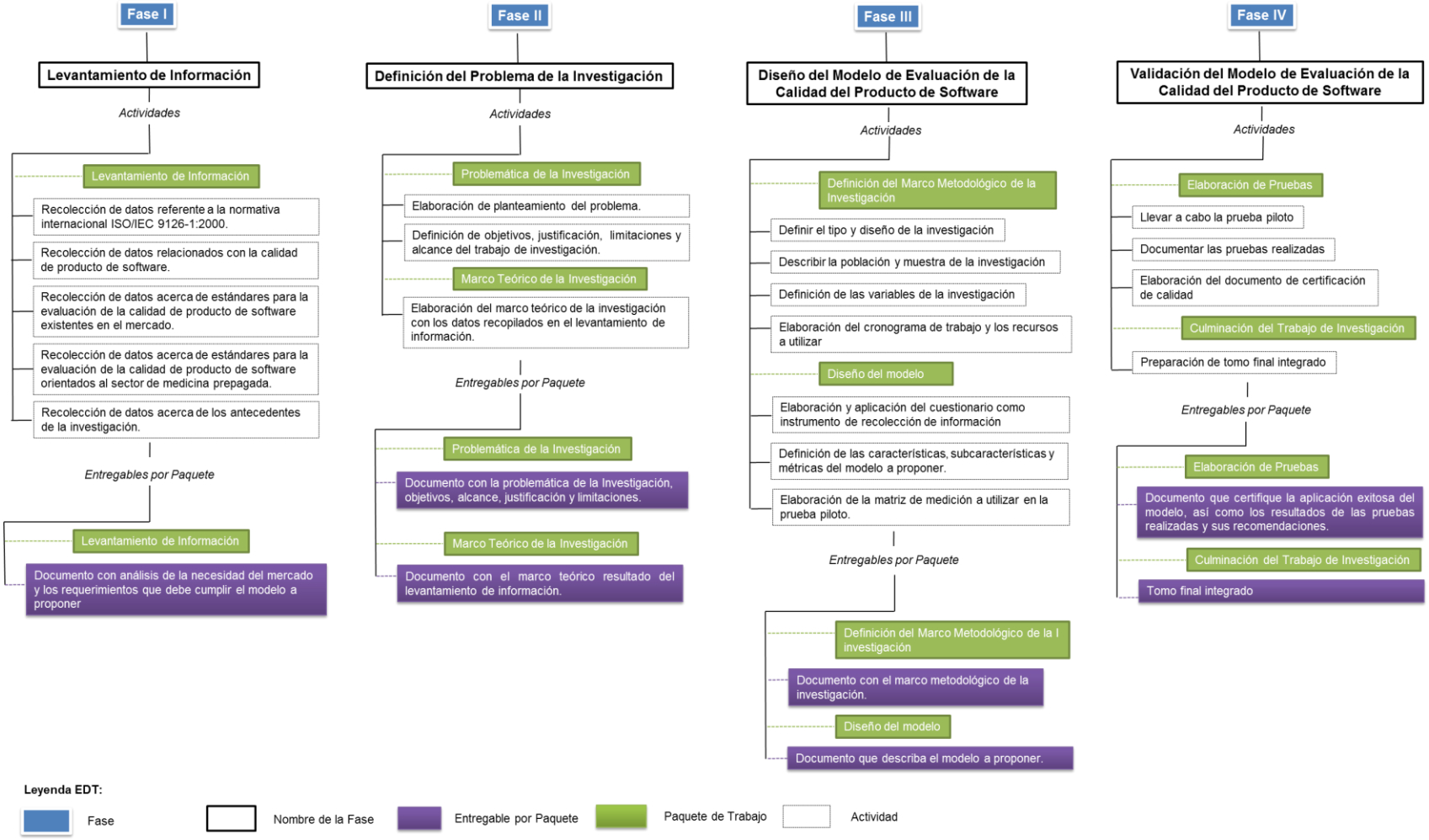

Figura #5.Estructura Desagregada de Trabajo
Fuente: Elaborado por el autor.

3.9 Aspectos Éticos

La Real Academia Española define la ética como “Parte de la filosofía que trata del bien y del fundamento de sus valores.”

Dentro de la investigación científica, la ética busca el cumplimiento de las leyes que rijan la investigación, velar por el correcto uso de los datos e información confidencial que en ella se suministre, evitar el plagio de la investigación, entre otros aspectos. La rama encarga de estas tareas se denomina ética de la investigación científica, y Miranda (2013) la define de la siguiente forma:

> Rama de la ética especial, que versa sobre el modo como los principios comunes de la moralidad de los actos humanos se especifican en un ámbito determinado de la vida del hombre, en este caso, el de la investigación científica y el trabajo académico

Este trabajo de investigación se rige bajo las regulaciones estipuladas por el Fondo Nacional de Ciencia, Tecnología e Innovación (FONACIT) y la Superintendencia de la Actividad Aseguradora (SUDEASEG), por lo que se respetarán los aspectos éticos y legales dispuestos por ellos.

De igual forma se acata, dentro del marco ético, las leyes contra el plagio, respecto de la investigación y el derecho de autor, haciendo referencia o identificando cada aporte con su ejecutor o creador.

3.10 Cronograma

El cronograma de las actividades que se llevó a cabo durante este trabajo de investigación se puede detallar en las figuras 6 y 7, en las cuales se presenta un diagrama de Gantt que expone la duración de cada una de las tareas definidas, es decir, es una herramienta que permitió planificar el tiempo invertido en cada actividad a ejecutar para la culminación del royecto.

Cada actividad esta identifica por un id único, por el cual se podrá ubicar en el diagrama de Gantt.

3.11 Recursos

A continuación se muestra una matriz con los recursos utilizados para llevar a cabo esta investigación.

Tabla #5. Recursos necesarios para llevar a cabo la investigación

Material	Tipo	Cantidad	Costo (Bsf.)	Horas Hombre	Costo total (Bsf.)
Recursos de oficina	Material	No Aplica	20.000	No Aplica	20.000
Impresora	Material	1	40.000	No Aplica	40.000
Hojas blancas	Material	1 (Resma)	2.500	No Aplica	2.500
Base de Datos a la cual accederá la aplicación	Base de Datos	1	0	No Aplica	0
Servidor web	Software	1	0	No Aplica	0
Aplicaciones de entorno para el desarrollo de aplicaciones	Software	1	0	No Aplica	0
Aplicaciones de tipo oficina, para la elaboración de encuesta, reportes, entre otros.	Software	5	4.000	No Aplica	4.000
Seminario de Investigación I.	Capacitación	1	5.052	200	5.052
Seminario de Investigación II.	Capacitación	1	6.062	200	6.062
Tutor	Capacitación	1	0	20	0
Inscripción del Trabajo de Grado de Maestría	Capacitación	1	9 U.C. x 3.500	No Aplica	31.500
				Total (Bsf.)	109.114

Fuente: Elaborado por el autor.

ID	Nombre	Fecha de ini..	Fecha de fin	Dura...	Antecesores
4	Fase I: Levantamiento de Información	01/10/15	21/10/15	15	
17	Recolección de datos referente a la normativa internacional ISO/IEC 9126-1:2000.	01/10/15	05/10/15	3	
18	Recolección de datos relacionados con la calidad de producto de software	06/10/15	08/10/15	3	
20	Recolección de datos de estándares para la evaluación de la calidad de producto de software para el sector de medicina prepagat	14/10/15	16/10/15	3	
19	Recolección de datos acerca de estándares para la evaluación de la calidad de producto de software existentes en el mercad	09/10/15	13/10/15	3	
21	Recolección de datos acerca de los antecedentes de la investigación	19/10/15	21/10/15	3	
30	Fase II: Definición del Problema de la Investigación	22/10/15	18/11/15	20	
35	Elaboración de planteamiento del problema	22/10/15	26/10/15	3	17,18,19,20
36	Definición de objetivos, justificación, limitaciones y alcance del trabajo de investigació	27/10/15	02/11/15	5	
38	Elaboración del marco teórico de la investigación con los datos recopilados en el levantamiento de informacić	02/11/15	18/11/15	13	21
39	Fase III: Diseño del Modelo de Evaluación de la Calidad del Producto de Softwa	19/11/15	27/05/16	137	
41	Definir el tipo y diseño de la investigación	19/11/15	23/11/15	3	
44	Describir la población y muestra de la investigació	24/11/15	26/11/15	3	
45	Definición de las variables de la investigación	27/11/15	04/12/15	6	
46	Elaboración del cronograma de trabajo y los recursos a utiliz	07/01/16	12/01/16	4	
47	Elaboración y aplicación del cuestionario como instrumento de recolección de informacić	11/04/16	22/04/16	10	45
48	Definición de las características, subcaracterísticas y métricas del modelo a propone	25/04/16	20/05/16	20	47
49	Elaboración de la matriz de medición a utilizar en la prueba pilot	23/05/16	27/05/16	5	
69	Fase IV: Validación del Modelo de Evaluación de la Calidad del Producto de Softwa	30/05/16	01/07/16	25	
70	Llevar a cabo la prueba pilot	30/05/16	03/06/16	5	49
71	Documentar las pruebas realizadas	06/06/16	10/06/16	5	70
72	Elaboración del documento de certificación de calida	13/06/16	17/06/16	5	71
76	Preparación de tomo final integrado	20/06/16	01/07/16	10	

Figura #6.Cronograma del Proyecto Parte I
Fuente: Elaborado por el autor.

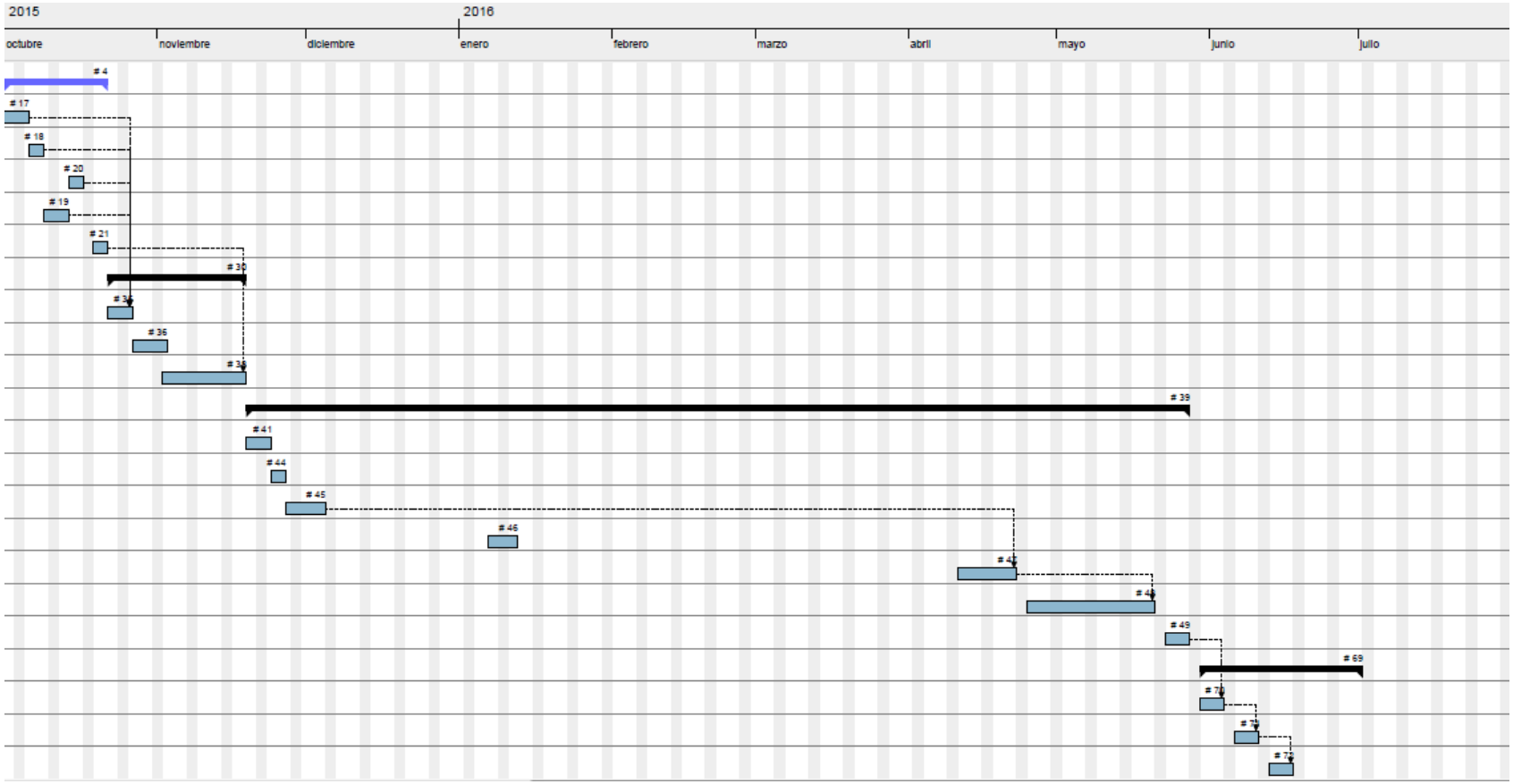

Figura #7.Cronograma del Proyecto Parte II
Fuente: Elaborado por el autor.

CAPÍTULO IV: ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

El presente capítulo consiste en el análisis de los datos obtenidos de la aplicación de los instrumentos de recolección, con el objetivo de comprender y contextualizar la situación en la cual se realiza la investigación.

A continuación se describen los instrumentos de recolección de datos utilizados, su proceso de validación así como los resultados obtenidos de su aplicación.

Desarrollo de los Instrumentos de Recolección de Datos

Con el objetivo de recaudar información relacionada con los elementos de calidad evaluados en los software de la medicina prepagada, se diseñaron dos instrumentos de recolección de datos, los cuales fueron aplicados en las organizaciones indicadas como muestra en el en el marco metodológico de la presente investigación.

1.- Encuesta

Aplicación de un cuestionario de 14 preguntas a una muestra de individuos. Con la encuesta se busca conocer las tendencias de calidad de las empresas de medicina prepagada Sanitas y PlanSanitas, evaluar si aplican algún modelo o estándar, y como llevan a cabo este proceso.

El diseño aplicado se puede observar en Anexo A y B.

2.- Guía de Observación

La guía de observación consta de 19 preguntas divididas en secciones que corresponden a las características evaluadas en la norma ISO/IEC 9126. Con este instrumento se busca determinar cuáles y en qué medida son aplicadas de las empresas de medicina prepagada Sanitas y PlanSanitas, indicando con el nivel 5 que siempre se toma en cuenta o se lleva a cabo el comportamiento de cada ítem y 0 el valor para expresar que nunca se aplica.

El diseño aplicado se puede observar en Anexo C y D.

Validación de los Instrumentos de Recolección de Datos

Con el objetivo de garantizar un correcto proceso de recolección de datos, los instrumentos fueron validados por tres expertos. Dicha colaboración se gestionó a través del documento de solicitud de evaluación, el cual se puede apreciar en el Anexo E.

Adicionalmente se suministró un formato a completar con la apreciación cualitativa y cualquier observación que se desee acotar, la misma se encuentra en el Anexo F.

Finalmente, con la finalidad de dejar evidencia de dicha validación, los evaluadores firmaron la constancia de evaluación del instrumento (Ver Anexo G, Anexo H. Anexo I).

Resultados de la Encuesta y Análisis de su Aplicación

1.- Posee la organización algún área encargada de evaluar y controlar la calidad de los productos de software.

Si	No
20%	80%

Figura #8.Resultado pregunta número 1 de la encuesta.

2.- La empresa dispone algún auditor externo encargado de evaluar y controlar la calidad de los productos de software.

Si	No
0	5

Figura #9.Resultado pregunta número 2 de la encuesta.

3.- El personal del área de tecnología de la empresa se encuentra capacitado en materia de calidad de software.

Si	No
1	4

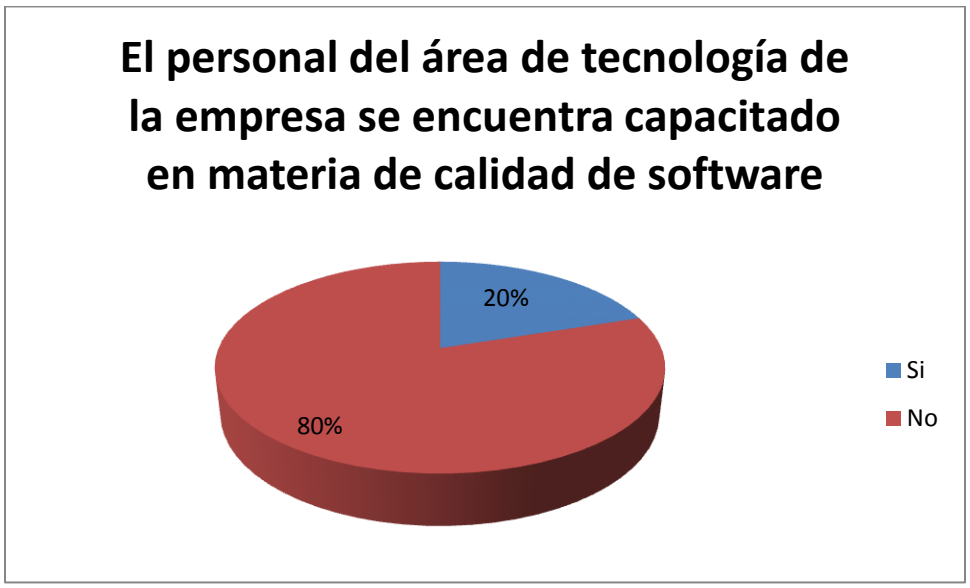

Figura #10.Resultado pregunta número 3 de la encuesta.

4.- Se maneja la evaluación de la calidad de los productos de software bajo algún modelo determinado.

Si	No
0	5

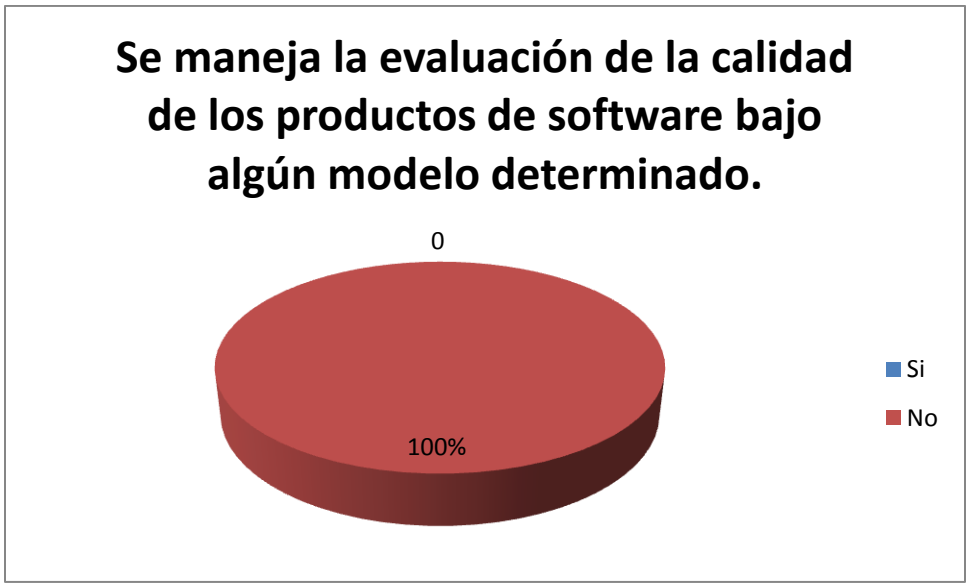

Figura #11.Resultado pregunta número 4 de la encuesta.

5.- Considera usted que la evaluación de la calidad de producto de software agrega/agregaría valor dentro de la organización.

Si	No
5	0

Figura #12.Resultado pregunta número 5 de la encuesta.

6.- El área de tecnología involucra al grupo de usuarios durante el desarrollo de los sistemas de información.

Si	No
2	3

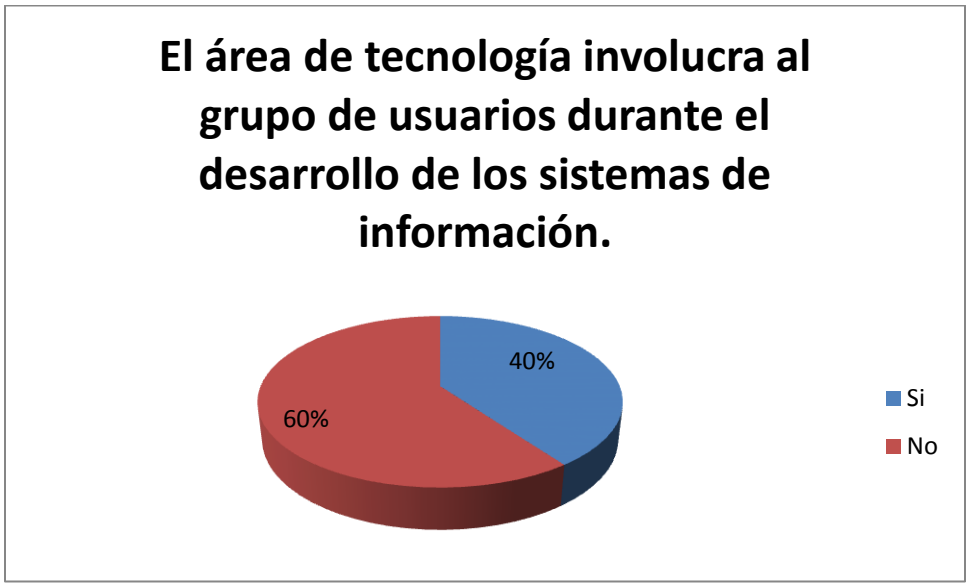

Figura #13.Resultado pregunta número 6 de la encuesta.

7.- Durante el desarrollo de software, se realiza seguimiento a los programadores, con el objetivo de validar el apego de las funcionalidades con la lista de requerimientos.

Si	No
1	5

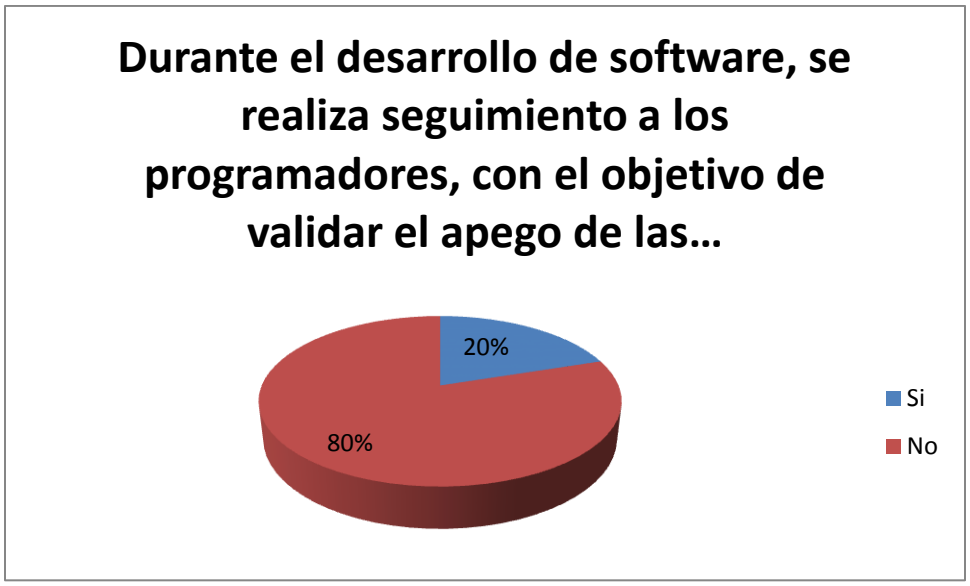

Figura #14.Resultado pregunta número 7 de la encuesta.

8.- Durante la etapa de diseño del software se toman en cuenta aspectos como: eficiente uso de los recursos, coexistencia, seguridad.

Si	No
1	4

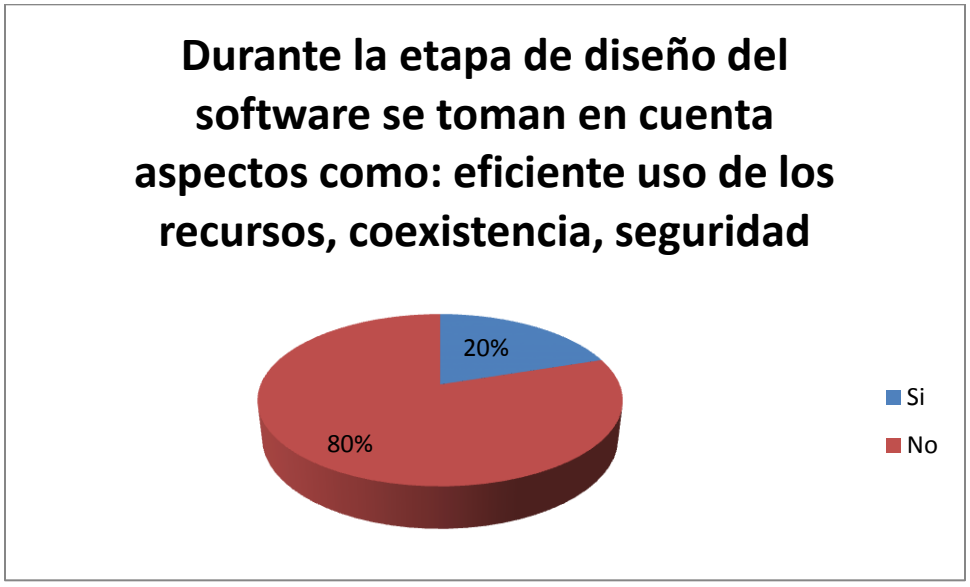

Figura #15.Resultado pregunta número 8 de la encuesta.

9.- Una vez culminado el producto de software, se realiza el levantamiento de la documentación técnica que facilite el mantenimiento de la aplicación.

Si	No
1	4

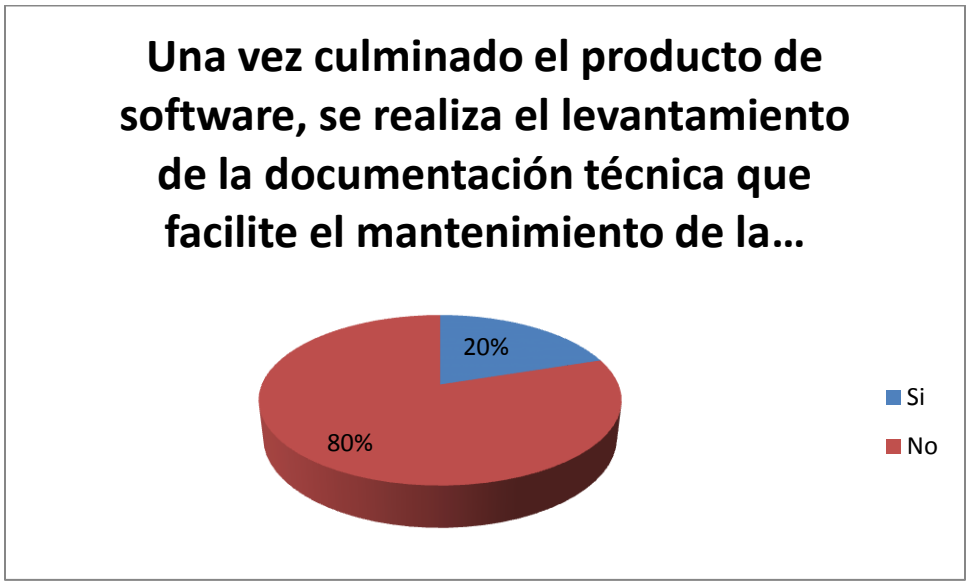

Figura #16.Resultado pregunta número 9 de la encuesta.

10.- Para cada aplicación disponible en la empresa se maneja un manual para el usuario, que permita aclarar sus dudas o hacer más fácil su capacitación.

Si	No
0	5

Figura #17.Resultado pregunta número 10 de la encuesta.

11.- La empresa maneja algún proceso de pruebas general para cada aplicación desarrollada.

Si	No
0	5

Figura #18.Resultado pregunta número 11 de la encuesta.

12.- Existe un área dedicada a realizar el proceso de pruebas de cada aplicación que se desarrolla.

Si	No
40%	60%

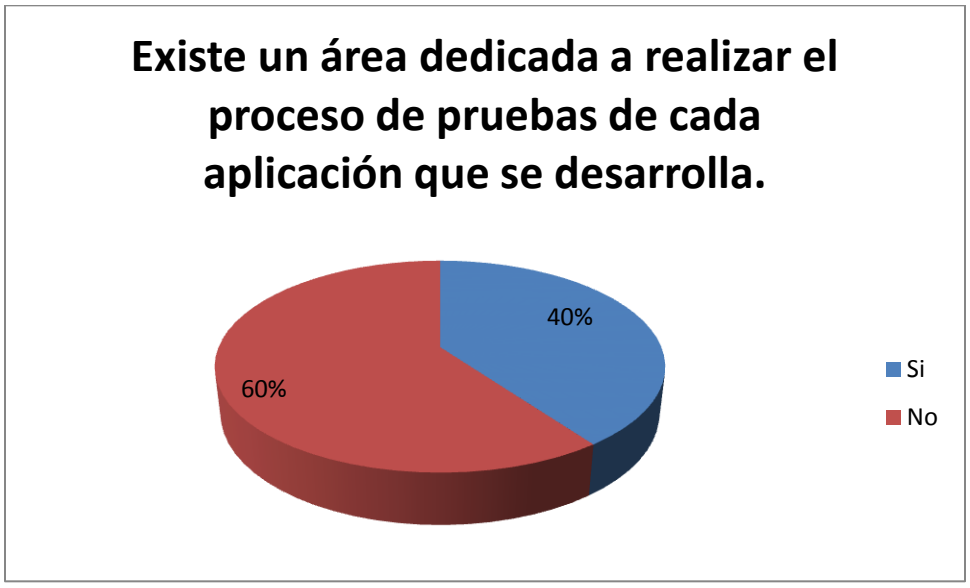

Figura #19.Resultado pregunta número 12 de la encuesta.

13.- La empresa contrata los servicios de proveedores para el desarrollo de aplicaciones.

Si	No
5	0

Figura #20.Resultado pregunta número 13 de la encuesta.

14.- De ser afirmativa la respuesta anterior, aplica algún modelo de evaluación de la calidad a los productos que desarrollan los proveedores.

Si	No
0	5

Figura # 21.Resultado pregunta número 14 de la encuesta.

De acuerdo a las gráficas presentadas, las cuales representan el porcentaje de cumplimiento de cada ítem de acuerdo a la opinión de los trabajadores de la empresa, se concluye lo siguiente:

a) La empresa no dispone de un área encargada de la evaluación y control de la calidad de los productos de software, esta actividad se lleva a cabo por el área de desarrollo de aplicaciones, los cuales deben asegurar el apego de las funcionalidades con la lista de requerimientos.

b) No se dispone de auditores externos encargados del proceso de evaluación de la calidad, por lo que la empresa posee control total de ello. Sin embargo se han realizado consultorías de buenas prácticas de seguridad en las aplicaciones

c) Se considera que el personal posee un mínimo conocimiento en materia de calidad de software.

d) El área de tecnología de la organización no aplica ningún modelo de evaluación de la calidad de productos de software.

e) Existe un alto interés en la evaluación de la calidad de los productos de software, ya que se considera que ello agrega un gran valor dentro de la organización.

f) Los usuarios solo son involucrados en el desarrollo de las aplicaciones con mesas de trabajo al momento de levantamiento del requerimiento y en algunas pruebas unitarias, por lo que en ocasiones los sistemas no cumplen con todas las funciones necesarias o no se adaptan por completo a las reglas de negocio de la empresa.

g) El eficiente uso de los recursos, coexistencia y seguridad son manejados en baja medida por el área de arquitectura de software durante la etapa de diseño de los sistemas.

h) En pocas ocasiones se realiza el levantamiento de la documentación técnica que facilite el mantenimiento de la aplicación.

i) No se manejan manuales de usuarios, lo que dificulta el proceso de capacitación y crea dependencia del área de desarrollo.

j) No existe una estructura general para la elaboración de pruebas ni un área encargado de esta tarea, por lo que la depuración de las aplicaciones se ve en riesgo.

k) La empresa maneja contratación de proveedores para el desarrollo de aplicaciones, sin embargo no se aplica ningún modelo ni actividades de evaluación de la calidad de los productos adquiridos.

Resultados de la Guía de Observación y Análisis de su Aplicación

#	Comportamiento a Evaluar	5	4	3	2	1	0	Observaciones
Funcionalidad								
1	Los productos de software cumplen con las funciones y objetivos para los cuales fueron creados.		x					
2	En el diseño de las aplicaciones se evalúa la interacción con otros sistemas.				x			
3	Los accesos a los sistemas se realizan solo por personal autorizado.			x				
4	Los sistemas desarrollados cumple con las normas y estándares, u otras prescripciones de la ingeniería de software				x			

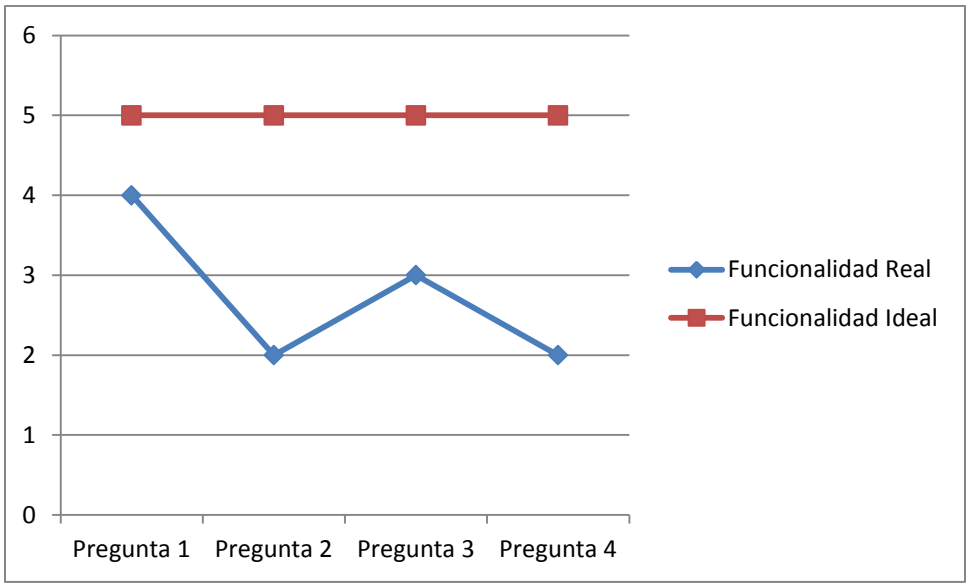

Figura #22.Resultado de la característica Funcionalidad de la guía de observación.

A pesar que las aplicaciones, en su mayoría cumplen con las funciones y objetivos para las cuales fueron creadas, se descuidan aspectos en materia de diseño y estándares de ingeniería de software. La característica funcionalidad se encuentra por debajo de los valores ideales.

#	Comportamiento a Evaluar	5	4	3	2	1	0	Observaciones
Fiabilidad								
5	Antes de colocar un sistema en ambiente de producción, se eliminan las fallas observadas en el proceso de pruebas.				x			
6	En los sistemas se contemplan tareas de recuperabilidad de datos en caso de fallas.					x		
7	Se manejan niveles de desempeño alto en las aplicaciones, considerando de esta forma que son tolerantes a fallos.					x		

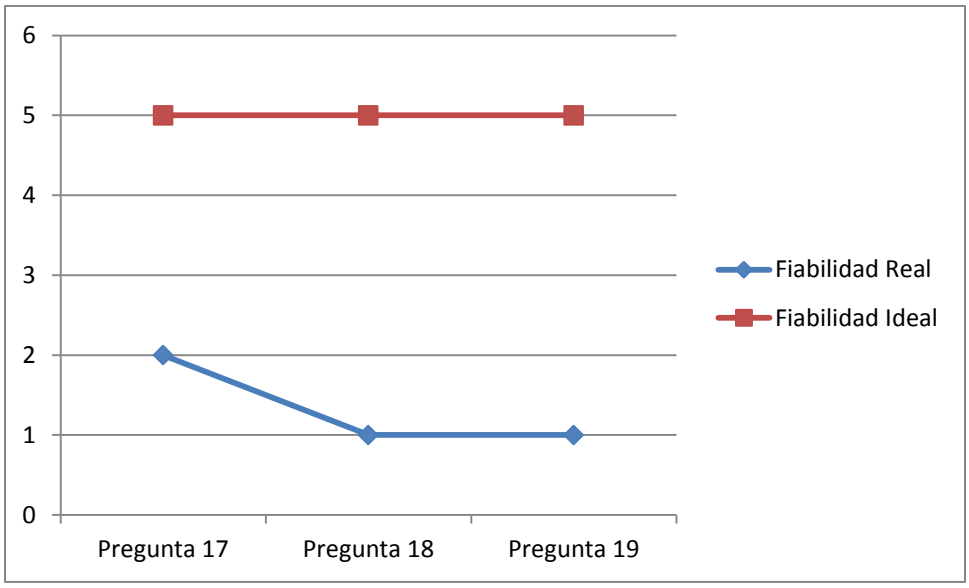

Figura #23.Resultado de la característica Fiabilidad de la guía de observación.

La característica de fiabilidad se ve muy afectada en la empresa. Sus valores se encuentran muy por debajo de aquellos que garantiza calidad del producto. La recuperabilidad y tolerancia a fallos son los puntos con menos énfasis.

#	Comportamiento a Evaluar	5	4	3	2	1	0	Observaciones
Usabilidad								
8	El diseño de los sistemas es atractivo y compresible para los usuarios.			x				
9	El esfuerzo de los usuarios en comprender el funcionamiento de las aplicaciones es bajo.			x				
10	El proceso de aprendizaje y capacitación de los usuarios es extenso.				x			

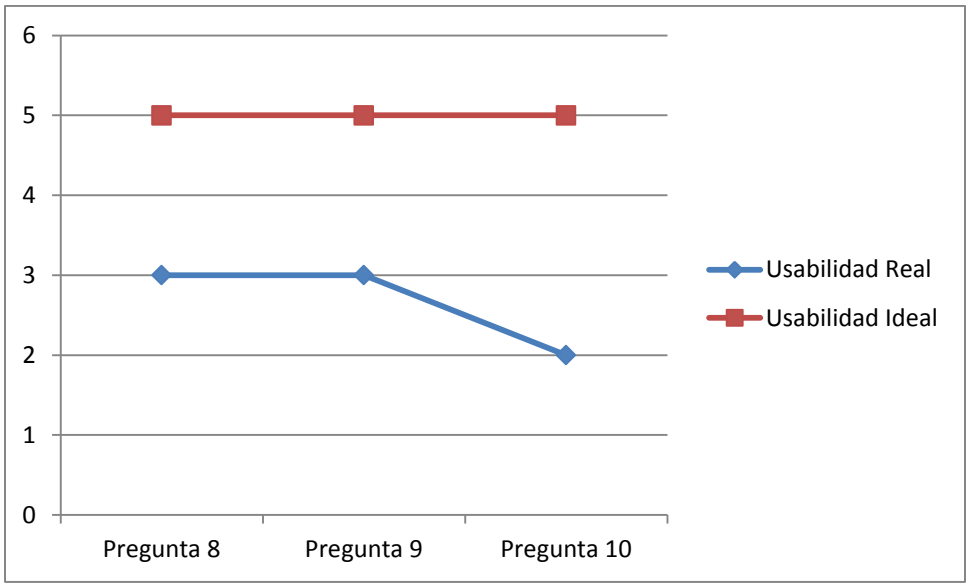

Figura #24.Resultado de la característica Usabilidad de la guía de observación.

Las aplicaciones son diseñadas y construidas con un nivel de facilidad de uso aceptable, por lo que el esfuerzo de los usuarios en comprender su funcionamiento no se ve afectado en gran medida, sin embargo, el proceso de capacitación no es manejado con la prioridad que este requiere. La característica Usabilidad se encuentra por debajo de los valores ideales.

#	Comportamiento a Evaluar	5	4	3	2	1	0	Observaciones
Eficiencia								
11	Durante la construcción del software, son evaluados sus tiempos de respuesta					x		
12	Los recursos a consumir por la aplicación son evaluados durante su diseño y construcción					x		

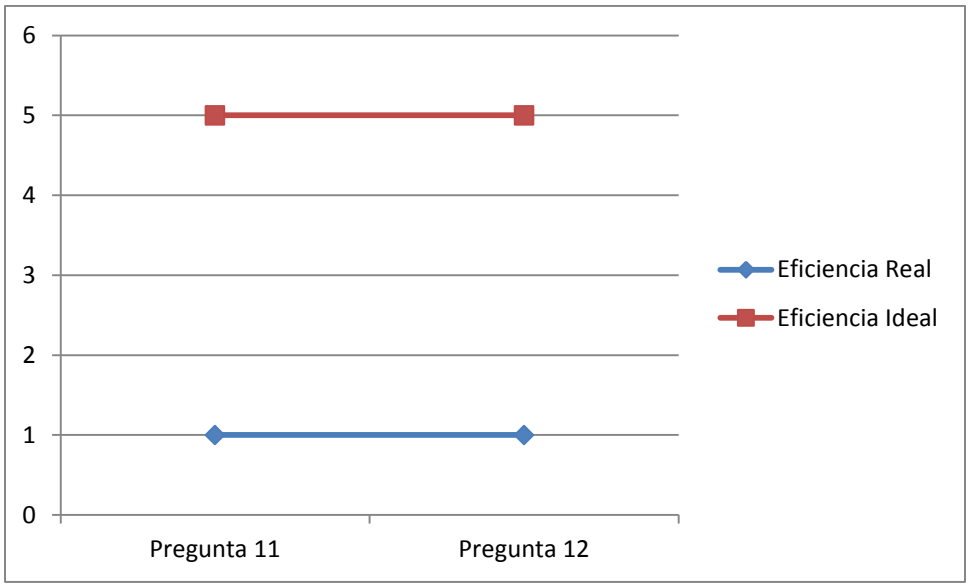

Figura #24.Resultado de la característica Eficiencia de la guía de observación.

La eficiencia se considera una de las características más descuidada en la organización. El correcto manejo de los recursos no se encuentra estipulado como una actividad de gran importancia dentro de la etapa de desarrollo.

#	Comportamiento a Evaluar	5	4	3	2	1	0	Observaciones
Mantenibilidad								
13	Se pueden aplicar cambios preventivos, correctivos y de ampliación a los sistemas sin afectar su funcionamiento.			x				
14	Los procesos de pruebas de las aplicaciones son sencillos y no requiere de esfuerzos adicionales.				x			
15	El esfuerzo requerido para los cambios referentes a corrección de fallas es bajo.				x			
16	En caso de fallas o inconvenientes con los sistemas, el proceso de diagnóstico es rápido y sencillo.				x			

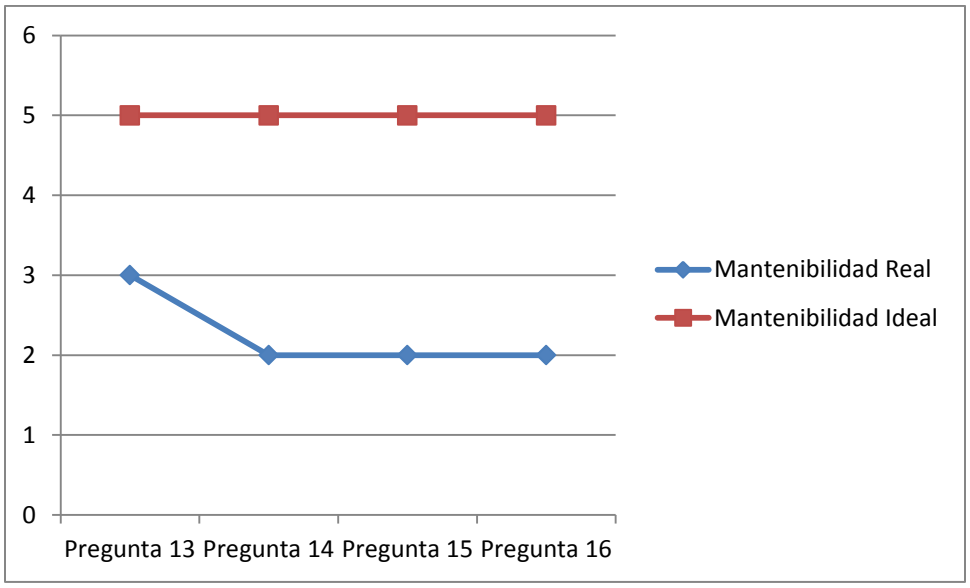

Figura #25.Resultado de la característica Mantenibilidad de la guía de observación.

Al no disponer estándares de calidad e ingeniería de software bien establecidos dentro de los procesos de diseño de los sistemas, la característica de Mantenibilidad se ubica por debajo de los valores ideales, ya que se requiere grandes esfuerzos para realizar alguna modificación, bien sea de alto o bajo impacto, así como el tiempo invertido en dar soluciones a inconvenientes o fallas.

#	Comportamiento a Evaluar	5	4	3	2	1	0	Observaciones
Portabilidad								
17	Se dispone de un plan de acción sencillo para la instalación o puesta en marcha del software.				x			
18	El funcionamiento de las aplicaciones no se ve afectado por el traslado a otros ambientes.					x		
19	Las aplicaciones pueden ser reemplazadas por otras que manejen los mismos propósitos o funciones.					x		

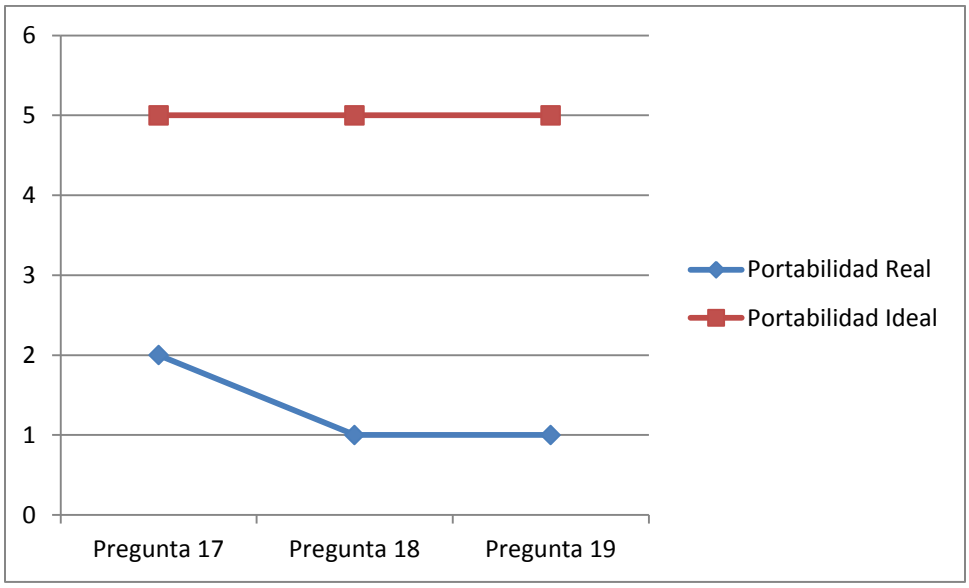

Figura #26.Resultado de la característica Portabilidad de la guía de observación.

La capacidad de migración o traslado de las aplicaciones no se encuentra dentro de las características de los software desarrollados, así como su facilidad de instalación a través de un proceso estructurado que evite inconvenientes con la plataforma o errores en su despliegue. La característica Portabilidad encuentra muy baja con respecto a los valores ideales.

CAPÍTULO V: DESARROLLO DE LA PROPUESTA

A continuación se describe el diseño del modelo de evaluación de la calidad del producto de software propuesto en la presente investigación, el cual consiste en 11 características, 33 subcaracterísticas y sus respectivas métricas. Con ello se busca dar solución a gran parte de las complicaciones presentes en las organizaciones al momento de evaluar la calidad de sus productos de software. (Ver figura #27)

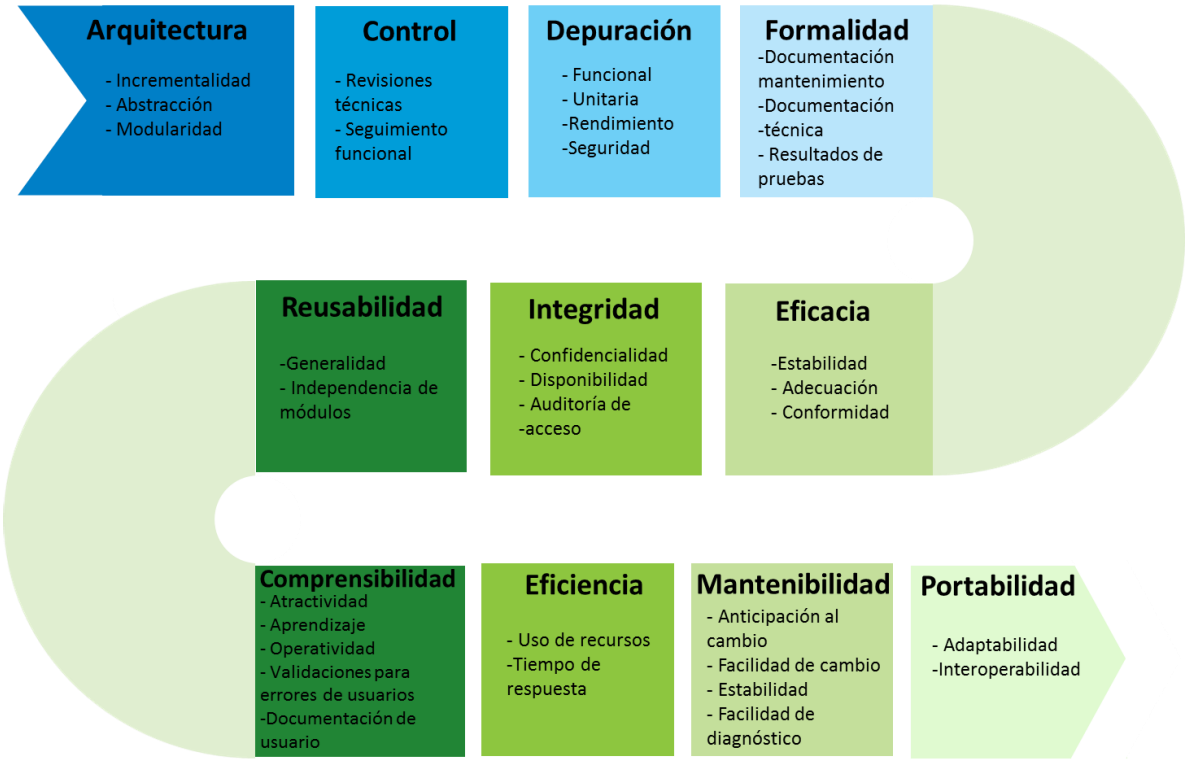

Figura #27.Descripción del modelo propuesto
Fuente: Elaboración del autor.

Características

El modelo propuesto se muestra en la figura #28, consiste en once características que permitirán evaluar la calidad del producto de software, cuatro de ellas deben verificarse durante su desarrollo y construcción, sin interferir con

las metodologías seleccionadas ni modelos para la evaluación del proceso, y las siete restante serán aplicadas al sistema construido.

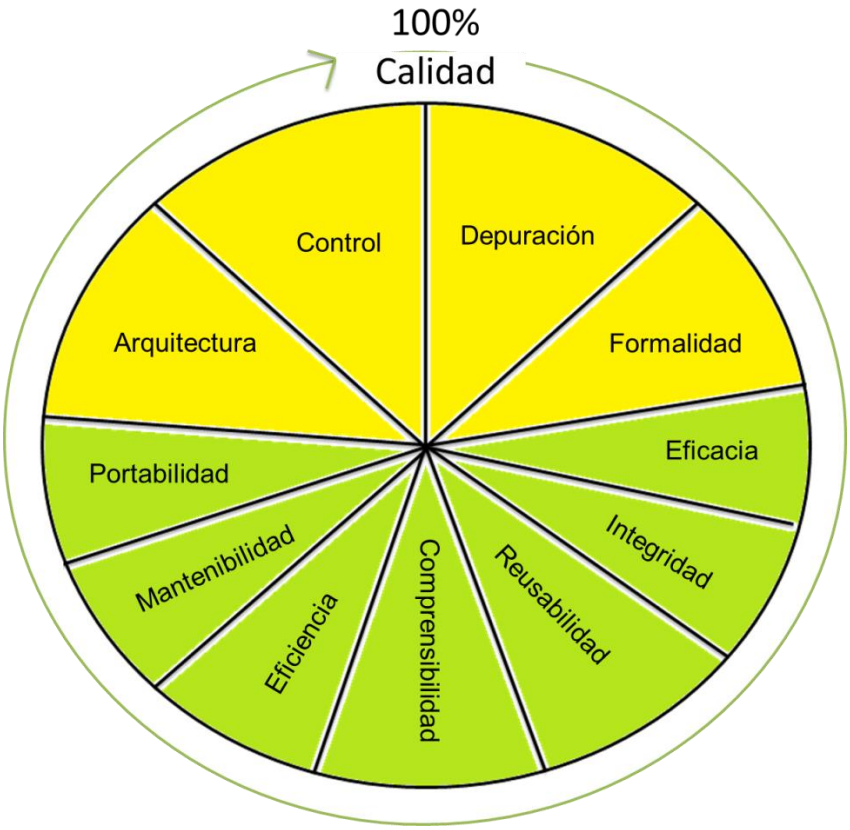

Figura #28.Características del modelo de evaluación de la calidad de producto de software propuesto.
Fuente: Elaboración del autor.

Las características a evaluar durante la creación del software no hacen referencia a ningún procedimiento a seguir durante esta fase, únicamente describen objetivos que se deben cumplir para garantizar que se logre la calidad del producto.

A continuación se delimita en que consiste cada una de ellas:

1) Arquitectura: Estructura bajo la cual se puede desarrollar el software para que cumpla con los requerimientos del cliente, además de proyectarse para usos futuros, interactuando con sistemas externos o sus componentes sirvan de apoyo para la construcción de otro software.

2) Control: Acciones de apoyo que ayudarán a alcanzar las metas de calidad, a través de la validación de cumplimiento de la arquitectura.

3) Depuración: Funciones de pruebas y auditorias para la disminución de errores en tiempo de ejecución e instalación del software.
4) Formalidad: Recursos de tipo documentación requeridos para el fácil manejo del proceso de mantenimiento y expansión.
5) Eficacia: Capacidad del software de cumplir con los requerimientos o necesidades para las cuales fue creado, es decir, cumplimiento funcional.
6) Integridad: Cualidad del software que hace referencia a la seguridad, es decir prevenir vulnerabilidades y riesgos
7) Reusabilidad: Atributo del software que permite utilizar más de una vez un código, fragmento de él o módulos con el mismo fin.
8) Comprensibilidad: Competencia encargada de medir la cantidad de esfuerzo implementado por los usuarios para comprender el funcionamiento del software y usarlo. Hace referencia a la curva de aprendizaje entre el usuario y la aplicación.
9) Eficiencia: Habilidad de cumplir los objetivos del software haciendo uso únicamente de los recursos necesarios para su éxito.
10) Mantenibilidad: Competencia encargada de medir la cantidad de esfuerzo requerido por los programadores para modificar de forma preventiva, correctiva o extensiva el software.
11) Portabilidad: Competencia encargada de medir la cantidad de esfuerzo necesario para lograr que un sistema pueda continuar su funcionamiento en otro ambiente al cual es transferido.

Subcaracterísticas

Cada característica que compone el modelo, a su vez posee subcaracterísticas, las cuales son un conjunto de atributos a evaluar por cada una de ellas. Para este modelo propuesto se tienen las siguientes:
1) Arquitectura

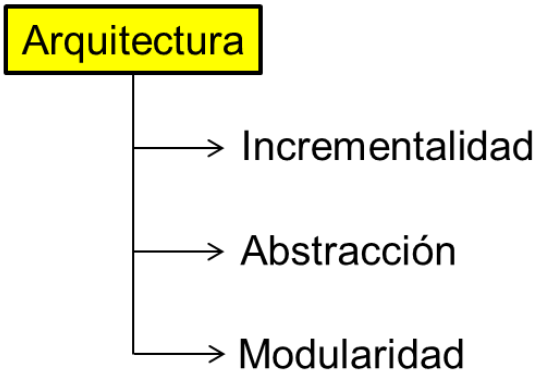

Figura #29.Subcaracterísticas de la Arquitectura.
Fuente: Elaboración del autor.

a. Incrementalidad: Proceso de desarrollo por pasos o incremento, en donde los avances se realizan una vez se haya cumplido una meta o alcance definido, es decir, llevar a cabo el proyecto por aproximaciones.

b. Abstracción: Capacidad de separar los detalles o comportamientos específicos que son requeridos en el diseño del software, de aquellos aspectos que son de menor importancia. En ella se evita almacenar información o incluir funcionalidades poco necesarias.

c. Modularidad: Separación de un sistema por módulos o fragmentos menos complejos. En esta subcaracterística se debe tomar en cuenta el factor de alta cohesión (los módulos deben cumplir con las funciones para las cuales fueron creadas y delegar aquellas que involucren a otras entidades) y bajo acoplamiento (poca dependencia entre los módulos).

2) Control

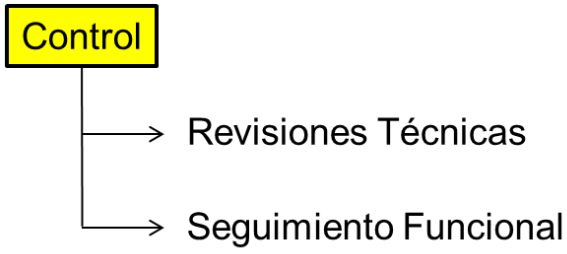

Figura #30.Subcaracterísticas del Control.
Fuente: Elaboración del autor.

a. Revisiones técnicas: Validación del cumplimiento técnico de los entregables, con el objetivo de detectar de forma temprana problemas o errores técnicos o no cumplimiento de estándares de calidad,

b. Seguimiento funcional: Evaluación del cumplimiento de las funciones y requerimiento del sistema así como control del cronograma del proyecto, evitando retrasos que puedan afectar la calidad del producto.

3) Depuración

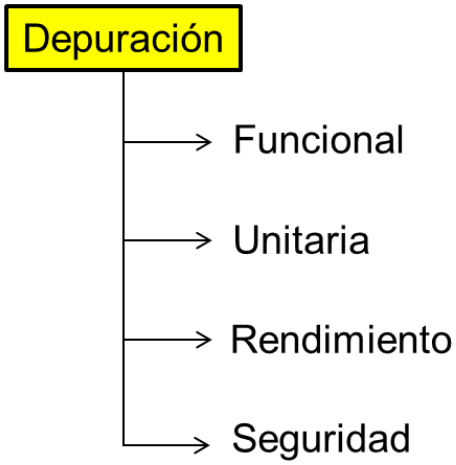

Figura #31.Subcaracterísticas de la Depuración.
Fuente: Elaboración del autor.

a. Funcional: Elaboración de pruebas que validen en tiempo de ejecución que las funciones que componen el software cumplen con las especificaciones y requisitos para las cuales fue creado.

b. Unitaria: Elaboración de pruebas que validen en tiempo de ejecución el correcto funcionamiento de los módulos que componen el sistema por separado.

c. Rendimiento: Elaboración de pruebas que validen en tiempo de ejecución los tiempos de respuesta en condiciones específicas de trabajo.

d. Seguridad: Elaboración de pruebas que indiquen las vulnerabilidades y riesgos a los que se encuentra expuesto el software, con el objetivo de tomar medidas de seguridad al respecto.

4) Formalidad

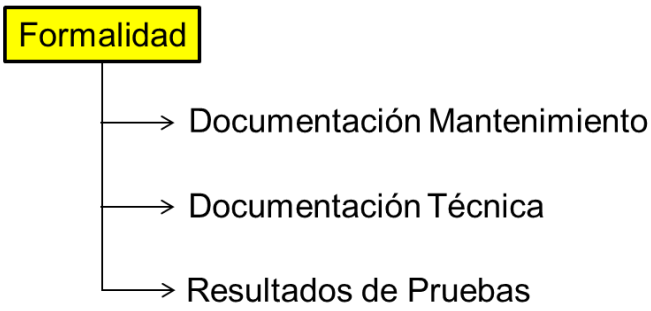

Figura #32.Subcaracterísticas de la Formalidad.
Fuente: Elaboración del autor.

a. Documentación mantenimiento: Documentación con el histórico de los cambios preventivos o correctivos del sistema y sus implicaciones e impacto en el diseño original, con el objetivo diferenciar el código original de cualquier modificación, de manera que el software pueda ser auditado con mejores resultados.

b. Documentación técnica: Documentos que contiene la descripción técnica y funcional del software, incluye aspectos como diagramas, requerimientos de hardware y software, interacciones con otros sistemas, rutinas o procesos complejos, entre otros.

c. Resultados de pruebas: Informe con los resultados obtenidos del conjunto de pruebas recomendadas para un software, indicando las acciones a tomar, bien sea solventar errores detectados o llevar a un ambiente de producción el sistema construido.

5) Eficacia

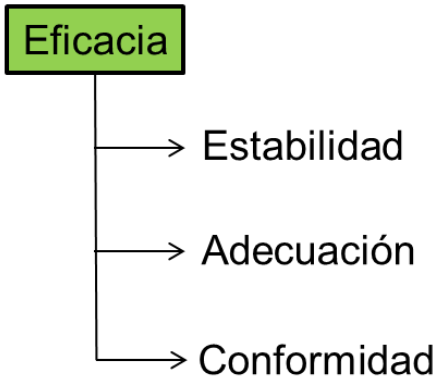

Figura #33.Subcaracterísticas de la Eficacia.
Fuente: Elaboración del autor.

a. Estabilidad: Atributo que hace referencia al nivel de fallos mínimo que debe tener el software para responder de manera óptima en un ambiente de producción.
b. Adecuación: Capacidad del software de cumplir con un conjunto de funcionalidades que satisface los requerimientos y objetivos especificados por el usuario.
d. Conformidad: Capacidad del software de cumplir con los acuerdos y resultados esperados y concretados al inicio de proyecto.

6) Integridad

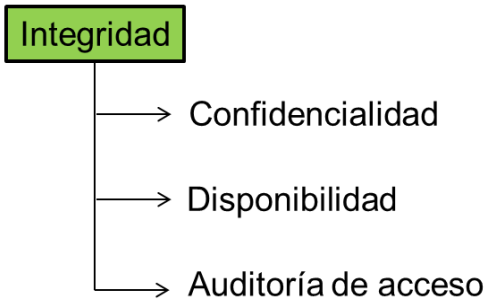

Figura #34.Subcaracterísticas de la Integridad.
Fuente: Elaboración del autor.

a. Confidencialidad: Atributo que hace referencia a la capacidad del software de brindar la información únicamente al personal autorizado para tal fin.
b. Disponibilidad: Atributo que hace referencia al tiempo durante el cual el sistema se mantiene respondiendo ante aquellas funciones para las cuales fue creado, es decir, el porcentaje del tiempo de servicio.
c. Auditoría de acceso: Capacidad del software de gestionar los accesos a la información del software únicamente para el personal autorizado, además de registrar cualquier modificación que este realice.

7) Reusabilidad

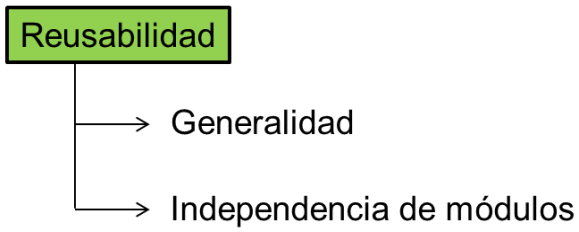

Figura #35.Subcaracterísticas de la Reusabilidad.
Fuente: Elaboración del autor.

a. Generalidad: Propiedad del software de aportar la solución que abarque la mayor cantidad de casos posibles, es decir, buscar la respuesta más general de un problema.

b. Independencia de módulos: Atributo que hace referencia al bajo acoplamiento que debe disponer un módulo para que pueda ser reusable bien sea dentro del mismo código fuente o en sistemas distintos.

8) Comprensibilidad

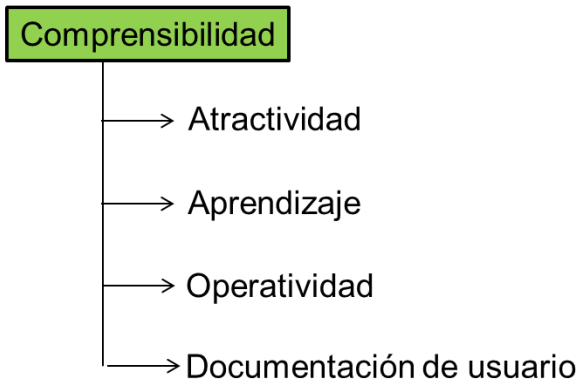

Figura #36.Subcaracterísticas de la Comprensibilidad.
Fuente: Elaboración del autor.

a. Atractividad: Capacidad del software de ser agradable ante el usuario. Hace referencia al componente vista del software.

b. Aprendizaje: Atributo que hace referencia al esfuerzo requerido por el usuario para aprender sobre el manejo y uso del sistema.

c. Operatividad: Capacidad del software de adaptarse al control por parte del usuario para responder a las tareas que este debe llevar a cabo.

d. Validaciones para errores de usuarios: Capacidad de proteger los datos que procesa el sistema y evitar errores en tiempo de ejecución causados por el manejo del usuario.

e. Documentación de usuario: Constituye aquellos manuales de usuario que facilitan el proceso de enseñanza y permita consultar ante cualquier duda de su funcionamiento sin depender de los desarrolladores que llevaron a cabo el proyecto.

9) Eficiencia

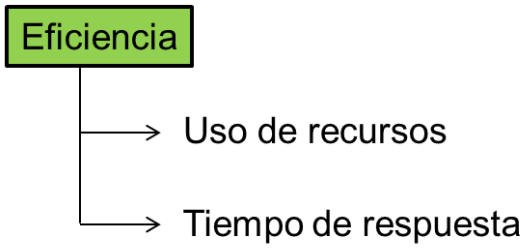

Figura #37.Subcaracterísticas de la Eficiencia.
Fuente: Elaboración del autor.

a. Uso de recursos: Capacidad del software de procesar ciertas peticiones en el lapso más corto posible consumiendo únicamente la cantidad de recursos necesarios. Incluyendo para este caso como un recurso el almacenamiento.

b. Tiempo de respuesta: Capacidad del software de procesar ciertas peticiones y otorgar una respuesta en las tasas de tiempo adecuadas.

10) Mantenibilidad

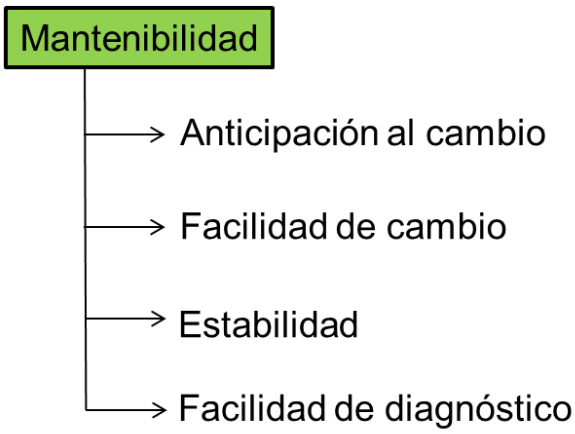

Figura #38.Subcaracterísticas de la Mantenibilidad.
Fuente: Elaboración del autor.

a. Anticipación al cambio: Capacidad de la arquitectura del software de anticipar cambios bien sea preventivos o correctivos, con el objetivo que estos se desarrollen en un futuro de la manera más fácil posible.

b. Facilidad de cambio: Atributo del software que hace referencia al esfuerzo requerido por el programador para llevar a cabo un cambio sin que este afecte el funcionamiento del sistema.

c. Estabilidad: Capacidad del software de mantener los niveles de fallos o errores mínimos ante cualquier modificación realizada, es decir, el porcentaje de riesgo de llevar a cabo un cambio debe ser muy bajo.

d. Facilidad de diagnóstico: Atributo del software que hace referencia al esfuerzo requerido por el programador para determinar las causas de un inconveniente o error presentado en el sistema.

11) Portabilidad

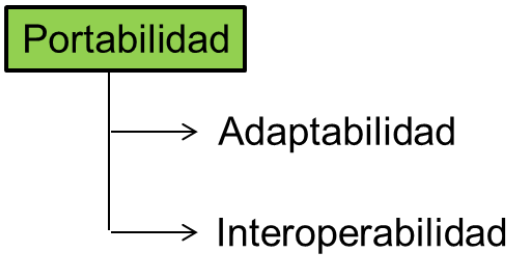

Figura #39.Subcaracterísticas de la Portabilidad.
Fuente: Elaboración del autor.

a. Adaptabilidad: Atributo que hace referencia a capacidad del software de funcionar de manera correcta o sin implicaciones negativas en un ambiente (hardware y software) distinto al cual se ha instalado inicialmente.
b. Interoperabilidad: Capacidad de un software de coexistir con otros sistemas e intercambiar recursos (como información), sin que esto afecte su funcionamiento

Métricas

Las métricas en el proceso de evaluación de la calidad de software es la medida en la que un producto cumple o posee un atributo del modelo.

Cada subcaracterística tendrá asignada una métrica o método de medición, a través del cual se indicará si el software posee o no cierto atributo.

La clasificación de las métricas del modelo propuesto son las siguientes:

Tabla #6. Tipos de métricas del modelo propuesto

Tipo de métrica	Descripción
Métricas de desarrollo	Buscan medir las actividades que se realizan durante la construcción del sistema, sin indicar un procedimiento específico para llevarlos a cabo o metodología de desarrollo.
Métricas de software	Tienen como objetivo la verificación del cumplimiento de los principios básicos de desarrollo exitoso del software.
Métricas de efectividad	Métricas asociadas a la medición del correcto uso de los recursos consumidos por el sistema para el procesamiento de datos.
Métrica de funcionalidad	Buscan medir la aproximación del producto creado con la necesidad manifestada por el cliente.
Métrica de complejidad	Hace referencia a la dificultad del código que constituye el sistema, así como de su uso por parte de los usuarios finales.

Tipo de métrica	Descripción
Métrica de cambio	Buscan medir la complejidad de realizar un cambio en el sistema, bien sea en el código o en el ambiente que se ejecuta.
Métrica Técnicas	Se centra en medir la calidad del cumplimiento de sus funciones en tiempo de ejecución.
Métrica de Seguridad	Encargadas de evaluar si existen medidas de seguridad en el sistema, con el objetivo de resguardar la información que en él se procesa.

Fuente: Elaboración del autor.

Para el modelo propuesto las métricas son las siguientes:

1) Característica: Arquitectura

1.1) Subcaracterística: Incrementalidad

1.1.1) Métrica:

Tabla #7. Métricas de la subcaracterística incrementalidad del modelo propuesto

Nombre	Desarrollo por pasos o incremento
Tipo de métrica	Métricas de desarrollo
Medición	¿Cumple el desarrollo del producto de software con el atributo de Incrementalidad?
Método de aplicación	Asegurar que el proceso de desarrollo se lleve a cabo por aproximaciones o avances.
Interpretación	Si -> Cumple con el atributo No -> No cumple con el atributo

Fuente: Elaboración del autor.

1.2) Subcaracterística: Abstracción

1.2.1) Métrica:

Tabla #8. Métricas de la subcaracterística abstracción del modelo propuesto

Nombre	Separar los detalles del diseño del software.
Tipo de métrica	Métricas de desarrollo.
Medición	¿Los objetos del producto de software solo poseen los atributos esenciales y asociados a sus funciones?

Método de aplicación	Asegurar que los objetos se diseñen únicamente con los atributos asociadas a sus funcionalidades.
Interpretación	Si -> Cumple con el atributo No -> No cumple con el atributo

Fuente: Elaboración del autor.

1.2) Subcaracterística: Modularidad

1.2.1) Métrica:

Tabla #9. Métricas de la subcaracterística modularidad del modelo propuesto

Nombre	Separación del sistema por módulos.
Tipo de métrica	Métricas de software.
Método de aplicación	Determinar la cantidad de módulos que constituyen el sistema, y adicionalmente cumplan con la característica de alta cohesión y bajo acoplamiento.
Fórmula	M = 1 - A/T M -> Modularidad T -> Total de módulos que constituyen el sistema A -> Módulos creados que cumplen con la característica de alta cohesión y bajo acoplamiento.
Interpretación	Rango de resultados 0 <= M <= 1 Si M tiende a 0 el sistema cumple con el atributo de modularidad.

Fuente: Elaboración del autor.

2) Característica: Control

2.1) Subcaracterística: Revisiones técnicas

2.1.1) Métrica:

Tabla #10. Métricas de la subcaracterística revisiones técnicas del modelo propuesto

Nombre	Validación del cumplimiento técnico de los entregables.
Tipo de métrica	Métricas de desarrollo.
Medición	¿Cada entregable del proyecto de desarrollo de software cumple con los estándares técnicos y funcionales?

Método de aplicación	Asegurar que cada entregable cumpla con los requerimientos funcionales y técnicos definidos en su alcance.
Interpretación	Si -> Cumple con el atributo No -> No cumple con el atributo

Fuente: Elaboración del autor.

2.2) Subcaracterística: Seguimiento funcional

2.2.1) Métrica:

Tabla #11. Métricas de la subcaracterística Seguimiento funcional del modelo propuesto

Nombre	Evaluación del cumplimiento de las funciones de los módulos, en el tiempo estipulado para su desarrollo.
Tipo de métrica	Métricas de desarrollo.
Medición	¿Cada módulo fue desarrollo en el tiempo estipulado, cumpliendo las funciones para las cuales fue diseñado?
Método de aplicación	Asegurar el cumplimiento de los tiempos de desarrollo y los objetivos bajo los cuales se diseñó cada módulo.
Interpretación	Si -> Cumple con el atributo No -> No cumple con el atributo

Fuente: Elaboración del autor.

3) Característica: Depuración

3.1) Subcaracterística: Funcional

3.1.1) Métrica:

Tabla #12. Métricas de la subcaracterística funcional del modelo propuesto

Nombre	Elaboración de pruebas que validen el comportamiento funcional del software
Tipo de métrica	Métricas de desarrollo
Medición	¿El software cumple con los requisitos funcionales que satisface las necesidades del usuario?
Método de aplicación	Asegurar mediante proceso de pruebas que el software desarrollado cumple con las funciones para las cuales fue creado.
Interpretación	Si -> Cumple con el atributo No -> No cumple con el atributo

Fuente: Elaboración del autor.

3.2) Subcaracterística: Unitaria

3.2.1) Métrica:

Tabla #13. Métricas de la subcaracterística unitaria del modelo propuesto

Nombre	Elaboración de pruebas que validen que cada módulo que compone el sistema cumple con las funciones descritas en su diseño.
Tipo de métrica	Métricas de desarrollo
Medición	¿Cada módulo que constituye el sistema cumple con los requisitos funcionales descritos en su diseño?
Método de aplicación	Asegurar mediante proceso de pruebas que cada módulo que compone el sistema cumple con las funciones para las cuales fue creado.
Interpretación	Si -> Cumple con el atributo No -> No cumple con el atributo

Fuente: Elaboración del autor.

3.3) Subcaracterística: Rendimiento

3.3.1) Métrica:

Tabla #14. Métricas de la subcaracterística rendimiento del modelo propuesto

Nombre	Elaboración de pruebas que validen tiempos de respuesta del sistema en condiciones específicas de trabajo.
Tipo de métrica	Métricas de desarrollo
Medición	¿Los tiempos de respuesta del sistema, una vez sometido a cargas específicas de trabajo, son los esperados?
Método de aplicación	Asegurar mediante proceso de pruebas que lo tiempos de respuesta del sistema son adecuados a su diseño.
Interpretación	Si -> Cumple con el atributo No -> No cumple con el atributo

Fuente: Elaboración del autor.

3.4) Subcaracterística: Seguridad

3.4.1) Métrica:

Tabla #15. Métricas de la subcaracterística seguridad del modelo propuesto

Nombre	Elaboración de pruebas que indiquen las vulnerabilidades y riesgos a los cuales está expuesto el sistema.
Tipo de métrica	Métricas de desarrollo
Medición	¿Las vulnerabilidades y riesgos a los cuales se encuentra expuesto el sistema fueron evaluados y mitigados en su mayoría?

Método de aplicación	Asegurar mediante proceso de pruebas de seguridad que el sistema se encuentra protegido de los riesgos y vulnerabilidades más comunes.
Interpretación	Si -> Cumple con el atributo No -> No cumple con el atributo

Fuente: Elaboración del autor.

4) Característica: Formalidad

4.1) Subcaracterística: Documentación mantenimiento

4.1.1) Métrica:

Tabla #16. Métricas de la subcaracterística documentación mantenimiento del modelo propuesto

Nombre	Documentación con el histórico de los cambios realizados al sistema.
Tipo de métrica	Métricas de desarrollo
Medición	¿Se lleva cabo una documentación estructurada con el histórico de cambios preventivos o correctivos del sistema y sus implicaciones e impacto en el diseño original?
Método de aplicación	Documentar los cambios realizados en el sistema a partir de su diseño, con el objetivo de mantener dicho histórico y facilitar futuras actividades de mantenimiento y mejor comprensión de los programadores que las lleven a cabo
Interpretación	Si -> Cumple con el atributo No -> No cumple con el atributo

Fuente: Elaboración del autor.

4.2) Subcaracterística: Documentación técnica

4.2.1) Métrica:

Tabla #17. Métricas de la subcaracterística documentación técnica del modelo propuesto

Nombre	Documentación que contiene la descripción técnica y funcional del software.
Tipo de métrica	Métricas de desarrollo
Medición	¿Se lleva cabo una documentación estructurada con la descripción técnica y funcional del software?

Método de aplicación	Documentar mediante diagramas, descripción de procesos y requerimientos, entre otros, el diseño del software desarrollado, con el objetivo de facilitar actividades de innovación y la comprensión de integrantes nuevos en el equipo de desarrollo.
Interpretación	Si -> Cumple con el atributo No -> No cumple con el atributo

Fuente: Elaboración del autor.

4.2) Subcaracterística: Resultados de pruebas

4.2.1) Métrica:

Tabla #18. Métricas de la subcaracterística resultados de pruebas del modelo propuesto

Nombre	Informe con los resultados obtenidos del conjunto de pruebas recomendadas para un software.
Tipo de métrica	Métricas de desarrollo
Medición	¿Una vez concluido el proceso de pruebas al software se elaboró un informe con los resultados obtenidos?
Método de aplicación	Documentar los resultados obtenidos del proceso de pruebas al sistema, indicando las acciones a tomar de acuerdo a los resultados.
Interpretación	Si -> Cumple con el atributo No -> No cumple con el atributo

Fuente: Elaboración del autor.

5) Característica: Eficacia

5.1) Subcaracterística: Estabilidad

5.1.1) Métrica:

Tabla #19. Métricas de la subcaracterística estabilidad del modelo propuesto

Nombre	Nivel de fallos mínimo que debe tener el software para responder de manera óptima
Tipo de métrica	Métricas Técnicas
Método de aplicación	Determinar la cantidad de módulos que presentan fallas en tiempo de ejecución.
Fórmula	E = 1 - MF/MT E -> Estabilidad MT -> Total de módulos que constituyen el sistema MF -> Cantidad de módulos que presentan fallas

Interpretación	Rango de resultados 0 <= E <= 1 Si E tiende a 1 el sistema cumple con el atributo de estabilidad

Fuente: Elaboración del autor.

5.2) Subcaracterística: Adecuación

5.2.1) Métrica:

Tabla #20. Métricas de la subcaracterística adecuación del modelo propuesto

Nombre	Cumplimiento de funcionalidades que satisface los requerimientos y objetivos especificados por el usuario.
Tipo de métrica	Métricas de funcionalidad
Método de aplicación	Determinar la cantidad de funcionalidades incluidas en la construcción del sistemas, necesarias para satisfacer las necesidades del usuario
Fórmula	A = 1 - FC/FR A -> Adecuación FR -> Total de funcionalidades requeridas FC -> Funcionalidad cumplidas en la construcción del sistema
Interpretación	Rango de resultados 0 <= A <= 1 Si A tiende a 0 el sistema cumple con el atributo de adecuación

Fuente: Elaboración del autor.

5.3) Subcaracterística: Conformidad

5.3.1) Métrica:

Tabla #21. Métricas de la subcaracterística conformidad del modelo propuesto

Nombre	Cumplimiento de acuerdos y resultados esperados y concretados al inicio de proyecto
Tipo de métrica	Métricas de funcionalidad
Medición	¿Se cumplieron en su totalidad los acuerdos concretados y el cliente se encuentra satisfecho con los resultados obtenidos?
Método de aplicación	Determinar la satisfacción o resultados esperados, así como acuerdos cumplidos con la construcción del sistema.
Interpretación	Si -> Cumple con el atributo
	No -> No cumple con el atributo

Fuente: Elaboración del autor.

6) Característica: Integridad

6.1) Subcaracterística: Confidencialidad

6.1.1) Métrica:

Tabla #22. Métricas de la subcaracterística confidencialidad del modelo propuesto

Nombre	Control de otorgar información únicamente al personal autorizado para tal fin.
Tipo de métrica	Métricas de Seguridad
Medición	¿Sólo el personal autorizado puede tener acceso a la información que procesa el sistema?
Método de aplicación	Determinar los niveles de confidencialidad que posee el software
Interpretación	Si -> Cumple con el atributo
	No -> No cumple con el atributo

Fuente: Elaboración del autor.

6.2) Subcaracterística: Disponibilidad

6.2.1) Métrica:

Tabla #23. Métricas de la subcaracterística disponibilidad del modelo propuesto

Nombre	Tiempo de servicio.
Tipo de métrica	Métricas Técnicas
Método de aplicación	Determinar el tiempo durante el cual el sistema se mantiene respondiendo ante aquellas funciones para las cuales fue creado.
Fórmula	D = 1 - TFS/TR D -> Disponibilidad TFS -> Tiempo fuera de servicio TR -> Tiempo total de respuesta del sistema mientras se estudia este atributo
Interpretación	Rango de resultados 0 <= D <= 1 Si D tiende a 1 el sistema cumple con el atributo de disponibilidad

Fuente: Elaboración del autor.

6.3) Subcaracterística: Auditoría de acceso

6.3.1) Métrica:

Tabla #24. Métricas de la subcaracterística auditoría de acceso del modelo propuesto

Nombre	Gestión los accesos a la información del software y registrar cualquier modificación que el usuario realice.
Tipo de métrica	Métricas de Seguridad

Medición	¿El sistema desarrollado gestiona y registra los accesos de los usuarios a la información que este procesa, así como cualquier cambio o modificación que se lleve a cabo?
Método de aplicación	Controlar el acceso y acciones que el usuario realice al ingresar al sistema.
Interpretación	Si -> Cumple con el atributo
	No -> No cumple con el atributo

Fuente: Elaboración del autor.

7) Característica: Reusabilidad

7.1) Subcaracterística: Generalidad

7.1.1) Métrica:

Tabla #25. Métricas de la subcaracterística generalidad del modelo propuesto

Nombre	Buscar la respuesta más general de un problema.
Tipo de métrica	Métricas de software
Medición	¿El sistema aportar las soluciones que abarquen la mayor cantidad de casos posibles?
Método de aplicación	Determinar el nivel de generalidad con que se desarrolló el software.
Interpretación	Si -> Cumple con el atributo
	No -> No cumple con el atributo

Fuente: Elaboración del autor.

7.2) Subcaracterística: Independencia de módulos

7.2.1) Métrica:

Tabla #26. Métricas de la subcaracterística independencia de módulos del modelo propuesto

Nombre	Bajo acoplamiento de los módulos que constituyen el sistema
Tipo de métrica	Métricas de software
Método de aplicación	Determinar el nivel de dependencia de cada uno de los módulos que constituyen el sistema
Fórmula	IM = 1 - MD/TM D -> Independencia de módulos MD -> Módulos de los cuales depende su funcionamiento TM -> Total de módulos del sistema.
Interpretación	Rango de resultados 0 <= IM <= 1 Si IM tiende a 1 el módulo posee bajo acoplamiento.

Fuente: Elaboración del autor.

8) Característica: Comprensibilidad

8.1) Subcaracterística: Atractividad

8.1.1) Métrica:

Tabla #27. Métricas de la subcaracterística Atractividad del modelo propuesto

Nombre	Evaluación de la capacidad del software de ser agradable ante el usuario.
Tipo de métrica	Métricas de complejidad
Medición	¿El diseño de las vistas del sistema es agradable e intuitivo para el usuario final?
Método de aplicación	Determinar el nivel de atractividad de las vistas del sistema para el usuario final.
Interpretación	Si -> Cumple con el atributo
	No -> No cumple con el atributo

Fuente: Elaboración del autor.

8.2) Subcaracterística: Aprendizaje

8.2.1) Métrica:

Tabla #28. Métricas de la subcaracterística aprendizaje del modelo propuesto

Nombre	Medir el esfuerzo requerido por el usuario para aprender sobre el manejo y uso del sistema
Tipo de métrica	Métricas de complejidad
Método de aplicación	Determinar si el esfuerzo que debe invertir el usuario en aprender el manejo del sistema es mayor al estipulado en el entrenamiento
Fórmula	AP = 1 - HE/HT AP -> Aprendizaje HE -> Horas adicionales de entrenamiento para el manejo del sistema HT -> Horas total de entrenamiento para el manejo del sistema
Interpretación	Rango de resultados 0 <= AP <= 1 Si AP tiende a 1 el esfuerzo requerido por los usuarios para aprender el manejo del sistema es únicamente el necesario.

Fuente: Elaboración del autor.

8.3) Subcaracterística: Operatividad

8.3.1) Métrica:

Tabla #29. Métricas de la subcaracterística operatividad del modelo propuesto

Nombre	Evaluación de la capacidad del software para responder a las tareas que debe cumplir bajo el control del usuario
Tipo de métrica	Métrica de complejidad
Medición	¿El sistema lleva a cabo las tareas para las cuales fue creado bajo el control del usuario final?
Método de aplicación	Determinar el nivel de cumplimiento de las funcionalidades del sistema bajo el control del usuario
Interpretación	Si -> Cumple con el atributo
	No -> No cumple con el atributo

Fuente: Elaboración del autor.

8.4) Subcaracterística: Documentación de usuario

8.4.1) Métrica: Validaciones para errores de usuarios

Tabla #30. Métricas de la subcaracterística validaciones para errores de usuarios del modelo propuesto

Nombre	Medir la capacidad del software de proteger los datos que procesa el sistema y evitar errores en tiempo de ejecución causados por el manejo del usuario.
Tipo de métrica	Métrica de complejidad
Método de aplicación	Determinar si el sistema proteger los datos que procesa el sistema y evitar errores en tiempo de ejecución causados por el manejo del usuario.
Fórmula	VE = 1 - MIV/MTV AP -> Validaciones para errores de usuarios MIV -> Cantidad de módulos que validan la información sensible a errores de los usuarios MTV -> Total de módulos que manejan información sensible a errores de los usuarios
Interpretación	Rango de resultados 0 <= VE <= 1 Si VE tiende a 0 el sistema cumple con la capacidad de proteger los datos y a sí mismo de errores de los usuarios.

Fuente: Elaboración del autor.

8.5) Subcaracterística: Documentación de usuario

8.5.1) Métrica:

Tabla #31. Métricas de la subcaracterística documentación de usuario del modelo propuesto

Nombre	Documentación orientada a la facilidad de aprendizaje del usuario
Tipo de métrica	Métricas de complejidad
Medición	¿Se redactó un manual de usuario que facilite el proceso de enseñanza al usuario y permita consultarlo ante cualquier duda de su funcionamiento?
Método de aplicación	Documentar como el usuario debe manejar la aplicación para llevar a cabo las tareas para las cuales fue creado, sin depender de los desarrolladores del producto.
Interpretación	Si -> Cumple con el atributo No -> No cumple con el atributo

Fuente: Elaboración del autor.

9) Característica: Eficiencia

9.1) Subcaracterística: Uso de recursos

9.1.1) Métrica:

Tabla #32. Métricas de la subcaracterística uso de recursos del modelo propuesto

Nombre	Evaluación del consumo de recursos por la aplicación.
Tipo de métrica	Métricas de efectividad
Medición	¿El sistema hace uso únicamente de los recursos necesarios para procesar las peticiones?
Método de aplicación	Determinar la cantidad de recursos consumidos por la aplicación y compararlo con los valores esperados.
Interpretación	Si -> Cumple con el atributo
	No -> No cumple con el atributo

Fuente: Elaboración del autor.

9.2) Subcaracterística: Tiempo de respuesta

9.2.1) Métrica:

Tabla #33. Métricas de la subcaracterística tiempo de respuesta del modelo propuesto

Nombre	Evaluación de los tiempos de respuesta del sistema
Tipo de métrica	Métricas de efectividad
Método de aplicación	Determinar si el sistema cumple con los tiempos de respuesta esperados.
Fórmula	ATR = 1 - TE/TR ATR -> Tiempo de respuesta

	TR -> Tiempo real de respuesta TE -> Tiempo estimado de respuesta
Interpretación	Rango de resultados 0 <= ATR <= 1 Si ATR tiende a 0 el sistema cumple con los tiempos de respuesta diseñados.

Fuente: Elaboración del autor.

10) Característica: Mantenibilidad

10.1) Subcaracterística: Anticipación al cambio

10.1.1) Métrica:

Tabla #34. Métricas de la subcaracterística anticipación al cambio del modelo propuesto

Nombre	Capacidad de la arquitectura del software de anticipar cambios bien sea preventivos o correctivos.
Tipo de métrica	Métricas de cambio
Medición	¿El sistema está capacitado para anticipar cambios bien sea preventivos o correctivos?
Método de aplicación	Determinar la capacidad de cambio del sistema
Interpretación	Si -> Cumple con el atributo
	No -> No cumple con el atributo

Fuente: Elaboración del autor.

10.2) Subcaracterística: Facilidad de cambio

10.2.1) Métrica:

Tabla #35. Métricas de la subcaracterística facilidad de cambio del modelo propuesto

Nombre	Medir el esfuerzo requerido por el programador para llevar a cabo un cambio en el sistema
Tipo de métrica	Métricas de cambio
Método de aplicación	Determinar si el esfuerzo que debe invertir el programador en llevar a cabo alguna modificación en el sistema es mayor al estipulado.
Fórmula	FC = 1 - HE/HIP FC -> Facilidad de cambio HE -> Horas estipuladas para el cambio HIP -> Horas invertidas por el programador
Interpretación	Rango de resultados 0 <= FC <= 1 Si FC tiende a 0 el esfuerzo requerido por el programador para llevar a cabo un cambio es únicamente el necesario.

Fuente: Elaboración del autor.

10.3) Subcaracterística: Estabilidad

10.3.1) Métrica:

Tabla #36. Métricas de la subcaracterística estabilidad del modelo propuesto

Nombre	Evaluación de la capacidad de cada módulo del software de mantener los niveles de fallos o errores mínimos ante cualquier modificación realizada
Tipo de métrica	Métricas de cambio
Método de aplicación	Determinar el nivel de riesgo de llevar a cabo un cambio en el sistema
Fórmula	E = 1 - CE/CT AP -> Estabilidad CE -> Cambios exitosos CT -> Cambios totales planificados
Interpretación	Rango de resultados 0 <= E <= 1 Si E tiende a 0 el sistema es estable ante los cambios.

Fuente: Elaboración del autor.

10.4) Subcaracterística: Facilidad de diagnóstico

10.4.1) Métrica:

Tabla #37. Métricas de la subcaracterística facilidad de diagnóstico del modelo propuesto

Nombre	Capacidad de la arquitectura del software de anticipar cambios bien sea preventivos o correctivos.
Tipo de métrica	Métricas de cambio
Medición	¿El sistema está capacitado para anticipar cambios bien sea preventivos o correctivos?
Método de aplicación	Determinar la capacidad de cambio del sistema
Interpretación	Si -> Cumple con el atributo
	No -> No cumple con el atributo

Fuente: Elaboración del autor.

11) Característica: Portabilidad

11.1) Subcaracterística: Adaptabilidad

11.1.1) Métrica:

Tabla #38. Métricas de la subcaracterística adaptabilidad del modelo propuesto

Nombre	Capacidad del software de funcionar de manera correcta en un ambiente distinto al cual se ha instalado inicialmente.
Tipo de métrica	Métricas de cambio

Medición	¿El sistema está preparado para funcionar de manera correcta en un ambiente (hardware y software) distinto al cual se ha instalado inicialmente?
Método de aplicación	Determinar la capacidad de cambio de ambiente del sistema
Interpretación	Si -> Cumple con el atributo
	No -> No cumple con el atributo

Fuente: Elaboración del autor.

11.2) Subcaracterística: Interoperabilidad

11.2.1) Métrica:

Tabla #39. Métricas de la subcaracterística interoperabilidad del modelo propuesto

Nombre	Capacidad del software de coexistir con otros sistemas e intercambiar recursos
Tipo de métrica	Métricas de cambio
Medición	¿El sistema está capacitado para coexistir con otros sistemas e intercambiar recursos sin que esto afecte su funcionamiento?
Método de aplicación	Determinar la capacidad de coexistir con otros sistemas e intercambiar recursos
Interpretación	Si -> Cumple con el atributo
	No -> No cumple con el atributo

Fuente: Elaboración del autor.

CAPÍTULO VI: EVALUACIÓN DE LA PROPUESTA

El presente capítulo se lleva a cabo con la finalidad de validar el modelo propuesto en el presente trabajo de investigación, a través de su aplicación en la página web de las empresas Sanitas y PlanSanitas, las cuales fueron seleccionadas como muestra.

A continuación se muestra la matriz de evaluación utilizada así como el análisis, conclusión y recomendaciones para cada caso.

Matriz de evaluación de la calidad para la página web de la empresa Sanitas de Venezuela

DATOS DE LA EMPRESA	
Empresa:	Sanitas Venezuela S.A
Rif:	J-30555933-3
DATOS DEL EVALUADOR	
Nombre del Evaluador:	Belkis Carrasco
Cargo en la empresa:	Analista de Aplicaciones Senior
DATOS DE LA EVALUACIÓN	
Fecha de la Evaluación:	2 de junio de 2016
Nombre de la Aplicación:	Página Web Sanitas Venezuela

Característica	Subcaracterística	Métrica	Valor correspondiente a la Aplicación			Valor en porcentaje para la evaluación
Arquitectura	Incrementalidad	¿Cumple el desarrollo del producto de software con el atributo de Incrementalidad?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%
			X			
	Abstracción	¿Los objetos del producto de software solo poseen los atributos esenciales y asociados a sus funciones?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%
			X			

Característica	Subcaracterística	Métrica	Valor correspondiente a la Aplicación			Valor en porcentaje para la evaluación
	Modularidad	**M = 1 - A/T** M -> Modularidad T -> Total de módulos que constituyen el sistema A -> Módulos creados que cumplen con la característica de alta cohesión y bajo acoplamiento.	M= 1 – 19/19 **M=0** 0 <= M <= 1			3,030%
Control	Revisiones técnicas	¿Cada entregable del proyecto de desarrollo de software cumple con los estándares técnicos y funcionales?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	
			X			3,030%
	Seguimiento funcional	¿Cada módulo fue desarrollo en el tiempo estipulado, cumpliendo las funciones para las cuales fue diseñado?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%
			X			
Depuración	Funcional	¿El software cumple con los requisitos funcionales que satisface las necesidades del usuario?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%
			X			
	Unitaria	¿Cada módulo que	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%

Característica	Subcaracterística	Métrica	Valor correspondiente a la Aplicación			Valor en porcentaje para la evaluación
		constituye el sistema cumple con los requisitos funcionales descritos en su diseño?	X			
	Rendimiento	¿Cada módulo que constituye el sistema cumple con los requisitos funcionales descritos en su diseño?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%
			X			
	Seguridad	¿Las vulnerabilidades y riesgos a los cuales se encuentra expuesto el sistema fueron evaluados y mitigados en su mayoría?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	0%
					X	
Formalidad	Documentación mantenimiento	¿Se lleva cabo una documentación estructurada con el histórico de cambios preventivos o correctivos del sistema y sus implicaciones e impacto en el diseño original?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	0%
					X	
	Documentación técnica	¿Se lleva cabo una	**SI**	**Medianamente (50% -> 1,515)**	**NO**	0%

Característica	Subcaracterística	Métrica	Valor correspondiente a la Aplicación			Valor en porcentaje para la evaluación
		documentación estructurada con la descripción técnica y funcional del software?			X	
	Resultados de pruebas	¿Una vez concluido el proceso de pruebas al software se elaboró un informe con los resultados obtenidos?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	0%
					X	
Eficacia	Estabilidad	**E = 1 - MF/MT** E -> Estabilidad MT -> Total de módulos que constituyen el sistema MF -> Cantidad de módulos que presentan fallas	E = 1 – 0/19 **E=1** 0 <= E <= 1			3,030%
	Adecuación	**A = 1 - FC/FR** A -> Adecuación FR -> Total de funcionalidades requeridas FC -> Funcionalidad cumplidas en la construcción del sistema	A = 1 – 15/15 **A=0** 0 <= A <= 1			3,030%

Característica	Subcaracterística	Métrica	Valor correspondiente a la Aplicación			Valor en porcentaje para la evaluación
	Conformidad	¿Se cumplieron en su totalidad los acuerdos concretados y el cliente se encuentra satisfecho con los resultados obtenidos?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%
			X			
Integridad	Confidencialidad	¿Sólo el personal autorizado puede tener acceso a la información que procesa el sistema?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%
			X			
	Disponibilidad	**D = 1 - TFS/TR** D -> Disponibilidad TFS -> Tiempo fuera de servicio TR -> Tiempo total de respuesta del sistema mientras se estudia este atributo	D = 1 – 0,16 h/8 h **D= 0,98** 0 <= D <= 1			2,969%
	Auditoría de acceso	¿El sistema desarrollado gestiona y registra los accesos de los usuarios a la información que este procesa, así como cualquier	**SI**	**Medianamente (50% -> 1,515)**	**NO**	0%
					X	

Característica	Subcaracterística	Métrica	Valor correspondiente a la Aplicación			Valor en porcentaje para la evaluación
		cambio o modificación que se lleve a cabo?				
Reusabilidad	Generalidad	¿El sistema aportar las soluciones que abarquen la mayor cantidad de casos posibles?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%
			X			
	Independencia de módulos	**IM = 1 - MD/TM** IM -> Independencia de módulos MD -> Módulos de los cuales depende su funcionamiento TM -> Total de módulos del sistema.	IM = 1 – 0/19 **IM= 1** 0 <= IM <= 1			3,030%
Comprensibilidad	Atractividad	¿El diseño de las vistas del sistema es agradable e intuitivo para el usuario final?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%
			X			
	Aprendizaje	**AP = 1 - HE/HT** AP -> Aprendizaje HE -> Horas adicionales de entrenamiento para el manejo del sistema HT -> Horas total de	AP = 1 – 0/8 **AP= 1** 0 <= AP <= 1			3,030%

<table>
<tr><th>Característica</th><th>Subcaracterística</th><th>Métrica</th><th colspan="3">Valor correspondiente a la Aplicación</th><th>Valor en porcentaje para la evaluación</th></tr>
<tr><td rowspan="4"></td><td></td><td>entrenamiento para el manejo del sistema</td><td colspan="3"></td><td></td></tr>
<tr><td rowspan="2">Operatividad</td><td rowspan="2">¿El sistema lleva a cabo las tareas para las cuales fue creado bajo el control del usuario final?</td><td>SI</td><td>Medianamente (50% -> 1,515)</td><td>NO</td><td rowspan="2">3,030%</td></tr>
<tr><td>X</td><td></td><td></td></tr>
<tr><td>Validaciones para errores de usuarios</td><td>VE = 1 - MIV/MTV
VE -> Validaciones para errores de usuarios
MIV -> Cantidad de módulos que validan la información sensible a errores de los usuarios
MTV -> Total de módulos que manejan información sensible a errores de los usuarios.</td><td colspan="3">VE = 1 – 7/7

VE = 0

0 <= VE <= 1</td><td>3,030%</td></tr>
<tr><td rowspan="2">Documentación de usuario</td><td rowspan="2">¿Se redactó un manual de usuario que facilite el proceso de enseñanza al usuario y permita consultarlo ante</td><td>SI</td><td>Medianamente (50% -> 1,515)</td><td>NO</td><td rowspan="2">0%</td></tr>
<tr><td></td><td></td><td>X</td></tr>
</table>

Característica	Subcaracterística	Métrica	Valor correspondiente a la Aplicación			Valor en porcentaje para la evaluación
		cualquier duda de su funcionamiento?				
Eficiencia	Uso de recursos	¿El sistema hace uso únicamente de los recursos necesarios para procesar las peticiones?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	0%
					X	
	Tiempo de respuesta	**ATR = 1 - TE/TR** ATR -> Tiempo de respuesta TR -> Tiempo real de respuesta TE -> Tiempo estimado de respuesta	ATR = 1 – 3.59 ms / 3.52 ms **ATR= 0.01** 0 <= ATR <= 1			3,030%
Mantenibilidad	Anticipación al cambio	¿El sistema está capacitado para anticipar cambios bien sea preventivos o correctivos?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%
			X			
	Facilidad de cambio	**FC = 1 - HE/HIP** FC -> Facilidad de cambio HE -> Horas estipuladas para el cambio HIP -> Horas invertidas por el programador	FC = 1 – 40/40 **FC = 0** 0 <= FC <= 1			3,030%

Característica	Subcaracterística	Métrica	Valor correspondiente a la Aplicación			Valor en porcentaje para la evaluación
	Estabilidad	**E = 1 - CE/CT** AP -> Estabilidad CE -> Cambios exitosos CT -> Cambios totales planificados	E = 1 – 4/4 = 0 **E=0** 0 <= E <= 1			3,030%
	Facilidad de diagnóstico	¿El sistema está capacitado para anticipar cambios bien sea preventivos o correctivos?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%
			X			
Portabilidad	Adaptabilidad	¿El sistema está preparado para funcionar de manera correcta en un ambiente (hardware y software) distinto al cual se ha instalado inicialmente?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%
			X			
	Interoperabilidad	¿El sistema está capacitado para coexistir con otros sistemas e intercambiar recursos sin que esto afecte su funcionamiento?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%
			X			

Matriz de evaluación de la calidad para la página web de la empresa PlanSanitas de Venezuela

DATOS DE LA EMPRESA	
Empresa:	PlanSanitas Venezuela S.A
Rif:	J-31143354-6
DATOS DEL EVALUADOR	
Nombre del Evaluador:	Belkis Carrasco
Cargo en la empresa:	Analista de Aplicaciones Senior
DATOS DE LA EVALUACIÓN	
Fecha de la Evaluación:	2 de junio de 2016
Nombre de la Aplicación:	Página Web PlanSanitas Venezuela

Característica	Subcaracterística	Métrica	Valor correspondiente a la Aplicación			Valor en porcentaje para la evaluación
Arquitectura	Incrementalidad	¿Cumple el desarrollo del producto de software con el atributo de Incrementalidad?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%
			X			
	Abstracción	¿Los objetos del producto de software solo poseen los atributos esenciales y asociados a sus funciones?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%
			X			
	Modularidad	**M = 1 - A/T** M -> Modularidad T -> Total de módulos que constituyen el sistema A -> Módulos creados que cumplen con la característica de alta cohesión y bajo acoplamiento	M= 1 – 19/19 **M=0** 0 <= M <= 1			3,030%

Característica	Subcaracterística	Métrica	Valor correspondiente a la Aplicación			Valor en porcentaje para la evaluación
		.				
Control	Revisiones técnicas	¿Cada entregable del proyecto de desarrollo de software cumple con los estándares técnicos y funcionales?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%
			X			
	Seguimiento funcional	¿Cada módulo fue desarrollo en el tiempo estipulado, cumpliendo las funciones para las cuales fue diseñado?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%
			X			
Depuración	Funcional	¿El software cumple con los requisitos funcionales que satisface las necesidades del usuario?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%
			X			
	Unitaria	¿Cada módulo que constituye el sistema cumple con los requisitos funcionales descritos en su diseño?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%
			X			

Característica	Subcaracterística	Métrica	Valor correspondiente a la Aplicación			Valor en porcentaje para la evaluación
	Rendimiento	¿Cada módulo que constituye el sistema cumple con los requisitos funcionales descritos en su diseño?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%
			X			
	Seguridad	¿Las vulnerabilidades y riesgos a los cuales se encuentra expuesto el sistema fueron evaluados y mitigados en su mayoría?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	0%
					X	
Formalidad	Documentación mantenimiento	¿Se lleva cabo una documentación estructurada con el histórico de cambios preventivos o correctivos del sistema y sus implicaciones e impacto en el diseño original?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	
					X	0%
	Documentación técnica	¿Se lleva cabo una documentación estructurada con la descripción técnica y funcional del software?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	0%
					X	

<table>
<tr><th>Característica</th><th>Subcaracterística</th><th>Métrica</th><th colspan="3">Valor correspondiente a la Aplicación</th><th>Valor en porcentaje para la evaluación</th></tr>
<tr><td rowspan="2"></td><td rowspan="2">Resultados de pruebas</td><td rowspan="2">¿Una vez concluido el proceso de pruebas al software se elaboró un informe con los resultados obtenidos?</td><td>SI</td><td>Medianamente (50% -> 1,515)</td><td>NO</td><td rowspan="2">0%</td></tr>
<tr><td></td><td></td><td>X</td></tr>
<tr><td rowspan="4">Eficacia</td><td>Estabilidad</td><td>E = 1 - MF/MT
E -> Estabilidad
MT -> Total de módulos que constituyen el sistema
MF -> Cantidad de módulos que presentan fallas</td><td colspan="3">E = 1 – 0/19
E=1
0 <= E <= 1</td><td>3,030%</td></tr>
<tr><td>Adecuación</td><td>A = 1 - FC/FR
A -> Adecuación
FR -> Total de funcionalidades requeridas
FC -> Funcionalidad cumplidas en la construcción del sistema</td><td colspan="3">A = 1 – 15/15
A=0
0 <= A <= 1</td><td>3,030%</td></tr>
<tr><td rowspan="2">Conformidad</td><td rowspan="2">¿Se cumplieron en su totalidad los acuerdos concretados y el cliente se encuentra satisfecho</td><td>SI</td><td>Medianamente (50% -> 1,515)</td><td>NO</td><td rowspan="2">3,030%</td></tr>
<tr><td>X</td><td></td><td></td></tr>
</table>

Característica	Subcaracterística	Métrica	Valor correspondiente a la Aplicación			Valor en porcentaje para la evaluación
		con los resultados obtenidos?				
Integridad	Confidencialidad	¿Sólo el personal autorizado puede tener acceso a la información que procesa el sistema?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%
			X			
	Disponibilidad	**D = 1 - TFS/TR** D -> Disponibilidad TFS -> Tiempo fuera de servicio TR -> Tiempo total de respuesta del sistema mientras se estudia este atributo	D = 1 – 0,16 h/8 h **D= 0,98** 0 <= D <= 1			2,969%
	Auditoría de acceso	¿El sistema desarrollado gestiona y registra los accesos de los usuarios a la información que este procesa, así como cualquier cambio o modificación que se lleve a cabo?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	0%
					X	
Reusabilidad	Generalidad	¿El sistema aportar las soluciones	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%

Característica	Subcaracterística	Métrica	Valor correspondiente a la Aplicación			Valor en porcentaje para la evaluación
		que abarquen la mayor cantidad de casos posibles?	X			
	Independencia de módulos	**IM = 1 - MD/TM** D -> Independencia de módulos MD -> Módulos de los cuales depende su funcionamiento TM -> Total de módulos del sistema.	IM = 1 – 0/19 **IM= 1** 0 <= IM <= 1			3,030%
Comprensibilidad	Atractividad	¿El diseño de las vistas del sistema es agradable e intuitivo para el usuario final?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%
			X			
	Aprendizaje	**AP = 1 - HE/HT** AP -> Aprendizaje HE -> Horas adicionales de entrenamiento para el manejo del sistema HT -> Horas total de entrenamiento para el manejo del sistema	AP = 1 – 0/8 **AP= 1** 0 <= AP <= 1			3,030%
	Operatividad	¿El sistema lleva a cabo	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%

Característica	Subcaracterística	Métrica	Valor correspondiente a la Aplicación			Valor en porcentaje para la evaluación
		las tareas para las cuales fue creado bajo el control del usuario final?	X			
	Validaciones para errores de usuarios	**VE = 1 - MIV/MTV** AP -> Validaciones para errores de usuarios MIV -> Cantidad de módulos que validan la información sensible a errores de los usuarios MTV -> Total de módulos que manejan información sensible a errores de los usuarios	VE = 1 – 7/7 **VE = 0** 0 <= VE <= 1			3,030%
	Documentación de usuario	¿Se redactó un manual de usuario que facilite el proceso de enseñanza al usuario y permita consultarlo ante cualquier duda de su funcionamiento?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	0%
					X	
Eficiencia	Uso de recursos	¿El sistema hace uso únicamente de los recursos necesarios	**SI**	**Medianamente (50% -> 1,515)**	**NO**	
					X	0%

Característica	Subcaracterística	Métrica	Valor correspondiente a la Aplicación			Valor en porcentaje para la evaluación
		para procesar las peticiones?				
	Tiempo de respuesta	**ATR = 1 - TE/TR** ATR -> Tiempo de respuesta TR -> Tiempo real de respuesta TE -> Tiempo estimado de respuesta	ATR = 1 – 3.42 ms / 4.8 ms **ATR = 0.28** 0 <= ATR <= 1			2,1816%
Mantenibilidad	Anticipación al cambio	¿El sistema está capacitado para anticipar cambios bien sea preventivos o correctivos?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%
			X			
	Facilidad de cambio	**FC = 1 - HE/HIP** FC -> Facilidad de cambio HE -> Horas estipuladas para el cambio HIP -> Horas invertidas por el programador	FC = 1 – 40/40 **FC = 0** 0 <= FC <= 1			3,030%
	Estabilidad	**E = 1 - CE/CT** AP -> Estabilidad CE -> Cambios exitosos CT -> Cambios totales planificados	E = 1 – 4/4 = 0 **E=0** 0 <= E <= 1			3,030%

Característica	Subcaracterística	Métrica	Valor correspondiente a la Aplicación			Valor en porcentaje para la evaluación
	Facilidad de diagnóstico	¿El sistema está capacitado para anticipar cambios bien sea preventivos o correctivos?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%
			X			
Portabilidad	Adaptabilidad	¿El sistema está preparado para funcionar de manera correcta en un ambiente (hardware y software) distinto al cual se ha instalado inicialmente?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%
			X			
	Interoperabilidad	¿El sistema está capacitado para coexistir con otros sistemas e intercambiar recursos sin que esto afecte su funcionamiento?	**SI**	**Medianamente (50% -> 1,515)**	**NO**	3,030%
			X			

Análisis de la evaluación de la calidad

A continuación se listan los resultados de la evaluación de la calidad de la página web de las empresas Sanitas de Venezuela y PlanSanitas de Venezuela, donde se calculó para cada métrica el porcentaje de cumplimiento, y de esta forma compararlos con los valores que se esperan tener para que el producto sea considerado 100% de calidad.

Tabla #40. Resultados aplicación del modelo propuesto a la página web de la empresa Sanitas de Venezuela

Característica	**Subcaracterística**	**Representación esperado para la evaluación en porcentaje**	**Representación real para la evaluación en porcentaje**
Arquitectura	Incrementalidad	3,030%	3,030%
	Abstracción	3,030%	3,030%
	Modularidad	3,030%	3,030%
Control	Revisiones técnicas	3,030%	3,030%
	Seguimiento funcional	3,030%	3,030%
Depuración	Funcional	3,030%	3,030%
	Unitaria	3,030%	3,030%
	Rendimiento	3,030%	3,030%
	Seguridad	3,030%	0%
Formalidad	Documentación mantenimiento	3,030%	0%
	Documentación técnica	3,030%	0%
	Resultados de pruebas	3,030%	0%
Eficacia	Estabilidad	3,030%	3,030%
	Adecuación	3,030%	3,030%
	Conformidad	3,030%	3,030%
Integridad	Confidencialidad	3,030%	3,030%
	Disponibilidad	3,030%	2,969%
	Auditoría de acceso	3,030%	0%
Reusabilidad	Generalidad	3,030%	3,030%
	Independencia de módulos	3,030%	3,030%
Comprensibilidad	Atractividad	3,030%	3,030%
	Aprendizaje	3,030%	3,030%
	Operatividad	3,030%	3,030%
	Validaciones para errores de usuarios	3,030%	3,030%
	Documentación de usuario	3,030%	0%

Característica	Subcaracterística	Representación esperado para la evaluación en porcentaje	Representación real para la evaluación en porcentaje
Eficiencia	Uso de recursos	3,030%	0%
	Tiempo de respuesta	3,030%	3,030%
Mantenibilidad	Anticipación al cambio	3,030%	3,030%
	Facilidad de cambio	3,030%	3,030%
	Estabilidad	3,030%	3,030%
	Facilidad de diagnóstico	3,030%	3,030%
Portabilidad	Adaptabilidad	3,030%	3,030%
	Interoperabilidad	3,030%	3,030%

Fuente: Elaboración del autor.

Tabla #41. Resultados aplicación del modelo propuesto a la página web de la empresa PlanSanitas de Venezuela

Característica	Subcaracterística	Representación esperado para la evaluación en porcentaje	Representación real para la evaluación en porcentaje
Arquitectura	Incrementalidad	3,03%	3,03%
	Abstracción	3,03%	3,03%
	Modularidad	3,03%	3,03%
Control	Revisiones técnicas	3,03%	3,03%
	Seguimiento funcional	3,03%	3,03%
Depuración	Funcional	3,03%	3,03%
	Unitaria	3,03%	3,03%
	Rendimiento	3,03%	3,03%
	Seguridad	3,03%	0%
Formalidad	Documentación mantenimiento	3,03%	0%
	Documentación técnica	3,03%	0%
	Resultados de pruebas	3,03%	0%
Eficacia	Estabilidad	3,03%	3,03%
	Adecuación	3,03%	3,03%

Característica	Subcaracterística	Representación esperado para la evaluación en porcentaje	Representación real para la evaluación en porcentaje
	Conformidad	3,03%	3,03%
Integridad	Confidencialidad	3,03%	3,03%
	Disponibilidad	3,03%	2,97%
	Auditoría de acceso	3,03%	0%
Reusabilidad	Generalidad	3,03%	3,03%
	Independencia de módulos	3,03%	3,03%
Comprensibilidad	Atractividad	3,03%	3,03%
	Aprendizaje	3,03%	3,03%
	Operatividad	3,03%	3,03%
	Validaciones para errores de usuarios	3,03%	3,03%
	Documentación de usuario	3,03%	0%
Eficiencia	Uso de recursos	3,03%	0%
	Tiempo de respuesta	3,03%	2,18%
Mantenibilidad	Anticipación al cambio	3,03%	3,03%
	Facilidad de cambio	3,03%	3,03%
	Estabilidad	3,03%	3,03%
	Facilidad de diagnóstico	3,03%	3,03%
Portabilidad	Adaptabilidad	3,03%	3,03%
	Interoperabilidad	3,03%	3,03%

Fuente: Elaboración del autor.

La representación en porcentaje para cada métrica de los productos evaluados se representan en las figuras 40 y 41, donde se tiene un gráfico de tipo radial con cada uno de los valores obtenidos en la evaluación, donde se puede observar que cada una de las subcaracterísticas evaluadas son independientes y con la misma prioridad.

Finalmente se enuncian algunas recomendaciones para mejorar las debilidades o fallas encontradas en la evaluación para ambos productos:

1.- Realizar un análisis de vulnerabilidades y riesgos que presenta la página web, con el objetivo de mitigarlos. Esta actividad se puede llevar a cabo mediante una consultoría o asesoría externa en caso que no se disponga de un departamento dedicado a estas tareas.

2.- Elaborar la documentación de mantenimiento necesaria para llevar el control e histórico de cambios preventivos o correctivos de la página web.

3.- Elaborar la documentación técnica, que contenga la descripción funcional del software, así como los aspectos técnicos bajo los cuales fue diseñado

4.- Para futuros cambios preventivos o correctivos de la página web, realizar el informe de los resultados obtenidos en las pruebas, con el objetivo de documentar el comportamiento de las nuevas funcionales o la corrección de errores.

5.- Mantener el registro de los accesos de usuarios a la información, modificaciones que se lleven a cabo durante su visita, con datos relevantes como fecha, hora y datos cambiados.

6.- Redactar la documentación que guíen al usuario en el uso de la aplicación, permita consultarlo ante cualquier duda de su funcionamiento y no dependa de los desarrolladores del software para llevar a cabo una tarea.

7.- Realizar un estudio del consumo de recursos necesarios para procesar las peticiones, en búsqueda de fallas o inconvenientes que se encuentren afectando los tiempos de respuesta o la disponibilidad de la aplicación.

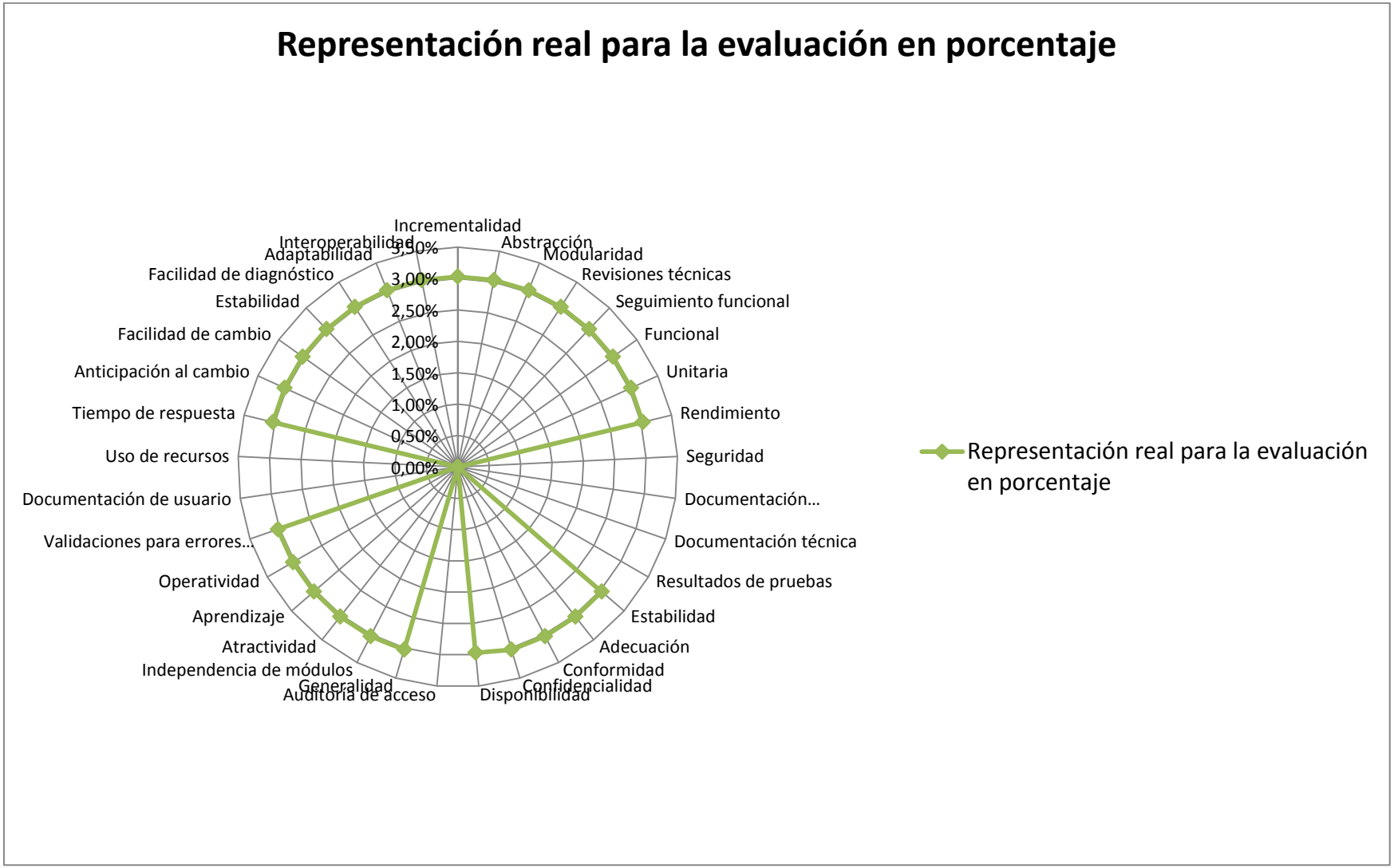

Figura #40.Gráfico de los resultados de la aplicación del modelo propuesto en la página web de la empresa Sanitas de Venezuela.
Fuente: Elaboración del autor.

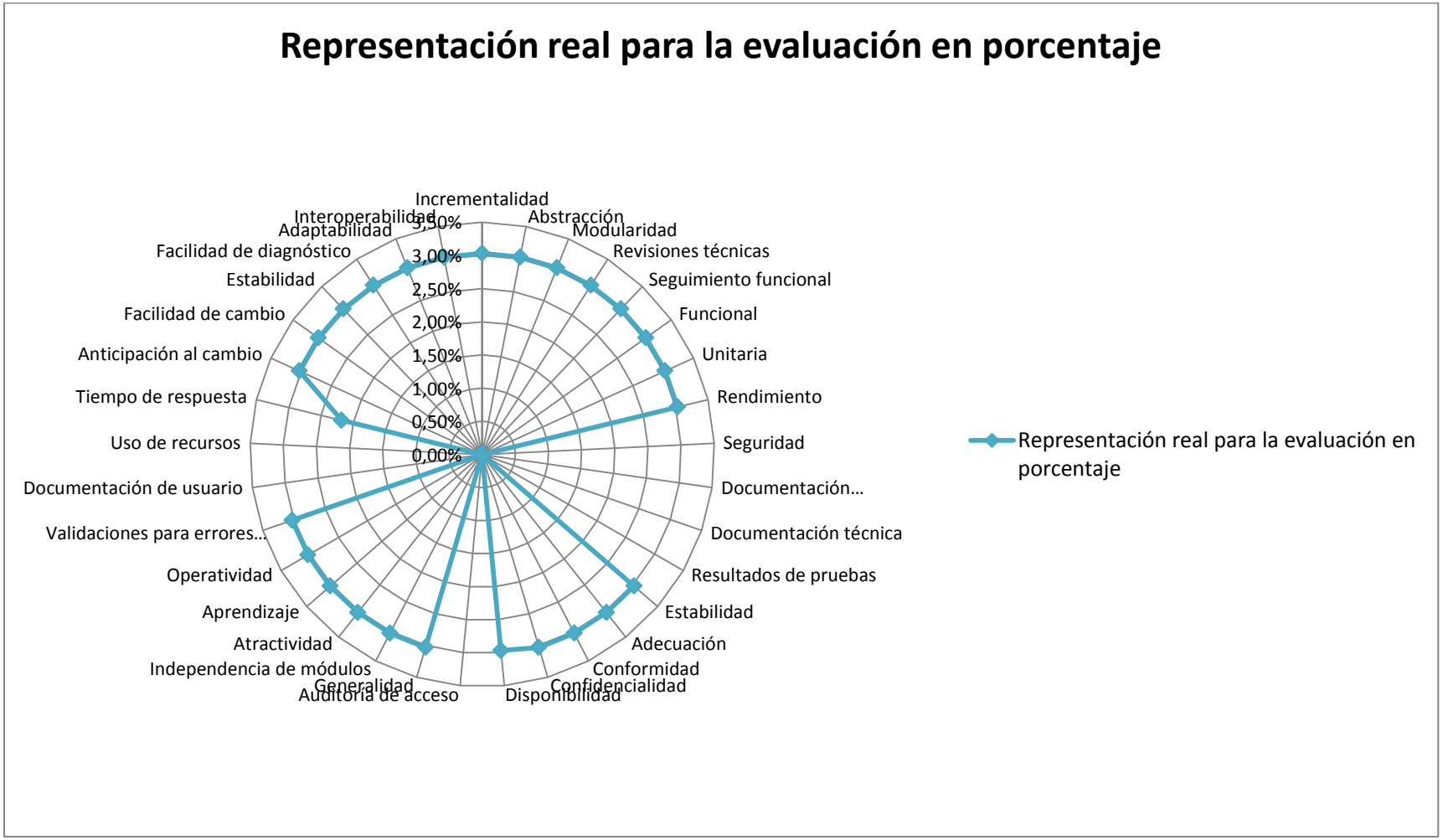

Figura #41.Gráfico de los resultados de la aplicación del modelo propuesto en la página web de la empresa PlanSanitas de Venezuela.
Fuente: Elaboración del autor.

CAPÍTULO VII: CONCLUSIONES Y RECOMENDACIONES

Los sistemas de tecnología de la información representan una unidad de gran apoyo dentro de las organizaciones, es por ello que la evaluación de su calidad se ha convertido en una tendencia.

La evaluación de la calidad del producto de software consiste en la aplicación de un conjunto de métricas que miden características y subcaracterísticas del modelo implementado.

Las normas o estándares existentes plantean que el proceso de evaluación se lleve a cabo en las fases finales del desarrollo, generando detección de errores en etapas tardías con grados de implicación máximos, lo cual representa la problemática planteada en el presente trabajo de grado. Para ello se propone como objetivo diseñar un modelo para la evaluación de la calidad del producto de software para el sector de medicina prepagada.

Para su elaboración se llevó a cabo un levantamiento de información y posterior análisis referente a los modelos, estándares o normativas que se encuentran en uso, e incluso en estudio, con la finalidad de conocer los éxitos y fracasos que se han enfrentado.

Adicionalmente fue necesario la recolección de datos de los elementos de calidad evaluados en los software de la medicina prepagada. Los instrumentos utilizados fueron la encuesta (Ver Anexo A y B) y la guía de observación (Ver Anexo C y D), aplicados a las empresas tomadas como muestra de este proyecto.

Una vez que se obtuvo las deficiencias y necesidades de este sector, se inició el diseño del modelo de evaluación de la calidad del producto de software de acuerdo a la información recolectada. En materia de calidad de producto de software, se tomó como base las normas ISO/IEC 9126.

La propuesta de este trabajo de investigación consiste en 11 características a evaluar a través de 33 subcaracterísticas y sus respectivas métricas.

Para su aplicación se creó una matriz de evaluación en la cual se indicará el cumplimiento o no y en qué medida cada una de las subcaracterísticas del modelo diseñado. (Ver Anexo J)

Finalmente para que el modelo tenga validez se debe aplicar y comprobar que a través del diseño elaborado se puede evaluar la calidad de los productos pertenecientes a las empresas que representan la muestra, los cuales fueron las páginas web oficiales de cada una de ellas.

Se realizaron un conjunto de análisis de los resultados obtenidos y se redactó un documento de certificación de la calidad donde estos fueron plasmados. (Ver Anexo K)

Recomendaciones

Continuar con el diseño de este modelo, de manera que pueda adaptarse a gran cantidad de sectores y pueda ser aplicado sin que esto involucre una gran inversión para las empresas.

Se recomienda además elaborar un artículo con el objetivo de informa el trabajo que se está llevando a cabo, y de esta forma incentivar a mas desarrolladores o empresas a probarlo y contar su experiencia, obteniendo de estos opiniones y oportunidades de mejoras al modelo.

Con el objetivo de obtener oportunidades de mejoras a través de opiniones, se sugiere incentivar a desarrolladores o empresas a aplicar el modelo propuesto y contar su experiencia.

Finalmente, desde la perspectiva de las empresas donde se aplique el modelo propuesto en este trabajo de grado de maestría, considerando que el resultado de la evaluación de la calidad de sus productos de software puede involucrar cambios y toma de decisiones para acciones correctivas que conlleven inversiones económicas, se deben proponer soluciones óptimas que permitan aproximar la subcaracterística al valor esperado al menor costo, y con tiempos de solución a corto plazo (de ser posible). En general el informe de recomendaciones debe basarse en un análisis del costo de implantar la solución propuesta y el beneficio que le suma a la calidad evaluada por el modelo.

En relación a este punto, García (2014) indica lo siguiente:

> …los principales beneficios de cumplir con los requisitos de calidad pueden incluir un menor reproceso, una mayor productividad, menores costos y una mayor satisfacción de los interesados. Un caso de negocio para cada actividad de calidad permite comparar el costo del procedimiento de calidad con el beneficio esperado.
> Hacer las cosas bien, dentro de los límites establecidos de calidad trae como resultado, beneficios que van más allá de sólo la parte económica. Obtener clientes satisfechos y no tener que repetir un trabajo son solo algunos de estos beneficios.

REFERENCIAS BIBLIOGRÁFICAS

Amo, F., Martínez, L., & Segovia, F. (2005). *Introducción a la ingeniería del software.* Zaragoza: Delta Publicaciones.

Arias, F. (1999). *El proyecto de investigación.* Caracas: Episteme.

Bertoa, M., Troya, J., & Vallecillo, A. (s.f.). *Aspectos de Calidad en el Desarrollo de Software Basado en Componentes.* Recuperado el Octubre de 2015, de http://www.researchgate.net/profile/Manuel_Bertoa/publication/228806539_Aspectos_de_calidad_en_el_desarrollo_de_software_basado_en_componentes/links/02bfe50d640cd33b77000000.pdf

Borbón, N. (12 de Marzo de 2013). *Norma de evaluación ISO/IEC 9126.* Recuperado el Octubre de 2015, de http://actividadreconocimiento-301569-8.blogspot.com/2013/03/norma-de-evaluacion-isoiec-9126.html

Carnegie Mellon University. (Febrero de 2016). Obtenido de Watts Humphrey: An Outrageous Commitment, A Lifelong Mission: https://www.sei.cmu.edu/watts/

Cendejas, J. (2014). *Implementación del modelo integral colaborativo (MDSIC) como fuente de innovación para el desarrollo ágil de software en las empresas de la zona centro - occidente en México.* Tesis de Doctorado, Universidad Popular Autónoma del Estado de Puebla, Puebla.

Constitución de la República Bolivariana de Venezuela. (1999).

Estándar ISO/IEC 9126-1. (2000).

Figarella, D., & Zamora, A. (2007). *Software venezolano de calidad internacional. El Centro de Excelencia en Ingeniería de Software (Ceisoft).* Caracas: Serie Cluster I. Obtenido de scioteca.caf.com/bitstream/handle/123456789/432/171.pdf?sequence=1&isAllowed=y

García, O. (7 de Enero de 2014). *El análisis Costo-Beneficio al planificar la calidad.* Obtenido de http://www.proyectum.lat/2014/01/07/el-analisis-costo-beneficio-al-planificar-la-calidad/

Garzón, M., Sampalo, M. d., & José, P. (2003). *Informátiva volumen IV.* Editorial Mad.

González, A. (20 de Diciembre de 2014). *Escuela de Organización Industrial.* Recuperado el Febrero de 2016, de https://www.eoi.es/blogs/mcalidadon/2014/12/20/la-estructura-desagregada-del-trabajo-edtwbs/

Guilarte, M. (15 de noviembre de 2013). *I Jornadas sobre Calidad de Producto Software.* Obtenido de http://www.muycomputerpro.com/2013/11/15/calidad-producto-software

Hernández, R., Fernández, C., & Baptista, P. (2010). *Metodología de la investigación.* México D.F.: McGraw-Hill.

Hernando, J. (Noviembre de 2013). *La Calidad del Producto Software.* Recuperado el Octubre de 2015, de http://atsistemas.com/wp-content/uploads/2013/12/20121201_articulo_calidad_producto_software_jesus_hernando_corrochano.pdf

Hurtado, J. (2010). *Guía para la comprensión holística de la ciencia.* Obtenido de http://dip.una.edu.ve/mpe/017metodologiaI/paginas/Hurtado,%20Guia%20para%20la%20comprension%20holistica%20de%20la%20ciencia%20Unidad%20III.pdf

Ley Orgánica del Sistema Venezolano para la Calidad. (2002).

Ley Orgánica N° 1.290, del 30 De Agosto de 2001, de Ciencia, Tecnología e Innovación. (s.f.).

Mendoza, L., Pérez, M., Grimán, A., & Rojas, T. (s.f.). *Algoritmo para la Evaluación de la Calidad Sistémica del Software.* Recuperado el Octubre de 2015, de http://www.lisi.usb.ve/publicaciones/02%20calidad%20sistemica/calidad_21.pdf

Mendoza, M. (Julio de 2010). *Modelo de referencia para la selección de herramientas de pruebas como soporte al proceso de desarrollo de software en pymes venezolanas.* Tesis de Maestría, Universidad Católica Andrés Bello, Caracas.

Miranda, A. (Agosto de 2013). *Plagio y Ética de la Investigación Científica.* Obtenido de http://www.scielo.cl/scielo.php?pid=S0718-34372013000200016&script=sci_arttext

Moreno, J., Bolaños, L., & Navia, M. (Junio de 2010). *Exploración de Modelos y Estándares de calidad para el producto software.* Recuperado el Octubre de 2015, de http://revistas.uis.edu.co/index.php/revistauisingenierias/article/view/1055/1434

Ortega, E., & Villavicencio, M. (s.f.). *Medición de la Calidad de Productos de Software en un Ambiente Académico Usando la Norma ISO/IEC-9126.* Recuperado el Octubre de 2015, de https://www.dspace.espol.edu.ec/bitstream/123456789/16366/1/MEDICION%20DE%20LA%20CALIDAD%20DE%20PRODUCTOS%20DE%20SOFTWARE%20USANDO%20LA%20NORMA%20ISO_IEC%209126%20EN%20UN%20AMBIENTE%20ACADE.pdf

Pentón, Á. (Septiembre de 2013). *Métrica para evaluar la seguridad de los SGIC.* Recuperado el Octubre de 2015, de https://www.google.co.ve/url?sa=t&rct=j&q=&esrc=s&source=web&cd=8&ved=0CE4QFjAHahUKEwih_LvJirzIAhVKgA0KHcyTDBY&url=http%3A%2F%2Fdialnet.unirioja.es%2Fdescarga%2Farticulo%2F4687249.pdf&usg=AFQjCNESJkHCYPYi30Vi8qpRYypt9nSooA&sig2=wT-6rMSReotPmpR-Y8Lffg&cad=

Pressman, R. (2010). *Ingeniería del Software.* México: Mc Graw Hill.

Project Management Institute, I. (2013). *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOOK)* (Quinta ed.).

Sánchez, W. (Agosto de 2011). *La usabilidad en Ingeniería de Software: definición y características.* Recuperado el Octubre de 2015, de http://www.redicces.org.sv/jspui/bitstream/10972/1937/1/2.%20La%20usabilidad%20en%20Ingenieria%20de%20Software-%20definicion%20y%20caracteristicas.pdf

Sommerville, I. (2005). *Ingeniería del Software.* Madrid: Pearson Addison Wesley.

Superintendencia de la Actividad Aseguradora (SUDEASEG). (2015). Obtenido de http://www.sudeaseg.gob.ve/

Viana, H., Gomes, J., & Jaén, M. (2008). *Cruzsalud: tecnología de información para el manejo de medicina prepagada a bajo costo.* Ediciones IESA.

ANEXOS

ANEXO A

Instrumento de recolección de Datos (Encuesta para la empresa Sanitas)

Universidad Católica Andrés Bello.
Vicerrectorado Académico.
Estudios de Postgrado
Área de Ingeniería.
Postgrado en Sistemas de Información.

En esta oportunidad requiero de su colaboración en responder las siguientes preguntas de la manera más objetiva posible. A través de ellas se busca conocer las condiciones y necesidades de la empresa **Sanitas** en materia de calidad de producto de software.

El objetivo de esta encuesta es únicamente de carácter académico, forma parte del levantamiento de información del proyecto de trabajo de grado de maestría *Modelo de evaluación de la calidad del producto de software para medicina prepagada*.

Consta de 14 preguntas, para las cuales debe indicar con una X dentro de las casillas correspondientes a la respuesta SI o NO, y en los casos que aplique escribir en el campo de observaciones la información requerida.

Considere adicionalmente que no existen respuestas buenas o malas, las mismas van acorde a su criterio, experiencia y observación dentro de la empresa.

#	Indicadores	Si	No	Observaciones
1	Posee la organización algún área encargada de evaluar y controlar la calidad de los productos de software.			
2	La empresa dispone algún auditor externo encargado de evaluar y controlar la calidad de los productos de software.			
3	El personal del área de tecnología de la empresa se encuentra capacitado en materia de calidad de software			
4	Se maneja la evaluación de la calidad de los productos de software bajo algún modelo determinado. Nota: En caso de ser afirmativo, indicar el nombre en las observaciones.			
5	Considera usted que la evaluación de la calidad de producto de software agrega/agregaría valor dentro de la organización.			

#	Indicadores	Si	No	Observaciones
6	El área de tecnología involucra al grupo de usuarios durante el desarrollo de los sistemas de información.			
7	Durante el desarrollo de software, se realiza seguimiento a los programadores, con el objetivo de validar el apego de las funcionalidades con la lista de requerimientos.			
8	Durante la etapa de diseño del software se toman en cuenta aspectos como: eficiente uso de los recursos, coexistencia, seguridad.			
9	Una vez culminado el producto de software, se realiza el levantamiento de la documentación técnica que facilite el mantenimiento de la aplicación.			
10	Para cada aplicación disponible en la empresa se maneja un manual para el usuario, que permita aclarar sus dudas o hacer más fácil su capacitación.			
11	La empresa maneja algún proceso de pruebas general para cada aplicación desarrollada.			
12	Existe un área dedicada a realizar el proceso de pruebas de cada aplicación que se desarrolla.			
13	La empresa contrata los servicios de proveedores para el desarrollo de aplicaciones.			
14	De ser afirmativa la respuesta anterior, aplica algún modelo de evaluación de la calidad a los productos que desarrollan los proveedores. Nota: Si la respuesta anterior es NO, indicar en las observaciones N/A			

Observaciones generales

_____
_____
_____

ANEXO B

Instrumento de recolección de Datos (Encuesta para la empresa PlanSanitas)

Universidad Católica Andrés Bello.
Vicerrectorado Académico.
Estudios de Postgrado
Área de Ingeniería.
Postgrado en Sistemas de Información.

En esta oportunidad requiero de su colaboración en responder las siguientes preguntas de la manera más objetiva posible. A través de ellas se busca conocer las condiciones y necesidades de la empresa **PlanSanitas** en materia de calidad de producto de software.

El objetivo de esta encuesta es únicamente de carácter académico, forma parte del levantamiento de información del proyecto de trabajo de grado de maestría *Modelo de evaluación de la calidad del producto de software para medicina prepagada*.

Consta de 14 preguntas, para las cuales debe indicar con una X dentro de las casillas correspondientes a la respuesta SI o NO, y en los casos que aplique escribir en el campo de observaciones la información requerida.

Considere adicionalmente que no existen respuestas buenas o malas, las mismas van acorde a su criterio, experiencia y observación dentro de la empresa.

#	Indicadores	Si	No	Observaciones
1	Posee la organización algún área encargada de evaluar y controlar la calidad de los productos de software.			
2	La empresa dispone algún auditor externo encargado de evaluar y controlar la calidad de los productos de software.			
3	El personal del área de tecnología de la empresa se encuentra capacitado en materia de calidad de software			
4	Se maneja la evaluación de la calidad de los productos de software bajo algún modelo determinado. Nota: En caso de ser afirmativo, indicar el nombre en las observaciones.			
5	Considera usted que la evaluación de la calidad de producto de software agrega/agregaría valor dentro de la organización.			

#	Indicadores	Si	No	Observaciones
6	El área de tecnología involucra al grupo de usuarios durante el desarrollo de los sistemas de información.			
7	Durante el desarrollo de software, se realiza seguimiento a los programadores, con el objetivo de validar el apego de las funcionalidades con la lista de requerimientos.			
8	Durante la etapa de diseño del software se toman en cuenta aspectos como: eficiente uso de los recursos, coexistencia, seguridad.			
9	Una vez culminado el producto de software, se realiza el levantamiento de la documentación técnica que facilite el mantenimiento de la aplicación.			
10	Para cada aplicación disponible en la empresa se maneja un manual para el usuario, que permita aclarar sus dudas o hacer más fácil su capacitación.			
11	La empresa maneja algún proceso de pruebas general para cada aplicación desarrollada.			
12	Existe un área dedicada a realizar el proceso de pruebas de cada aplicación que se desarrolla.			
13	La empresa contrata los servicios de proveedores para el desarrollo de aplicaciones.			
14	De ser afirmativa la respuesta anterior, aplica algún modelo de evaluación de la calidad a los productos que desarrollan los proveedores. Nota: Si la respuesta anterior es NO, indicar en las observaciones N/A			

Observaciones generales

_____
_____
_____

ANEXO C

Instrumento de recolección de Datos (Guía de Observación para la empresa Sanitas)

Universidad Católica Andrés Bello.
Vicerrectorado Académico.
Estudios de Postgrado
Área de Ingeniería.
Postgrado en Sistemas de Información.
Proyecto de trabajo de grado de maestría: **Modelo de evaluación de la calidad del producto de software para medicina prepagada.**

La siguiente guía de observación tiene como objetivo realizar el levantamiento de información del comportamiento general de las aplicaciones de la empresa **Sanitas** de acuerdo a las características de calidad más comunes dentro de los modelos existentes en el mercado

Consta de 19 aspectos a evaluar, para las cuales debe indicar con una X dentro de las casillas correspondientes al nivel de aplicación, donde 5 es *Siempre* y 0 es *Nunca*, y en los casos que aplique escribir en el campo de observaciones la información requerida.

#	Comportamiento a Evaluar	5	4	3	2	1	0	Observaciones
Funcionalidad								
1	Los productos de software cumplen con las funciones y objetivos para los cuales fueron creados.							
2	En el diseño de las aplicaciones se evalúa la interacción con otros sistemas.							
3	Los accesos a los sistemas se realizan solo por personal autorizado.							
4	Los sistemas desarrollados cumple con las normas y estándares, u otras prescripciones de la ingeniería de software							
Fiabilidad								
5	Antes de colocar un sistema en ambiente de producción, se eliminan las fallas observadas en el proceso de pruebas.							
6	En los sistemas se contemplan tareas de recuperabilidad de datos en caso de fallas.							
7	Se manejan niveles de desempeño alto en las aplicaciones, considerando de esta forma que son tolerantes a fallos.							
Usabilidad								

#	Comportamiento a Evaluar	5	4	3	2	1	0	Observaciones
8	El diseño de los sistemas es atractivo y compresible para los usuarios.							
9	El esfuerzo de los usuarios en comprender el funcionamiento de las aplicaciones es bajo.							
10	El proceso de aprendizaje y capacitación de los usuarios es extenso.							
Eficiencia								
11	Durante la construcción del software, son evaluados sus tiempos de respuesta							
12	Los recursos a consumir por la aplicación son evaluados durante su diseño y construcción							
Mantenibilidad								
13	Se pueden aplicar cambios preventivos, correctivos y de ampliación a los sistemas sin afectar su funcionamiento.							
14	Los procesos de pruebas de las aplicaciones son sencillos y no requiere de esfuerzos adicionales.							
15	El esfuerzo requerido para los cambios referentes a corrección de fallas es bajo.							
16	En caso de fallas o inconvenientes con los sistemas, el proceso de diagnóstico es rápido y sencillo.							
Portabilidad								
17	Se dispone de un plan de acción sencillo para la instalación o puesta en marcha del software.							
18	El funcionamiento de las aplicaciones no se ve afectado por el traslado a otros ambientes.							
19	Las aplicaciones pueden ser reemplazadas por otras que manejen los mismos propósitos o funciones.							

Observaciones generales

_____
_____
_____

ANEXO D

Instrumento de recolección de Datos (Guía de Observación para la empresa PlanSanitas)

Universidad Católica Andrés Bello.
Vicerrectorado Académico.
Estudios de Postgrado
Área de Ingeniería.
Postgrado en Sistemas de Información.
Proyecto de trabajo de grado de maestría: **Modelo de evaluación de la calidad del producto de software para medicina prepagada.**

La siguiente guía de observación tiene como objetivo realizar el levantamiento de información del comportamiento general de las aplicaciones de la empresa **PlanSanitas** de acuerdo a las características de calidad más comunes dentro de los modelos existentes en el mercado

Consta de 19 aspectos a evaluar, para las cuales debe indicar con una X dentro de las casillas correspondientes al nivel de aplicación, donde 5 es *Siempre* y 0 es *Nunca*, y en los casos que aplique escribir en el campo de observaciones la información requerida.

#	Comportamiento a Evaluar	5	4	3	2	1	0	Observaciones
Funcionalidad								
1	Los productos de software cumplen con las funciones y objetivos para los cuales fueron creados.							
2	En el diseño de las aplicaciones se evalúa la interacción con otros sistemas.							
3	Los accesos a los sistemas se realizan solo por personal autorizado.							
4	Los sistemas desarrollados cumple con las normas y estándares, u otras prescripciones de la ingeniería de software							
Fiabilidad								
5	Antes de colocar un sistema en ambiente de producción, se eliminan las fallas observadas en el proceso de pruebas.							
6	En los sistemas se contemplan tareas de recuperabilidad de datos en caso de fallas.							
7	Se manejan niveles de desempeño alto en las aplicaciones, considerando de esta forma que son tolerantes a fallos.							

#	Comportamiento a Evaluar	5	4	3	2	1	0	Observaciones
Usabilidad								
8	El diseño de los sistemas es atractivo y compresible para los usuarios.							
9	El esfuerzo de los usuarios en comprender el funcionamiento de las aplicaciones es bajo.							
10	El proceso de aprendizaje y capacitación de los usuarios es extenso.							
Eficiencia								
11	Durante la construcción del software, son evaluados sus tiempos de respuesta							
12	Los recursos a consumir por la aplicación son evaluados durante su diseño y construcción							
Mantenibilidad								
13	Se pueden aplicar cambios preventivos, correctivos y de ampliación a los sistemas sin afectar su funcionamiento.							
14	Los procesos de pruebas de las aplicaciones son sencillos y no requiere de esfuerzos adicionales.							
15	El esfuerzo requerido para los cambios referentes a corrección de fallas es bajo.							
16	En caso de fallas o inconvenientes con los sistemas, el proceso de diagnóstico es rápido y sencillo.							
Portabilidad								
17	Se dispone de un plan de acción sencillo para la instalación o puesta en marcha del software.							
18	El funcionamiento de las aplicaciones no se ve afectado por el traslado a otros ambientes.							
19	Las aplicaciones pueden ser reemplazadas por otras que manejen los mismos propósitos o funciones.							

Observaciones generales

_____
_____
_____

ANEXO E

Solicitud de Evaluación de Instrumento de Recolección de Datos por Experto

Universidad Católica Andrés Bello.
Vicerrectorado Académico.
Estudios de Postgrado
Área de Ingeniería.
Postgrado en Sistemas de Información.

Solicitud de Evaluación de Instrumento de Recolección de Datos por Experto

En esta oportunidad requiero de su colaboración en validar la congruencia, claridad y relación de las preguntas de los instrumento de recolección de datos **Cuestionario** y **Guía de Observación** con los objetivos planteados en el proyecto, los cuales serán aplicados con la finalidad de conocer las condiciones y necesidades de las empresas tomadas como muestra del sector de medicina preparada en materia de calidad de producto de software.

El mismo será aplicado en el trabajo de investigación: **Modelo de evaluación de la calidad del producto de software para medicina prepagada**

Agradezco de antemano la colaboración brindada.

Atentamente

_____

Ing. Belkis Carrasco

ANEXO F

Evaluación del Instrumento de Recolección de Datos

Universidad Católica Andrés Bello.
Vicerrectorado Académico.
Estudios de Postgrado
Área de Ingeniería.
Postgrado en Sistemas de Información.

Evaluación del Instrumento de Recolección de Datos

Estimado profesor, por medio del presente formato, requiero de su amable colaboración en validar la congruencia (característica que hace referencia a la relación lógica o vínculo entre dos o más cosas), claridad (hace referencia a expresar o abordar un punto con el mayor facilidad de comprensión posible) y relación de las preguntas de los instrumentos de recolección de datos con los objetivos planteados, como parte del desarrollo del trabajo de investigación: **Modelo de evaluación de la calidad del producto de software para medicina prepagada**, con la finalidad de garantizar la obtención correcta de los resultados.

CUESTIONARIO

# Pregunta	Congruencia		Claridad		Relación con los objetivos		Observaciones
	Cumple		Cumple		Cumple		
	Si	No	Si	No	Si	No	
1							
2							
3							
4							
5							

# Pregunta	Congruencia		Claridad		Relación con los objetivos		Observaciones
	Cumple		Cumple		Cumple		
	Si	No	Si	No	Si	No	
6							
7							
8							
9							
10							
11							
12							
13							
14							

GUÍA DE OBSERVACIÓN

# Pregunta	Congruencia		Claridad		Relación con los objetivos		Observaciones
	Cumple		Cumple		Cumple		
	Si	No	Si	No	Si	No	
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							

# Pregunta	Congruencia		Claridad		Relación con los objetivos		Observaciones
	Cumple		Cumple		Cumple		
	Si	No	Si	No	Si	No	
11							
12							
13							
14							
15							
16							
17							
18							
19							

ANEXO G

Constancia de Evaluación del Instrumento de Recolección de Datos Profesor I

Universidad Católica Andrés Bello.
Vicerrectorado Académico.
Estudios de Postgrado
Área de Ingeniería.
Postgrado en Sistemas de Información.

Constancia de Evaluación del Instrumento de Recolección de Datos

Yo, Pedro Nolasco Bonillo Ramos, titular de la cedula de identidad N° 10.868.538, de Profesión Doctor en Ciencias de la Computación, ejerciendo actualmente como evaluador, en la Universidad Católica Andrés Bello, por medio de la presente hago constar que he revisado con fines de validación los instrumentos de recolección de datos (Cuestionario y Guía de Observación), del Trabajo de Grado de Maestría: **Modelo de evaluación de la calidad del producto de software para medicina prepagada,** de la estudiante Belkis Elena Carrasco Sosa, cédula de identidad N° 19.453.820, quien opta por el título de Magíster en Sistemas de Información.

Sin otro particular al cual hacer referencia, se despide cordialmente,

_____

Pedro Nolasco Bonillo Ramos

C.I. 10.868.538

ANEXO H

Constancia de Evaluación del Instrumento de Recolección de Datos Profesor II

Universidad Católica Andrés Bello.
Vicerrectorado Académico.
Estudios de Postgrado
Área de Ingeniería.
Postgrado en Sistemas de Información.

Constancia de Evaluación del Instrumento de Recolección de Datos

Yo, Gloria Aponte, titular de la cedula de identidad N° 4.964.695, de Profesión Ing. Químico y Magister en Sistemas de Información, ejerciendo actualmente como evaluador, en la Universidad Católica Andrés Bello, por medio de la presente hago constar que he revisado con fines de validación los instrumentos de recolección de datos (Cuestionario y Guía de Observación), del Trabajo de Grado de Maestría: **Modelo de evaluación de la calidad del producto de software para medicina prepagada,** de la estudiante Belkis Elena Carrasco Sosa, cédula de identidad N° 19.453.820, quien opta por el título de Magíster en Sistemas de Información.

Sin otro particular al cual hacer referencia, se despide cordialmente,

_____

Gloria Aponte

C.I. 4964695

ANEXO I

Constancia de Evaluación del Instrumento de Recolección de Datos Profesor III

Universidad Católica Andrés Bello.
Vicerrectorado Académico.
Estudios de Postgrado
Área de Ingeniería.
Postgrado en Sistemas de Información.

Constancia de Evaluación del Instrumento de Recolección de Datos

Yo, María Esther Remedios, de Profesión Licenciada en Computación. Magíster en Sistemas de Información, ejerciendo actualmente como evaluador, en la Universidad Católica Andrés Bello, por medio de la presente hago constar que he revisado con fines de validación los instrumentos de recolección de datos (Cuestionario y Guía de Observación), del Trabajo de Grado de Maestría: **Modelo de evaluación de la calidad del producto de software para medicina prepagada,** de la estudiante Belkis Elena Carrasco Sosa, cédula de identidad N° 19.453.820, quien opta por el título de Magíster en Sistemas de Información.

Sin otro particular al cual hacer referencia, se despide cordialmente,

_____

María Esther Remedios

ANEXO J

Matriz de evaluación de la calidad del modelo propuesto

Universidad Católica Andrés Bello.
Vicerrectorado Académico.
Estudios de Postgrado
Área de Ingeniería.
Postgrado en Sistemas de Información.

MATRIZ DE EVALUACIÓN DE LA CALIDAD DEL PRODUCTO DE SOFTWARE DEL MODELO DISEÑADO

A continuación se presenta la matriz que permitirá evaluar la calidad de producto de software acorde al modelo planteado. La misma se divide en dos secciones, la primera corresponde al encabezado que contiene información referente al proceso de evaluación (empresa, persona encargada realizar la el proceso e información de la evaluación), y la segunda contiene la matriz de evaluación, constituida por las características, subcaracterísticas, métricas y el espacio respectivo para completar los valores del software a evaluar.

Finalmente, se debe tomar en cuenta que para el modelo planteado, la distribución de porcentaje de calidad de cada característica es igual e independiente, es decir, ninguna característica posee mayor prioridad, esto se traduce de la siguiente forma:

Característica	Subcaracterística	Representación en porcentaje para el modelo
Arquitectura	Incrementalidad	3,030%
	Abstracción	3,030%
	Modularidad	3,030%
Control	Revisiones técnicas	3,030%
	Seguimiento funcional	3,030%
Depuración	Funcional	3,030%
	Unitaria	3,030%
	Rendimiento	3,030%
	Seguridad	3,030%

Característica	Subcaracterística	Representación en porcentaje para el modelo
Formalidad	Documentación mantenimiento	3,030%
	Documentación técnica	3,030%
	Resultados de pruebas	3,030%
Eficacia	Estabilidad	3,030%
	Adecuación	3,030%
	Conformidad	3,030%
Integridad	Confidencialidad	3,030%
	Disponibilidad	3,030%
	Auditoría de acceso	3,030%
Reusabilidad	Generalidad	3,030%
	Independencia de módulos	3,030%
Comprensibilidad	Atractividad	3,030%
	Aprendizaje	3,030%
	Operatividad	3,030%
	Validaciones para errores de usuarios	3,030%
	Documentación de usuario	3,030%
Eficiencia	Uso de recursos	3,030%
	Tiempo de respuesta	3,030%
Mantenibilidad	Anticipación al cambio	3,030%
	Facilidad de cambio	3,030%
	Estabilidad	3,030%
	Facilidad de diagnóstico	3,030%
Portabilidad	Adaptabilidad	3,030%
	Interoperabilidad	3,030%

Una vez se lleve a cabo el proceso de evaluación, en el análisis de los resultados obtenidos se deberá calcular para cada métrica el porcentaje de cumplimiento, con el objetivo de representar el desempeño de la calidad del producto evaluado de manera cuantitativa.

La matriz diseñada es la siguiente:

DATOS DE LA EMPRESA	
Empresa:	
Rif:	
DATOS DEL EVALUADOR	
Nombre del Evaluador:	
Cargo en la empresa:	
DATOS DE LA EVALUACIÓN	
Fecha de la Evaluación:	
Nombre de la Aplicación:	

MATRIZ DE EVALUACIÓN DE LA CALIDAD

Característica	Subcaracterística	Métrica	Valor correspondiente a la Aplicación		
Arquitectura	Incrementalidad	¿Cumple el desarrollo del producto de software con el atributo de Incrementalidad?	**SI**	**Medianamente (50% -> 1,515)**	**NO**
	Abstracción	¿Los objetos del producto de software solo poseen los atributos esenciales y asociados a sus funciones?	**SI**	**Medianamente (50% -> 1,515)**	**NO**
	Modularidad	**M = 1 - A/T** M -> Modularidad T -> Total de módulos que constituyen el sistema A -> Módulos creados que cumplen con la característica de alta cohesión y bajo acoplamiento.	M= 1 – 0 <= M <= 1		
Control	Revisiones técnicas	¿Cada entregable del proyecto de desarrollo de software cumple con los estándares técnicos y funcionales?	**SI**	**Medianamente (50% -> 1,515)**	**NO**
	Seguimiento funcional	¿Cada módulo fue desarrollo en el tiempo estipulado, cumpliendo las funciones para las cuales fue diseñado?	**SI**	**Medianamente (50% -> 1,515)**	**NO**
Depuración	Funcional	¿El software cumple con los requisitos funcionales que satisface las necesidades del usuario?	**SI**	**Medianamente (50% -> 1,515)**	**NO**
	Unitaria	¿Cada módulo que constituye el sistema cumple con los requisitos funcionales descritos en su diseño?	**SI**	**Medianamente (50% -> 1,515)**	**NO**
	Rendimiento	¿Cada módulo que constituye el sistema	**SI**	**Medianamente (50% -> 1,515)**	**NO**

Característica	Subcaracterística	Métrica	Valor correspondiente a la Aplicación		
		cumple con los requisitos funcionales descritos en su diseño?			
	Seguridad	¿Las vulnerabilidades y riesgos a los cuales se encuentra expuesto el sistema fueron evaluados y mitigados en su mayoría?	**SI**	**Medianamente (50% -> 1,515)**	**NO**
Formalidad	Documentación mantenimiento	¿Se lleva cabo una documentación estructurada con el histórico de cambios preventivos o correctivos del sistema y sus implicaciones e impacto en el diseño original?	**SI**	**Medianamente (50% -> 1,515)**	**NO**
	Documentación técnica	¿Se lleva cabo una documentación estructurada con la descripción técnica y funcional del software?	**SI**	**Medianamente (50% -> 1,515)**	**NO**
	Resultados de pruebas	¿Una vez concluido el proceso de pruebas al software se elaboró un informe con los resultados obtenidos?	**SI**	**Medianamente (50% -> 1,515)**	**NO**
Eficacia	Estabilidad	**E = 1 - MF/MT** E -> Estabilidad MT -> Total de módulos que constituyen el sistema MF -> Cantidad de módulos que presentan fallas	E = 1 – 0 <= E <= 1		
	Adecuación	**A = 1 - FC/FR** A -> Adecuación FR -> Total de funcionalidades requeridas FC -> Funcionalidad cumplidas en la construcción del sistema	A = 1 – 0 <= A <= 1		
	Conformidad	¿Se cumplieron en su totalidad los acuerdos concretados y el cliente se encuentra satisfecho con los resultados obtenidos?	**SI**	**Medianamente (50% -> 1,515)**	**NO**
Integridad	Confidencialidad	¿Sólo el personal autorizado puede tener acceso a la información que procesa el sistema?	**SI**	**Medianamente (50% -> 1,515)**	**NO**

Característica	Subcaracterística	Métrica	Valor correspondiente a la Aplicación		
	Disponibilidad	**D = 1 - TFS/TR** D -> Disponibilidad TFS -> Tiempo fuera de servicio TR -> Tiempo total de respuesta del sistema mientras se estudia este atributo	D = 1 – 0 <= D <= 1		
	Auditoría de acceso	¿El sistema desarrollado gestiona y registra los accesos de los usuarios a la información que este procesa, así como cualquier cambio o modificación que se lleve a cabo?	**SI**	**Medianamente (50% -> 1,515)**	**NO**
Reusabilidad	Generalidad	¿El sistema aportar las soluciones que abarquen la mayor cantidad de casos posibles?	**SI**	**Medianamente (50% -> 1,515)**	**NO**
	Independencia de módulos	**IM = 1 - MD/TM** IM -> Independencia de módulos MD -> Módulos de los cuales depende su funcionamiento TM -> Total de módulos del sistema.	IM = 1 – 0 <= IM <= 1		
Comprensibilidad	Atractividad	¿El diseño de las vistas del sistema es agradable e intuitivo para el usuario final?	**SI**	**Medianamente (50% -> 1,515)**	**NO**
	Aprendizaje	**AP = 1 - HE/HT** AP -> Aprendizaje HE -> Horas adicionales de entrenamiento para el manejo del sistema HT -> Horas total de entrenamiento para el manejo del sistema	AP = 1 – 0 <= AP <= 1		
	Operatividad	¿El sistema lleva a cabo las tareas para las cuales fue creado bajo el control del usuario final?	**SI**	**Medianamente (50% -> 1,515)**	**NO**
	Validaciones para errores de usuarios	**VE = 1 - MIV/MTV** VE -> Validaciones para errores de usuarios MIV -> Cantidad de módulos que validan la información sensible a errores de los usuarios	VE = 1 -		

Característica	Subcaracterística	Métrica	Valor correspondiente a la Aplicación		
		MTV -> Total de módulos que manejan información sensible a errores de los usuarios	0 <= VE <= 1		
	Documentación de usuario	¿Se redactó un manual de usuario que facilite el proceso de enseñanza al usuario y permita consultarlo ante cualquier duda de su funcionamiento?	**SI**	**Medianamente (50% -> 1,515)**	**NO**
Eficiencia	Uso de recursos	¿El sistema hace uso únicamente de los recursos necesarios para procesar las peticiones?	**SI**	**Medianamente (50% -> 1,515)**	**NO**
	Tiempo de respuesta	**ATR = 1 - TE/TR** ATR -> Tiempo de respuesta TR -> Tiempo real de respuesta TE -> Tiempo estimado de respuesta	ATR = 1 – 0 <= ATR <= 1		
Mantenibilidad	Anticipación al cambio	¿El sistema está capacitado para anticipar cambios bien sea preventivos o correctivos?	**SI**	**Medianamente (50% -> 1,515)**	**NO**
	Facilidad de cambio	**FC = 1 - HE/HIP** FC -> Facilidad de cambio HE -> Horas estipuladas para el cambio HIP -> Horas invertidas por el programador	FC = 1 – 0 <= FC <= 1		
	Estabilidad	**E = 1 - CE/CT** E -> Estabilidad CE -> Cambios exitosos CT -> Cambios totales planificados	E = 1 – 0 <= E <= 1		
	Facilidad de diagnóstico	¿El sistema está capacitado para anticipar cambios bien sea preventivos o correctivos?	**SI**	**Medianamente (50% -> 1,515)**	**NO**
Portabilidad	Adaptabilidad	¿El sistema está preparado para funcionar de manera correcta en un ambiente (hardware y software) distinto al cual se ha instalado inicialmente?	**SI**	**Medianamente (50% -> 1,515)**	**NO**
	Interoperabilidad	¿El sistema está capacitado para coexistir	**SI**	**Medianamente (50% -> 1,515)**	**NO**

Característica	Subcaracterística	Métrica	Valor correspondiente a la Aplicación		
		con otros sistemas e intercambiar recursos sin que esto afecte su funcionamiento?			

Observaciones generales

_____
_____
_____
_____
_____
_____

ANEXO K

Documento de certificación de la calidad del modelo propuesto

Universidad Católica Andrés Bello.
Vicerrectorado Académico.
Estudios de Postgrado
Área de Ingeniería.
Postgrado en Sistemas de Información.

CERTIFICACIÓN DE LA CALIDAD DEL PRODUCTO DE SOFTWARE

Por medio del presente documento, yo, Belkis Elena Carrasco Sosa, hago constar en calidad de evaluadora, que el día X de XXX de XXXX se realizó la evaluación de la calidad de XXXXXXXX de la empresa XXXXXX (Rif: X-XXXXXXXX), obteniendo XXX% de cumplimiento de las métricas del modelo.

Se elaboró un análisis del proceso de evaluación de la calidad, donde se indicaron recomendaciones para mitigar las fallas o inconvenientes que dispone actualmente el software.

Finalmente no se estableció un tiempo límite de corrección de errores y próxima evaluación. (En caso de haber llegado a un acuerdo indicarlo a continuación)

Atentamente

_____

Belkis Elena Carrasco Sosa

C.I. 19.453.820