



REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD MONTEÁVILA
COMITÉ DE ESTUDIOS DE POSTGRADO



**ESPECIALIZACIÓN EN PLANIFICACIÓN,
DESARROLLO Y GESTIÓN DE PROYECTOS**

Plan de Implementación del Proyecto de Aplicación de un Modelo de Gestión de
Calidad para el Desarrollo de Software en Grupo CIS de Venezuela

**Trabajo Especial de Grado, para optar al Título de Especialista en
Planificación, Desarrollo y Gestión de Proyectos, presentado por:**
Díaz Ferreira, Alejandro Javier, C.I.: 81.988.234

Asesorado por:
González V., Manuel

Caracas, septiembre de 2017

**REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD MONTEÁVILA
COMITÉ DE ESTUDIOS DE POSTGRADO**

**ESPECIALIZACIÓN EN PLANIFICACIÓN, DESARROLLO Y GESTIÓN DE
PROYECTOS**

Plan de Implementación del Proyecto de Aplicación de un Modelo de Gestión de
Calidad para el Desarrollo de Software en Grupo CIS de Venezuela

**Trabajo Especial de Grado, para optar al Título de Especialista en
Planificación, Desarrollo y Gestión de Proyectos, presentado por:**

Díaz Ferreira, Alejandro Javier, C.I.: 81.988.234

Asesorado por:
González V., Manuel

Caracas, septiembre de 2017

Señores:

Universidad Monteávila

Comité de Estudios de Postgrado

Especialización en Planificación, Desarrollo y Gestión de Proyectos

Atención: Profesora Geraldine Cardozo

Referencia: **Aprobación de Asesoría**

Por medio de la presente le informo que hemos revisado el borrador final del Proyecto de Trabajo Especial de Grado de (los) Ciudadano (s): **Díaz Ferreira, Alejandro Javier**, titular de la Cédula de Identidad N° **81988234**; cuyo título tentativo es: “Plan de Implementación del Proyecto de Aplicación de un Modelo de Gestión de Calidad para el Desarrollo de Software en Grupo CIS de Venezuela”, la cual cumple con los requisitos vigentes de esta casa de estudio para asignarles jurado y su respectiva presentación.

A los 18 días del mes de Julio del 2017

Guillén Guedez, Ana Julia
Gonzalez Manuel

**ESPECIALIZACIÓN EN PLANIFICACIÓN,
DESARROLLO Y GESTIÓN DE PROYECTOS**

TRABAJO ESPECIAL DE GRADO

Plan de Implementación del Proyecto de Aplicación de un Modelo de Gestión de Calidad para el Desarrollo de Software en Grupo CIS de Venezuela

Autor: Díaz Ferreira, Alejandro Javier

Asesor: Gonzalez, Manuel

Año: 2017

RESUMEN

Grupo CIS de Venezuela es una empresa integradora de sistemas que provee al mercado local y regional servicios de información, a raíz de la diversificación de la cartera de productos surgió el departamento de Fábrica de Software. Como consecuencia de varios incidentes registrados con los productos entregados se planteó la necesidad de mejorar la calidad de los desarrollos acometidos y se originó la propuesta de implementar un modelo de gestión de calidad para el desarrollo de aplicaciones. El objetivo del presente trabajo consistió en proponer un Plan de Implementación de un Modelo de Gestión de la Calidad para el Desarrollo de Software y para lograrlo se realizó un análisis situacional, la visualización de modelos de calidad, la conceptualización del modelo a implementar y la definición del Plan de Ejecución del Proyecto de la Implementación del modelo establecido. La continuidad en el tiempo de las pequeñas y medianas empresas se sustenta en gran manera en la evolución hacia procesos más eficientes y menos costosos, la satisfacción del cliente y en general a generar productos de calidad, estas razones generaron la necesidad de adoptar un Modelo de Calidad. La investigación se desarrolló bajo la modalidad de Gestión de Proyectos, para el departamento de Fábrica de Software de Grupo CIS de Venezuela. Se utilizaron los conceptos de Proyecto, Plan de Ejecución del Proyecto, Calidad, Calidad del Producto de software, gestión de la Calidad e ISO/IEC 25000 SQUARE. El tipo de investigación utilizada fue de tipo aplicada y el diseño fue un Estudio de Caso. Para el análisis situacional se utilizó un cuestionario, a través del cual se obtuvieron los diferentes problemas que presentaban los productos elaborados y se realizó un análisis documental del proceso actual, posteriormente se evaluaron los modelos de calidad, se definió el modelo a implementar y se estableció el Plan de Ejecución de Proyecto para implementar el modelo propuesto. En el siguiente capítulo se realizó el análisis de resultado estableciendo la afectación del modelo de calidad en cada una de las etapas del ciclo de vida de los proyectos. Posteriormente se establecieron las lecciones aprendidas en cuanto a tiempo y costos del proyecto ejecutado y por último se plantearon las conclusiones y recomendaciones generadas durante la investigación realizada.

Línea de Trabajo: Generación de Proyectos y Factibilidad General

Palabras clave: [Desarrollo de Software, Gestión de Calidad, Implementación de Proyectos, Aseguramiento de Calidad, Control de Calidad]

Nomenclatura UNESCO: (33) Ciencias Tecnológicas, (3304) Tecnología de los Ordenadores, (330499) Desarrollo de Software.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS	x
ÍNDICE DE TABLAS	xi
ÍNDICE DE GRÁFICOS	xiii
LISTA DE ACRÓNIMOS Y SIGLAS.....	xiv
INTRODUCCIÓN	1
CAPITULO I. PLANTEAMIENTO DE LA INVESTIGACIÓN.....	3
Planteamiento de la Investigación	3
Interrogante y Sistemización de la Investigación	5
Objetivos de la Investigación	5
Objetivo General.....	5
Objetivos Específicos	6
Justificación e Importancia.....	6
Alcance y Delimitación de la Investigación	7
CAPITULO II. MARCO TEÓRICO.....	8
Antecedentes.....	8
Bases Teóricas	12
Plan de Ejecución del Proyecto	13
Plan de Gestión del Alcance	14
Estructura Desagregada de Trabajo	14
Plan de Gestión del Cronograma	14
Plan de Gestión de Costos.....	15
Plan de Gestión de Calidad.....	15

Plan de Gestión de Recursos Humanos	16
Plan de Gestión de Comunicaciones	16
Plan de Gestión de Riesgos.....	17
Plan de Gestión de las Adquisiciones	18
Plan de Gestión de los Interesados.....	18
Calidad	18
Calidad de Software	19
Gestión de la Calidad	19
ISO/IEC 25010 SQUARE (System and Software Quality Requirements and Evaluation).....	20
ISO/IEC 2500n - División de Gestión de Calidad	22
ISO/IEC 2501n - División de Modelo de Calidad.....	23
ISO/IEC 2502n - División de Medición de Calidad	24
ISO/IEC 2503n - División de Requisitos de Calidad.....	25
ISO/IEC 2504n - División de Evaluación de Calidad	25
Calidad Externa e Interna ISO/IEC 25010.....	26
Calidad en Uso ISO/IEC 25010.....	27
Definiciones ISO/IEC 25010.....	28
ISO/IEC 9000-3	29
CAPITULO III. MARCO METODOLÓGICO	31
Tipo de Investigación	31
Diseño de la Investigación	31
Unidad de Análisis	31
Técnicas y Herramientas de Recolección e Interpretación	32
Fases de la Investigación	33

Operacionalización de las Variables	36
Aspectos Éticos de la Investigación.....	37
CAPITULO IV. MARCO REFERENCIAL	38
Reseña Histórica.....	38
Directiva.....	38
Misión	38
Visión.....	39
Valores.....	39
Organigrama de la Unidad Estratégica de Negocios	40
Descripción de la Unidad Estratégica de Negocios	40
CAPITULO V. ANÁLISIS DE LA INVESTIGACIÓN DE CAMPO	42
Adecuación Funcional	43
Seguridad	44
Compatibilidad.....	45
Fiabilidad	46
Usabilidad.....	47
Eficiencia de Desempeño.....	48
Mantenibilidad	49
Portabilidad.....	50
Relación de las Preguntas con Respecto a los Objetivos Propuestos	51
CAPITULO VI. DESARROLLO DE LA INVESTIGACIÓN	53
Objetivo 1 - Realizar un Análisis Situacional, en cuanto a las Actividades de Calidad e Identificar Mejoras en las Diferentes Etapas del Ciclo de Vida de los Proyectos.....	53
Objetivo 2 - Evaluar los Modelos de Calidad de Software	59

Objetivo 3 - Definir un Modelo de Gestión de Calidad para el Ciclo de Vida de los Proyectos de Desarrollo de Software.....	66
Planificación de la Realización del Producto	66
Procesos Relacionados con el Cliente	68
Diseño y Desarrollo	68
Definición de Nivel de Importancia	69
Aplicación de Niveles de Importancia a Características de Calidad	70
Aplicación de Niveles de Importancia a Subcaracterísticas y Atributos.....	72
Métricas de Calidad Interna y Externa.....	76
Métricas de Calidad en Uso.....	106
Modelo de Indicadores y Métricas.....	118
Objetivo 4 - Diseñar el Proyecto de Ejecución del Plan de Implementación del Modelo de Gestión de Calidad Propuesto	120
Plan de Gestión del Alcance	120
Objetivos del Proyecto	120
Descripción del Alcance del Proyecto	120
Estructura Desagregada de Trabajo	120
Plan de Gestión del Cronograma	123
Plan de Gestión de Costos.....	124
Plan de Gestión de Calidad.....	125
Plan de Gestión de Recursos Humanos	126
Plan de Gestión de Comunicaciones	127
Plan de Gestión del Riesgos	128
Plan de Gestión del Adquisiciones	129
CAPITULO VII. ANÁLISIS DE RESULTADO.....	130
CAPITULO VIII. LECCIONES APRENDIDAS	133

Diccionario de la Estructura Desagregada de Trabajo.....	133
Factibilidad de la Investigación	134
CAPITULO IX. CONCLUSIONES Y RECOMENDACIONES	137
Conclusiones	137
Recomendaciones	138

ÍNDICE DE FIGURAS

Figura 1. Diagrama Causa-Efecto de Calidad del Producto de Software.....	4
Figura 2. Modelo de calidad del producto Fuente ISO/IEC 25010 (2010).....	27
Figura 3. Modelo de calidad en uso del producto software	28
Figura 4. Estructura Desagregada de Trabajo	35
Figura 5. Organigrama de la Unidad de Negocio.....	40
Figura 6. Diagrama Ishikawa – Análisis Situacional.....	53
Figura 7. Diagrama Ishikawa – Errores por Área de Conocimiento	57
Figura 8. Open Innovation Canvas – Implementación de un Modelo de Calidad..	63
Figura 9. EDS - PLANGECA.....	120
Figura 10. Organigrama Inicial A.....	122
Figura 11. Cronograma del Plan de Implementación	123
Figura 12. Camino Crítico del Cronograma.....	123
Figura 13. Canvas Modelo de Negocio: Plan de Implementación del Modelo de Gestión de la Calidad.....	131
Figura 14. Cronograma Original del Proyecto	134

ÍNDICE DE TABLAS

Tabla 1. Divisiones ISO/IEC 25000 SQUARE	21
Tabla 2. Cláusulas ISO 9000-3	30
Tabla 3. Procedimiento por Objetivos	34
Tabla 4. Operacionalización de Variables	36
Tabla 5. Preguntas – Adecuación Funcional.....	43
Tabla 6. Preguntas – Seguridad.....	44
Tabla 7. Preguntas – Compatibilidad	45
Tabla 8. Preguntas – Fiabilidad	46
Tabla 8. Preguntas – Usabilidad.....	47
Tabla 10. Preguntas – Eficiencia de Desempeño	48
Tabla 11. Preguntas – Mantenibilidad	49
Tabla 12. Preguntas – Portabilidad.....	50
Tabla 13. Proyectos Desarrollados en 2016	55
Tabla 14. Grupo Focal – Evaluación de Modelos de Calidad	60
Tabla 15. Comparación de Modelos de Calidad	61
Tabla 16. Niveles de Importancia.....	70
Tabla 17. Ejemplo de Niveles de Importancia para Características de Calidad Interna.....	71
Tabla 18. Ejemplo de Niveles de Importancia para Características de Calidad Externa.....	71
Tabla 19. Ejemplo de Niveles de Importancia para Características de Calidad en Uso.....	72
Tabla 20. Ejemplo de Niveles de Importancia para Subcaracterísticas y Atributos de Calidad Interna.....	72

Tabla 21. Ejemplo de Niveles de Importancia para Subcaracterísticas y Atributos de Calidad Externa.....	74
Tabla 22. Ejemplo de Niveles de Importancia para Subcaracterísticas y Atributos de Calidad en Uso.....	75
Tabla 23. Métricas de Calidad Interna/Externa para Adecuación Funcional	76
Tabla 24. Métricas de Calidad Interna/Externa para Fiabilidad.....	78
Tabla 25. Métricas de Calidad Interna/Externa para Eficiencia en el Desempeño	82
Tabla 26. Métricas de Calidad Interna/Externa para Facilidad de uso	86
Tabla 27. Métricas de Calidad Interna/Externa para Seguridad.....	91
Tabla 28. Métricas de Calidad Interna/Externa para Compatibilidad	94
Tabla 29. Métricas de Calidad Interna/Externa para Mantenibilidad	96
Tabla 30. Métricas de Calidad Interna/Externa para Portabilidad	101
Tabla 31. Métricas de Calidad en Uso para Efectividad.....	106
Tabla 32. Métricas de Calidad en Uso para Eficiencia	107
Tabla 33. Métricas de Calidad en Uso para Satisfacción.....	110
Tabla 34. Métricas de Calidad en Uso para Libertad de riesgo.....	112
Tabla 35. Métricas de Calidad en Uso para Cobertura de contexto.....	117
Tabla 36. Ejemplo de ponderación en porcentajes para la calidad interna	118
Tabla 37. Niveles de puntuación final para la calidad interna, externa y en uso .	119
Tabla 38. Niveles de puntuación final para la calidad interna, externa y en uso .	119
Tabla 39. Matriz de Actividades de Calidad	125
Tabla 40. Presupuesto de Recursos	126
Tabla 41. Matriz de Comunicaciones	127
Tabla 42. Matriz de Riesgos.....	128
Tabla 43. Diccionario de la EDT.....	133
Tabla 44. Presupuesto de Recursos	136

ÍNDICE DE GRÁFICOS

Gráfico 1– Respuestas Adecuación Funcional.....	44
Gráfico 2– Respuestas Seguridad	45
Gráfico 3 – Respuestas Compatibilidad	46
Gráfico 4 – Respuestas Fiabilidad	47
Gráfico 5 – Respuestas Usabilidad	48
Gráfico 6 – Respuestas Eficiencia de Desempeño	49
Gráfico 7 – Respuestas Mantenibilidad.....	50
Gráfico 8 – Respuestas Portabilidad.....	51
Gráfico 9 – Errores por Tipo.....	56
Gráfico 10 – Perfil de Progreso Financiero	124
Gráfico 11 – Histograma de Recursos	126
Gráfico 12 – Curva S de Esfuerzo.....	135
Gráfico 13 – Curva S de Recursos del Proyecto.....	136

LISTA DE ACRÓNIMOS Y SIGLAS

CMMI	Capability Maturity Model Integration
EDT	Estructura Desagregada de Trabajo
FELABAN	Federación Latinoamericana de Bancos
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Standardization Organization
PLANGECA	Plan de Gestión de la Calidad
PMI	Project Management Institute
PMBOK	Project Management Body of Knowledge
PRINCE	PRojects IN Controlled Environments
PYMES	Pequeñas y Medianas Empresas
SQUARE	System and Software Quality Requirements and Evaluation
WBS	Work Breakdown Structure
UNE	Una Norma Española
PLANGECA	Plan de Gestión de la Calidad
TEG	Trabajo Especial de Grado

INTRODUCCIÓN

A raíz de la situación país que afecta a las pequeñas y medianas empresas y la necesidad de mejorar los procesos productivos que sustentan la operación de Grupo CIS de Venezuela, surge la necesidad de implantar mecanismos de gestión de calidad para los proyectos desarrollados en el departamento de Fábrica de Software. Estos mecanismos generarán las acciones necesarias para supervisar las tareas de los recursos involucrados y el avance del proyecto, a través de métodos, técnicas e instrumentos diversos, mediante los cuales se conocerá si el proyecto está generando los productos con la calidad esperada.

Como resultado de la implementación se espera obtener mejoras en el proceso de Gestión de la Calidad, entre las cuales se destacan las siguientes:

- Minimizar las fallas en los productos entregados para pruebas y certificación.
- Minimizar las diferencias entre las funcionalidades desarrolladas y las esperadas por el cliente.
- Reducir las horas de desarrollo adicionales a las presupuestadas inicialmente, para corregir las fallas detectadas en el entregable.
- Reducir la insatisfacción por parte del cliente al no tener un producto de calidad desde la primera entrega.
- Obtener la aceptación final del producto de forma temprana.

El presente trabajo de investigación tiene como objetivo principal el desarrollo de un modelo, que contenga las acciones necesarias para establecer los procedimientos de gestión de la calidad en el proceso de desarrollo de software, realizado en el departamento de Fábrica de Software de Grupo CIS de Venezuela.

En el capítulo I “Planteamiento de la Investigación” se encuentran todos los elementos correspondientes a la situación actual, justificación e importancia de

promover una mejora y los objetivos, alcance y limitaciones de la investigación. En el capítulo II “Marco Teórico” se mencionan las referencias teóricas que soportan la investigación, los antecedentes que sirven como referencia y el marco legal en el cual se desarrollará la investigación. En el capítulo III “Marco Metodológico” se establecen los elementos metodológicos que rigen el desarrollo propuesto en el trabajo de investigación. En el capítulo IV “Marco Referencial” se realiza la descripción de la organización sobre la cual se realiza la investigación. En el capítulo V se realizó el Análisis de la Investigación de Campo, en donde se presentan los resultados de la herramienta aplicada. En el capítulo VI se presenta el desarrollo de los cuatro objetivos de la investigación, que incluyen la propuesta de Plan de Implementación. En el capítulo VII se desarrolla el análisis de resultados. En el capítulo VIII se detallan las lecciones aprendidas y en el capítulo IX las conclusiones y recomendaciones del proyecto realizado. Por último, se detallan las referencias bibliográficas utilizadas.

CAPITULO I. PLANTEAMIENTO DE LA INVESTIGACIÓN

1. Planteamiento de la Investigación

Grupo CIS de Venezuela es una empresa nacida en el año 2005, la cual se conforma con la finalidad de proveer al mercado nacional y regional, sistemas informáticos orientados a instituciones financieras. A lo largo de estos años y evolucionando con el mercado nacional y la situación económica actual, ha diversificado la cartera de productos de la empresa. Dentro de los productos y servicios actuales destaca la Fábrica de Software, departamento que atiende no solo a las instituciones financieras si no a empresas en general, aumentando de este modo el número de clientes potenciales.

El departamento de Fábrica de Software cuenta con un grupo de consultores de servicios de información que trabaja en el desarrollo de aplicaciones, a partir de un conjunto de requerimientos solicitados por el cliente. El esquema utilizado para los proyectos desarrollados es el Ciclo de Vida de Software en Cascada. Si bien se estableció un modelo para el desarrollo de programas desde el comienzo del departamento, a la fecha no existe ninguna metodología o práctica formal para la gestión de la calidad de los requerimientos trabajados.

A raíz de diversos incidentes ocurridos con los productos entregables generados por el departamento de fábrica de software, la gerencia realizó una recopilación y evidenció un grupo de problemas que se detallan a continuación: Pruebas incompletas que generaron errores, re-trabajo y costos adicionales, Cronogramas con estimaciones erradas y sin validación, problemas con la gestión de los controles de cambio al evidenciarse falta de validación por los interesados, Control de Calidad y Aseguramiento de la Calidad, áreas en las cuales se desarrollan prácticas informales sin el debido procedimiento y documentación.

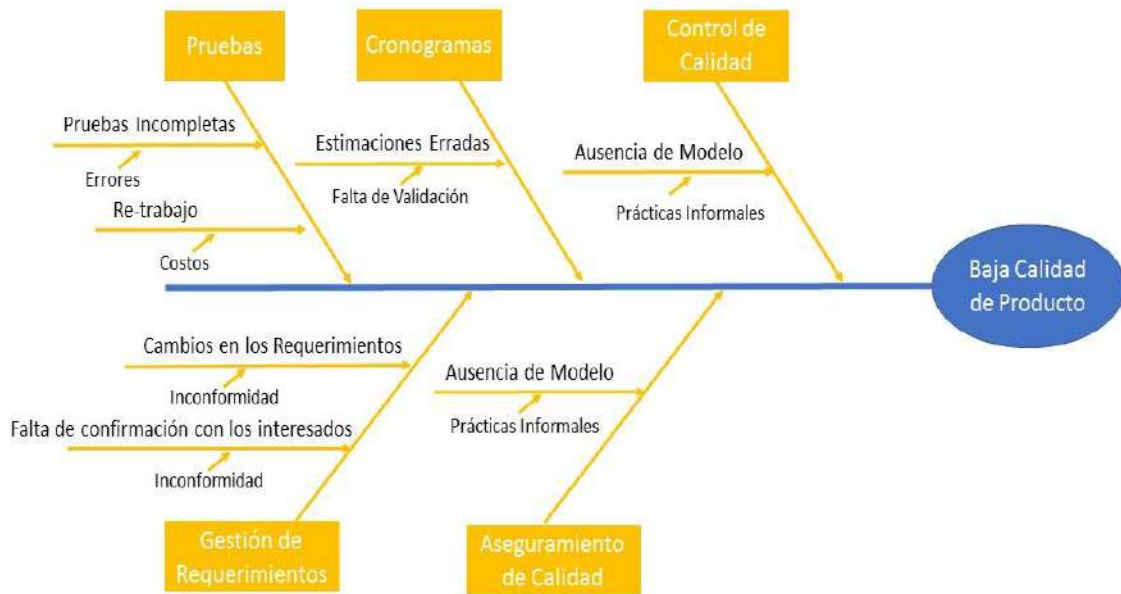


Figura 1. Diagrama Causa-Efecto de Calidad del Producto de Software

Fuente: Adaptado de Ishikawa (1943).

Como resultado de esquema de trabajo actual se han generado problemas en los entregables, que incluyen:

- Fallas en el producto entregado para pruebas y certificación.
- Diferencias entre las funcionalidades desarrolladas y las esperadas por el cliente.
- Horas de desarrollo adicionales a las presupuestadas inicialmente, para corregir las fallas detectadas en el entregable.
- Insatisfacción por parte del cliente al no tener un producto de calidad desde la primera entrega.
- Dificultades para obtener la aceptación final del producto.

Tomando en consideración lo antes expuesto y la necesidad de adaptarse a un mercado con problemas, que requiere productos de calidad a bajo costo y en

tiempos cortos, la empresa plantea la necesidad de mejorar la calidad de los proyectos emprendidos por el departamento de Fábrica de Software.

2. Interrogante y Sistemización de la Investigación

Interrogante de la Investigación

¿Cómo se puede mejorar la calidad del producto de los proyectos emprendidos por el departamento de Fábrica de Software de la organización en estudio?

Sistemización de la Investigación

¿Cómo se desarrollan las actividades de gestión de la calidad en los procesos realizados en el departamento de Fábrica de Software de Grupo CIS de Venezuela?

¿Cuáles son las etapas del Ciclo de Vida de proyectos que requieren mejoras?

¿Cuáles serían los procesos, actividades y herramientas necesarias para realizar la gestión de la calidad del Desarrollo de Software?

¿Cuál es el modelo de calidad de Desarrollo de Software que se adapta a la Fábrica de Software de Grupo CIS de Venezuela?

¿Cuáles serían las fases para implantar el modelo propuesto?

3. Objetivos de la Investigación

a. Objetivo General

Proponer un Plan de Implementación del Proyecto de Aplicación de un Modelo de Gestión de Calidad para el Desarrollo de Software en Grupo CIS de Venezuela, que permita mejorar la calidad del producto entregado al cliente.

b. Objetivos Específicos

- Realizar un análisis situacional de las actividades de gestión de la calidad en los proyectos, en la organización en estudio.
- Evaluar los Modelos de Calidad de Software.
- Definir un Modelo de Gestión de Calidad para el Ciclo de Vida de los proyectos de desarrollo de software.
- Diseñar el Plan de Ejecución del Proyecto de la Implementación del Modelo de Gestión de Calidad propuesto.

4. Justificación e Importancia

A nivel del posgrado de Planificación, Desarrollo y Gestión de Proyectos, se plantea un proyecto para desarrollar un plan de implementación enmarcado en las buenas prácticas de proyecto.

Grupo CIS de Venezuela es una empresa perteneciente al sector denominado PYMES, y como tal, afectada de forma importante por la situación económica que vive el país. La continuidad en el tiempo de este tipo de empresas se basa de gran manera, en la capacidad de evolucionar hacia procesos más eficientes y menos costosos.

El departamento de Fábrica de Software surgió como parte de este proceso evolutivo y a partir de la necesidad de diversificar la cartera de productos. Desde sus comienzos, se basó en un esquema tradicional de Ciclo de Vida, más nunca se formalizaron procedimientos de gestión de calidad de proyectos. Esto generó diferentes consecuencias, siendo las más importantes las horas necesarias para corregir errores detectados luego de entregar al cliente, y la insatisfacción por parte del cliente al no tener un producto de calidad desde la primera entrega.

Por lo antes expuesto se planteó la necesidad de implementar procedimientos de gestión de calidad que permitan mitigar las causas de los problemas descritos.

5. Alcance y Delimitación de la Investigación

La investigación se desarrolló bajo la modalidad de Gestión de Proyectos, especialmente como proyecto implementación de los grupos de procesos de gestión de calidad.

Abarcó únicamente a la empresa Grupo CIS de Venezuela y en particular al departamento de Fábrica de Software. El período de desarrollo de la investigación fué el comprendido entre el 01 de septiembre de 2016 y el 30 de junio de 2017.

CAPITULO II. MARCO TEÓRICO

2.1. Antecedentes

1.- Delgado (2014). ELABORACIÓN DE UN MODELO DE GESTIÓN DE RIESGOS BASADO EN LA METODOLOGÍA DE GESTIÓN DE PROYECTOS PARA EL PROYECTO DE IMPLEMENTACIÓN DE SOFTWARE FINANCIERO, FIDUCIARIO Y DE INVERSIÓN EN LA EMPRESA GESTOR INC.

Resumen: El presente estudio determina un modelo de gestión de riesgos en base a la metodología y mejores prácticas utilizadas por PMBOK y CMMI entre otras; se desarrolla en base de las 5 fases principales de gestión de riesgos: preparación, identificación, análisis cuantitativo, respuesta, Seguimiento y Control; establecido el modelo , se aplica en el proyecto "Provisión e Implementación de un Sistema Integrado de Gestión Bancaria para el BIESS. (BIESS-IMP-002- Implementación de Gestor Inversiones en el Banco del IESS)". Como resultado de la aplicación se determina los riesgos de cada uno de los procesos, se identifica el nivel de exposición de...

Aporte al Proyecto: El aporte de este trabajo será de tipo referencial, al tratar uno de los planes del Plan de Ejecución, como lo es el riesgo, bajo el estándar del PMI.

Palabras Clave: Modelo de Gestión de Riesgos, Proceso de Implementación, PMI, PMBOK.

2.- Baldeón (2015). MÉTODO PARA LA EVALUACIÓN DE CALIDAD DE SOFTWARE BASADO EN ISO/IEC 25000.

Resumen: La gestión de la calidad del producto es un factor crítico de éxito en los proyectos de desarrollo de software. En este sentido, el presente trabajo de investigación propone un método basado en ISO/IEC 25000 (2005) para evaluar la calidad de los entregables de un proyecto. Este método proporciona los lineamientos necesarios para contribuir al

incremento de la calidad del producto final y asegurar el cumplimiento de los requisitos del usuario. En esta investigación, se revisa la literatura relacionada al estudio, se muestra el análisis de la norma ISO/IEC 25000 (2005) y sus principales divisiones; luego se detalla el método propuesto para evaluar la calidad del producto software considerando los entregables desde la etapa de análisis.

Aporte al Proyecto: El aporte de este trabajo será de tipo referencial, al tratar sobre la norma ISO/IEC 25000.

Palabras Clave: Calidad de software, Evaluación de calidad, ISO/IEC 25000.

3.- Barzola y Henríquez (2015). DISEÑO DE UNA METODOLOGÍA DE CERTIFICACIÓN DE PRODUCTOS DE SOFTWARE ORIENTADO AL SECTOR PÚBLICO.

Resumen: En la actualidad el cliente es mucho más exhaustivo y crítico con los productos o servicios que adquiere u obtiene, entre ellos los servicios que ofrecen las soluciones de tecnología de información, y muchas veces son la cadena que ayuda a que determinada empresa tenga una mala reputación por la insatisfacción recibida.

Se realizó la investigación del porqué de los continuos errores en las aplicaciones que eran ofrecidas al usuario; se determinó que no se habían establecido procesos de desarrollo de software adecuados para poder asegurar la calidad del software que es implementado, y algunas de las razones por las que no lo habían efectuado eran por desconocimiento de estándares de calidad.

Aporte al Proyecto: El aporte de este trabajo será de tipo referencial, al tratar sobre una metodología de certificación de la calidad en procesos de Desarrollo de Software.

Palabras Clave: Estándares de Calidad, Modelos de Proceso, Metodologías de Desarrollo de Software.

4.- Basso (2014). PROPUESTA DE MÉTRICAS PARA PROYECTOS DE EXPLOTACIÓN DE INFORMACIÓN.

Resumen: Los Proyectos de Explotación de Información requieren de un proceso de planificación para estimar el esfuerzo, el tiempo y medir diferentes aspectos del producto para garantizar la calidad del mismo. Los procesos de desarrollo tradicionales y las métricas usuales de la Ingeniería de Software y la Ingeniería del Conocimiento no son adecuados para estos proyectos, ya que las etapas de desarrollo y los parámetros utilizados son de naturaleza y características diferentes. En ese contexto, se ha definido un Modelo de Proceso de Desarrollo para Proyectos Explotación de Información. No obstante, existe la necesidad de abordar métricas específicas aplicables a este proceso.

Aporte al Proyecto: El aporte de este trabajo será de tipo referencial, ya que aporta información acerca de las posibilidades de medición.

Palabras Clave: Métricas en el Desarrollo de Proyectos, Métricas en Ingeniería de Software.

5.- Meza (2014). MARCO DE TRABAJO PARA LA GESTIÓN DE LA CALIDAD EN PROYECTOS DE DESARROLLO DE SOFTWARE BASADO EN PMBOK Y CMMI DEV.

Resumen: Actualmente los proyectos informáticos muestran una alta tasa de fracaso. Existen diversos autores como Piorun, The Standish Group 2012, entre otros que han realizado investigaciones en torno a este tema y la mayoría coincide en que aproximadamente un 39% de los proyectos informáticos iniciados, finalizan de forma exitosa mientras el restante 61% son finalizados con algún tipo de problema o simplemente no finalizan. Entendiéndose por exitosa a que el proyecto finaliza a tiempo y dentro del presupuesto cumpliendo los requerimientos solicitados por el cliente; a este concepto se ha denominado la triple restricción. El presente trabajo, propone el diseño y construcción de un Marco de Trabajo que ayude a los proyectos de Desarrollo de Software.

Aporte al Proyecto: El aporte de este trabajo será de tipo referencial, al estar enmarcado en las prácticas del PMI.

Palabras Clave: Calidad del Proceso de Desarrollo de Software, CMMI, PMI, Gestión de la Calidad.

6.- Balseca, (2014). EVALUACIÓN DE CALIDAD DE PRODUCTOS DE SOFTWARE EN EMPRESAS DE DESARROLLO DE SOFTWARE APLICANDO LA NORMA ISO/IEC 25000.

Resumen: El objetivo del presente trabajo es realizar la evaluación de calidad de productos de software en empresas de desarrollo de software aplicando la norma ISO/IEC 25000, evaluando de esta manera el sistema LogiNotificador de la empresa Logiciel Cía. Ltda., siguiendo el proceso de evaluación y personalizando el modelo de calidad que tiene la norma. El presente proyecto tiene 4 capítulos que se describen a continuación: El primer capítulo trata acerca del análisis de las características y productos que ofrecen las empresas desarrolladoras de software, estableciendo a continuación el estudio de la norma de calidad ISO/IEC 25000 con un desglose sinóptico de la misma.

Aporte al Proyecto: El aporte de este trabajo será de tipo referencial, al aplicar el modelo de calidad propuesto por la norma ISO/IEC 25000 a la evaluación de un producto de software.

Palabras Clave: Estándares de Calidad, ISO/IEC 25000, Indicadores, Métricas.

7.- Ramos (2016). DISEÑO DE UN MODELO DE EVALUACIÓN DE LA CALIDAD DE PRODUCTOS DE SOFTWARE, BASADO EN MÉTRICAS EXTERNAS Y USABILIDAD APLICADO A UN CASO DE ESTUDIO.

Resumen: El desarrollo de este proyecto tiene la finalidad de convertirse en una herramienta de medición de calidad de los productos de software permitiendo determinar el nivel de calidad que poseen los sistemas adquiridos por las empresas. El propuesto Modelo de Evaluación de la

Calidad de Productos de Software surge de la investigación comparativa de características externas y en uso. El presente documento está conformado por cuatro capítulos, detallados a continuación: Primer Capítulo, presenta una comparación de los modelos más conocidos de Calidad, los cuales sirven de base para establecer el modelo. Además, define la relación entre las normas ISO/IEC 9126, ISO/IEC 14598 e ISO/IEC 25000 y el Modelo McCall.

Aporte al Proyecto: El aporte de este trabajo será de tipo referencial, al comparar modelos de calidad y plantear un modelo de calidad para la evaluación de un producto de software.

Palabras Clave: Modelos de Calidad de Producto de Software, ISO/IEC 25000, ISO/IEC 9126, ISO/IEC 14598, Indicadores, Métricas.

2.2. Bases Teóricas

2.2.1. Proyecto

Según el Project Management Institute (PMI, 2013), un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único. La naturaleza temporal de los proyectos implica que un tienen un principio y un final definidos. El final se alcanza cuando se logran los objetivos establecidos, cuando se termina porque sus objetivos no se cumplirán o no pueden ser cumplidos, o cuando ya no existe la necesidad que dio origen al proyecto. Asimismo, se puede poner fin a un proyecto si el cliente (cliente, patrocinador o líder) desea terminarlo.

La Office of Government Commerce (Prince 2, 2009) describe un proyecto como “un entorno de gestión que se crea con el propósito de entregar uno o más productos de negocio de acuerdo con el caso de negocio especificado”. Este entorno de gestión es temporal, por ejemplo, para la vida del proyecto,

y difiere de “gestión lineal” la cual es más duradera y generalmente se ocupa de la actividad base.

2.2.2. Plan de Ejecución del Proyecto

Según el PMI (2013), el Plan para la Dirección del Proyecto es el documento que describe el modo en que el proyecto será ejecutado, monitoreado y controlado. Integra y consolida todos los planes y líneas base secundarios de los procesos de planificación.

Las líneas base del proyecto incluyen, entre otras:

- Línea base del alcance
- Línea base del cronograma
- Línea base de costos

Los planes secundarios incluyen, entre otros:

- Plan de gestión del alcance
- Plan de gestión del cronograma
- Plan de gestión de costos
- Plan de gestión de calidad
- Plan de gestión de recursos humanos
- Plan de gestión de las comunicaciones
- Plan de gestión de riesgos
- Plan de gestión de las adquisiciones
- Plan de gestión de los interesados

Chamoun (2002), indica que el Plan de Proyecto funciona como un plan de vuelo o navegación contra el cual compararemos el avance, para evaluar periódicamente el desempeño del proyecto.

2.2.2.1. Plan de Gestión del Alcance

El PMI (2013) establece que el Plan para la Gestión del alcance es un componente del plan para la dirección del proyecto o programa que describe cómo será definido, desarrollado, monitoreado, controlado y verificado el alcance. Los componentes de un plan para la gestión del alcance incluyen:

- El proceso para elaborar un enunciado detallado del alcance del proyecto.
- El proceso que permite la creación de la EDT/WBS a partir del enunciado detallado del alcance del proyecto.
- El proceso que establece cómo se mantendrá y aprobará la EDT/WBS.
- El proceso que especifica cómo se obtendrá la aceptación formal de los entregables del proyecto que se hayan completado.
- El proceso para controlar cómo se procesarán las solicitudes de cambio relativas al enunciado del alcance detallado del proyecto.

2.2.2.1.1. Estructura Desagregada de Trabajo

Lledó (2011) define la Estructura desagregada de trabajo (EDT), como una especie de organigrama jerárquico del proyecto donde se subdivide el mismo en menores componentes. El nivel más bajo de cada división se denomina “paquete de trabajo” y el primer nivel jerárquico de la EDT podría ser el ciclo de vida del proyecto.

2.2.2.2. Plan de Gestión del Cronograma

El PMI (2013) lo define como un componente del plan para la dirección del proyecto que establece los criterios y las actividades a llevar a cabo para desarrollar, seguir y controlar el cronograma. Según las necesidades del

proyecto, el plan de gestión del cronograma puede ser formal o informal, de carácter detallado o más general, e incluye los umbrales de control apropiados.

2.2.2.3. Plan de Gestión de Costos

El Plan de Gestión de Costos describe la forma en que se planificarán, estructurarán y controlarán los costos del proyecto. Los procesos de gestión de costos, así como sus herramientas y técnicas asociadas, se documentan en el Plan de Gestión de Costos, PMI (2013).

El Plan de Gestión de Costos puede contener: Unidades de medida, nivel de precisión, nivel de exactitud, enlaces con los procedimientos de la organización, umbrales de control, reglas para la medición de desempeño, formatos de los informes, descripción de los procesos y detalles adicionales, tales como la descripción de la selección estratégica del financiamiento, o procedimiento empleado para tener en cuenta las fluctuaciones en los tipos de cambio y/o procedimiento para el registro de los costos del proyecto.

2.2.2.4. Plan de Gestión de Calidad

El plan de gestión de calidad es un componente del plan para la dirección del proyecto que describe cómo se implementarán las políticas de calidad de una organización. Describe la manera en que el equipo participante, planea cumplir los requisitos de calidad establecidos para el proyecto, PMI (2013).

Chamoun (2002), indica que los objetivos de la administración de la calidad de un proyecto son: asegurar que satisfaga las necesidades para las cuales inició, identificar los estándares de calidad relevantes y determinar cómo satisfacer dichos estándares.

2.2.2.5. Plan de Gestión de Recursos Humanos

Según el PMI (2013), el plan de gestión de recursos humanos, el cual forma parte del plan para la dirección del proyecto, proporciona una guía sobre el modo en que se deberían definir, adquirir, dirigir y finalmente liberar los recursos humanos del proyecto. Incluye, entre otros, los siguientes elementos:

- Roles y Responsabilidades.
- Organigramas del Proyecto.
- Plan para la Administración del personal.

2.2.2.6. Plan de Gestión de Comunicaciones

El PMI (2013), establece que el plan de gestión de las comunicaciones es un componente del plan para la dirección del proyecto y describe la forma en que se planificarán, estructurarán, monitorearán y controlarán las comunicaciones del proyecto.

Entre los componentes del plan se pueden mencionar los siguientes:

- Los requisitos de comunicación de los interesados.
- La información que debe ser comunicada, incluidos el idioma, el formato, el contenido y el nivel de detalle.
- El motivo de la distribución de dicha información.
- El plazo y la frecuencia para la distribución de la información requerida y para la recepción de la confirmación o respuesta, si corresponde.
- La persona responsable de comunicar la información.
- La persona responsable de autorizar la divulgación de información confidencial.
- La persona o los grupos que recibirán la información.

- Los métodos o tecnologías utilizados para transmitir la información, tales como memorandos, correo electrónico y/o comunicados de prensa.
- Los recursos asignados para las actividades de comunicación, incluidos el tiempo y el presupuesto.
- El proceso de escalamiento, con identificación de los plazos y la cadena de mando (nombres) para el escalamiento de aquellos incidentes que no puedan resolverse a un nivel inferior.
- El método para actualizar y refinar el plan de gestión de las comunicaciones a medida que el proyecto avanza y se desarrolla.
- Un glosario de la terminología común.
- Diagramas de flujo de la información que circula dentro del proyecto, los flujos de trabajo con la posible secuencia de autorizaciones, la lista de informes y los planes de reuniones, etc.
- Restricciones en materia de comunicación, generalmente derivadas de una legislación o normativa específica, de la tecnología, de las políticas de la organización, etc.

2.2.2.7. Plan de Gestión de Riesgos

El plan de gestión de riesgos es un componente del plan para la dirección del proyecto y describe el modo en que se estructurarán y se llevarán a cabo las actividades de gestión de riesgos. El plan de gestión de riesgos incluye lo siguiente: Metodología, Roles y Responsabilidades, Presupuesto, Calendario, Categorías de Riesgo, Definiciones de la probabilidad e impacto de los riesgos, Matriz de probabilidad e impacto, Revisión de las tolerancias de los interesados, Formatos de los informes y Seguimiento, PMI (2013).

2.2.2.8. Plan de gestión de las Adquisiciones

El plan de gestión de las adquisiciones es un componente del plan para la dirección del proyecto que describe cómo un equipo de proyecto adquirirá bienes y servicios desde fuera de la organización ejecutante. Describe cómo se gestionarán los procesos de adquisición, desde la elaboración de los documentos de la adquisición hasta el cierre del contrato, PMI (2013).

2.2.2.9. Plan de gestión de los Interesados

El plan de gestión de los interesados es un componente del plan para la dirección del proyecto e identifica las estrategias de gestión necesarias para involucrar a los interesados de manera eficaz. El plan de gestión de los interesados puede ser formal o informal, muy detallado o formulado de manera general, en función de las necesidades del proyecto, PMI (2013).

2.2.3. Calidad

El concepto de calidad varía dependiendo del autor u organización citada. Según la norma ISO 9001 (2008) de la Organización Internacional para la Estandarización, calidad es la totalidad de características de un producto o servicio que le otorgan la habilidad para satisfacer las necesidades explícitas o implícitas del cliente.

Según Crosby (1979), la calidad es cero defectos haciendo las cosas correctamente desde el principio. Feigenbaum (1961) la define como la satisfacción de las expectativas del cliente y Deming (1989) como la satisfacción del cliente.

2.2.4. Calidad del Producto de Software

Pressman (1998) define la calidad del software como: “la concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente”.

Según la IEEE (1990), la calidad del software es el grado con el que un sistema, componente o proceso cumple con los requisitos especificados y las necesidades del cliente o usuario.

La norma ISO/IEC 25040 (2010) lo define como un conjunto de características y la relación entre las mismas, que conforman la base para especificar requerimientos de calidad y evaluar la calidad.

2.2.5. Gestión de la Calidad

Según la norma ISO 9001:2008, la gestión de la calidad consiste en la realización de actividades coordinadas que permiten dirigir y controlar una organización en lo relativo a la calidad.

Berlinches (2002), la define como un conjunto de actividades encaminadas a planificar, organizar y controlar la función de la calidad de la empresa”. (p.5).

El PMI (2013) establece que los procesos de gestión de la calidad incluyen lo siguiente:

- **Planificación de la calidad:** consiste en el proceso de identificar los requerimientos y/o estándares de calidad para el proyecto y sus entregables y documentar como el proyecto cumplirá con los requerimientos de calidad.

- **Aseguramiento de la calidad:** consiste en el proceso de auditar los requerimientos de calidad y los resultados de las medidas de control de calidad, para asegurar que están siendo utilizados estándares de calidad y definiciones operacionales apropiadas.
- **Control de la calidad:** el proceso de monitorear y recopilar los resultados obtenidos de las actividades de calidad para evaluar el desempeño y recomendar los cambios necesarios.

2.2.6. ISO/IEC 25000 SQUARE (System and Software Quality Requirements and Evaluation).

La norma ISO/IEC 25000 constituye una serie de normas basadas en la ISO 9126 y en la ISO 14598 (Evaluación del Software), y su objetivo principal es guiar el desarrollo de los productos de software con la especificación y evaluación de requisitos de calidad. Establece criterios para la especificación de requisitos de calidad de productos software, sus métricas y su evaluación.

SQuaRE está formada por las divisiones siguientes:

Tabla 1. Divisiones ISO/IEC 25000 SQUARE

NORMA	CARACTERÍSTICA
ISO/IEC 2500n	División de gestión de calidad. Los estándares que forman esta división definen todos los modelos comunes, términos y referencias a los que se alude en las demás divisiones de SQuaRE.
ISO/IEC 2501n	División del modelo de calidad. El estándar que conforma esta división presenta un modelo de calidad detallado, incluyendo características para la calidad interna, externa y en uso.
ISO/IEC 2502n	División de mediciones de calidad. Los estándares pertenecientes a esta división incluyen un modelo de referencia de calidad del producto software, definiciones matemáticas de las métricas de calidad y una guía práctica para su aplicación. Presenta aplicaciones de métricas para la calidad de software interna, externa y en uso.
ISO/IEC 2503n	División de requisitos de calidad. Los estándares que forman parte de esta división ayudan a especificar los requisitos de calidad. Estos requisitos pueden ser usados en el proceso de especificación de requisitos de calidad para un producto software que va a ser desarrollado ó como entrada para un proceso de evaluación. El proceso de definición de requisitos se guía por el establecido en la norma ISO/IEC 15288 (ISO, 2003).
ISO/IEC 2504n	División de evaluación de la calidad. Estos estándares proporcionan requisitos, recomendaciones y guías para la evaluación de un producto software, tanto si la llevan a cabo evaluadores, como clientes o desarrolladores.
ISO/IEC 25050–25099	Estándares de extensión SQuaRE. Incluyen requisitos para la calidad de productos de software “Off-The-Self” y para el formato común de la industria (CIF) para informes de usabilidad.

Fuente: ISO/IEC 25000 SQUARE (2014)

Al igual que la norma ISO/IEC 9126, este estándar define tres vistas diferenciadas en el estudio de la calidad de un producto:

Vista interna: Ésta vista se ocupa de las propiedades del software como: el tamaño, la complejidad o la conformidad con las normas de orientación a objetos. Puede utilizarse desde las primeras fases del desarrollo, permitiendo detectar deficiencias en el software en edades muy tempranas del ciclo de vida del software.

Vista externa: Vista que analiza el comportamiento del software en producción y estudia sus atributos, por ejemplo: el rendimiento de un software en una máquina determinada, el uso de memoria de un programa o el tiempo de funcionamiento entre fallos. Necesita que el producto software este completo y se utilizará por tanto en el pase a producción del producto, siendo muy dependiente de la máquina donde se ejecute.

Vista en uso: Mide la productividad y efectividad del usuario final al utilizar el software. También estudia el producto software finalizado será dependiente del usuario y estará condicionada a los factores personales del mismo.

El cumplimiento de los requisitos de calidad interna se comprobará en un proceso de verificación que permitirá medirlo, el cumplimiento de los requisitos de calidad externa se comprobará en un proceso de validación que permitirá medirlo y finalmente la satisfacción de las necesidades de la calidad del producto se comprobarán en un proceso de evaluación que permitirá medir la calidad en uso.

2.2.6.1. ISO/IEC 2500n - División de Gestión de Calidad

Las normas que forman este apartado definen todos los modelos, términos y definiciones comunes referenciados por todas las otras normas de la familia 25000. Actualmente, esta división se encuentra formada por:

- ISO/IEC 25000 (2005) – Guía para SQuaRE: contiene el modelo de la arquitectura de SQuaRE, la terminología de la familia, un resumen de las partes, los usuarios previstos y las partes asociadas, así como los modelos de referencia.
- ISO/IEC 25001 – Planeamiento y Gestión. Establece los requisitos y orientaciones para gestionar la evaluación y especificación de los requisitos del producto software. Proporciona requisitos y recomendaciones para una organización responsable de la implementación y administración de las especificaciones de los requisitos de calidad de sistemas y productos de software, y las actividades de evaluación a través de la provisión de tecnología, herramientas, experiencias y habilidades de gestión.

El papel del grupo de evaluación incluye motivar a los empleados y capacitarlos para la especificación de los requisitos y las actividades de evaluación, preparación de documentos apropiados, la identificación o el desarrollo de los métodos requeridos, y responder a las preguntas sobre las tecnologías relevantes.

La gestión de la tecnología está relacionada con la especificación de requerimientos de calidad de sistemas y software, así como de los procesos, mediciones y herramientas de evaluación. Esto incluye la gestión del desarrollo, la adquisición, la normalización, el control, la transferencia y la retroalimentación de las experiencias de especificación de requisitos y tecnología de evaluación dentro de la organización.

2.2.6.2. ISO/IEC 2501n - División de Modelo de Calidad

Las normas de este apartado presentan modelos de calidad detallados incluyendo características para calidad interna, externa y en uso del producto software. Actualmente, esta división se encuentra formada por:

- ISO/IEC 25010 (2010) –Modelos de Calidad de Software y Sistemas. Describe el modelo de calidad para el producto software y para la calidad en uso. Esta norma presenta las características y sub-características de calidad frente a las cuales se debe evaluar el producto software.
- ISO/IEC 25012 – Modelo de Calidad de Datos. Define un modelo general para la calidad de los datos, aplicable a aquellos que se encuentran almacenados de manera estructurada y forman parte de un Sistema de Información.

2.2.6.3. ISO/IEC 2502n - División de Medición de Calidad

Estas normas incluyen un modelo de referencia de la medición de la calidad del producto, definiciones de medidas de calidad (interna, externa y en uso) y guías prácticas para su aplicación. Actualmente esta división se encuentra formada por:

- ISO/IEC 25020 - Guía y Modelo de Referencia para la medición. Presenta una explicación introductoria y un modelo de referencia común a los elementos de medición de la calidad. También proporciona una guía para que los usuarios seleccionen o desarrollen y apliquen medidas propuestas por normas ISO.
- ISO/IEC 25021 (2011) - Elementos de Medida de Calidad. Define y especifica un conjunto recomendado de métricas base y derivadas que puedan ser usadas a lo largo de todo el ciclo de vida del desarrollo software. Este conjunto de métricas se utilizará como entrada en el proceso de medida de la calidad interna, externa y en el uso, también especifica la forma de crear nuevas métricas de calidad en el modelo.

- ISO/IEC 25022 - Medición de Calidad en Uso. Define específicamente las métricas para realizar la medición de la calidad en uso del producto.
- ISO/IEC 25023 (2013) - Medición de la Calidad del Producto Software y Sistemas. Define específicamente las métricas para realizar la medición de la calidad de productos y sistemas software.
- ISO/IEC 25024 - Medición de la Calidad de Datos: define específicamente las métricas para realizar la medición de la calidad de datos.

2.2.6.4. ISO/IEC 2503n - División de Requisitos de Calidad

Las normas que forman este apartado ayudan a especificar requisitos de calidad que pueden ser utilizados como entrada del proceso de evaluación. Para ello, este apartado se compone de:

- ISO/IEC 25030 (2007) – Requisitos de Calidad: provee de un conjunto de recomendaciones para realizar la especificación de los requisitos de calidad del producto software.

2.2.6.5. ISO/IEC 2504n - División de Evaluación de Calidad

Este apartado incluye normas que proporcionan requisitos, recomendaciones y guías para llevar a cabo el proceso de evaluación del producto software. Se encuentra formada por:

- ISO/IEC 25040 (2010) - Guía y Modelo de Referencia de la Evaluación. Propone un modelo de referencia general para la evaluación, que considera las entradas al proceso de evaluación, las restricciones y los recursos necesarios para obtener las correspondientes salidas.

- ISO/IEC 25041 (2012) - Guía de Evaluación para Desarrolladores, Adquirentes y Evaluadores Independientes. Describe los requisitos y recomendaciones para la implementación práctica de la evaluación del producto software desde el punto de vista de los desarrolladores, de los adquirentes y de los evaluadores independientes.
- ISO/IEC 25042 - Módulos de Evaluación. Define lo que la Norma considera un módulo de evaluación y la documentación, estructura y contenido que se debe utilizar a la hora de definir uno de estos módulos.
- ISO/IEC 25045 – Módulo de Evaluación para Recuperabilidad: Define un módulo para la evaluación de la sub-característica “Recuperabilidad” (Recoverability).

2.2.6.6. Calidad Externa e Interna ISO/IEC 25010

La norma ISO/IEC 25010 (2010) define la calidad interna como: “la totalidad de las características del producto software desde una perspectiva interna. La calidad interna es medida y evaluada en base a los requerimientos de calidad interna. Los detalles de la calidad del producto software pueden ser mejorados durante la implementación, revisión y prueba del código software, pero la naturaleza fundamental de la calidad del producto software representada por la calidad interna permanece sin cambios a menos que sea rediseñado”.

La calidad externa es definida como: “la totalidad de las características del producto software desde una perspectiva externa. Es la calidad cuando el software es ejecutado, la cual es típicamente medida y evaluada mientras se prueba en un ambiente simulado con datos simulados y usando métricas

externas. Durante las pruebas, muchas fallas serán descubiertas y eliminadas. Sin embargo, algunas fallas todavía pueden permanecer después de las pruebas. Como es difícil corregir la arquitectura de software u otros aspectos fundamentales del diseño del software, el diseño fundamental permanece sin cambios a través de las pruebas”.

La Figura 2 representa el modelo de calidad interna o externa, que muestra un conjunto de 8 características: funcionalidad, eficiencia en el rendimiento, compatibilidad, usabilidad, confiabilidad, seguridad, facilidad de mantenimiento y portabilidad.



Figura 2. Modelo de calidad del producto Fuente ISO/IEC 25010 (2010)

2.2.6.7. Calidad en Uso ISO/IEC 25010

La norma ISO/IEC 25010 (2010) define la calidad en uso como la perspectiva del usuario de la calidad del producto software cuando este es usado en un ambiente específico y un contexto de uso específico. Esta mide la extensión para la cual los usuarios pueden conseguir sus metas en un ambiente particular, en vez de medir las propiedades del software en sí mismo.

Un usuario es cualquier tipo de posible operador y cuyos requerimientos pueden ser diferentes; por ejemplo, un operador del software tiene un requerimiento diferente que un responsable del mantenimiento del software.

En la Figura 3 se presenta el modelo de calidad en uso que muestra un conjunto de cinco características: efectividad, eficiencia, satisfacción, libertad de riesgo, cobertura de contexto.



Figura 3. Modelo de calidad en uso del producto software

Fuente ISO/IEC 25010 (2010)

2.2.6.8. Definiciones ISO/IEC 25010

SQuaRE hace un uso intensivo de ciertos conceptos que deben entenderse para poder comprender el estándar en su conjunto. El documento oficial contiene la descripción de 64 conceptos que incluye definiciones desde “acquirer” hasta “verification”. A continuación, se detallan los más utilizados:

Atributo: Propiedad inherente de una entidad que puede distinguirse cuantitativa o cualitativamente ya sea manual o automáticamente.

Calidad interna: Capacidad de un conjunto estático de atributos para satisfacer las necesidades declaradas e implícitas de un producto software bajo ciertas condiciones especificadas.

Calidad externa: Capacidad de un producto software para desarrollar el comportamiento de un sistema de forma que satisfaga las necesidades declaradas e implícitas de un sistema utilizado bajo ciertas condiciones especificadas.

Calidad en uso: Grado en que un producto satisface objetivos con efectividad, seguridad, satisfacción y productividad.

(Medida) primitiva: Medida, tanto base como derivada, utilizada para medir la calidad del software.

Medida base: Conjunto formado por la medida definida en términos de un atributo más el método para su cuantificación.

Medida derivada: Medida obtenida a partir dos o más medidas base.

Módulo de evaluación: Módulo tecnológico para la medida de características, sub-características y atributos de evaluación, incluyendo métodos y técnicas de evaluación, entradas a procesar, datos a recoger y medir, y herramientas y procedimientos de apoyo.

Validación: Confirmación por medio de pruebas objetivas de que se satisfacen los requisitos para un uso específico o para una aplicación.

Verificación: Confirmación por medio de pruebas objetivas de que se satisfacen los requisitos especificados.

2.2.7. ISO/IEC 9000-3.

La norma ISO/IEC 9000-3 ofrece el marco general sobre procesos de ingeniería de software en software de computador. Es una norma derivada de la norma ISO 9001 descrita como: “Guía para la aplicación de ISO 9001 para el desarrollo, implementación y mantenimiento de software”. Indica los aspectos que se deberían tener en cuenta independientemente de la tecnología usada, las metodologías de desarrollo, la secuencia de actividades y la estructura de la organización. Aunque no da una visión específica de cada aspecto, sí abarca procesos adicionales al desarrollo de software y sobre todo su relación con el sistema de gestión de calidad de la organización.

La Tabla 1 muestra los puntos considerados en la norma NTC-ISO/IEC 9000-3, para cada sección se tienen en cuenta los procedimientos orientados a la calidad:

Tabla 2. Cláusulas ISO 9000-3

NUMERO	CLAUSULA
4.1	Administración de la Responsabilidad
4.2	Sistema de Calidad
4.3	Auditorías Internas del Sistema de Calidad
4.4	Acción Correctora
5.1	General
5.2	Revisión del Contrato
5.3	Especificación de los requerimientos de la
5.4	Planificación del desarrollo
5.5	Planificación de la Calidad
5.6	Diseño e Implementación
5.7	Testeo y Validación
5.8	Aceptación
5.9	Generación, Entrega e Instalación
5.10	Mantenimiento
6.1	Administración de la Configuración
6.2	Documentos de Control
6.3	Calidad de los Archivos
6.4	Medidas
6.5	Reglas y Convenciones
6.6	Herramientas y Técnicas
6.7	Compra
6.8	Productos de software incluidos
6.9	Formación

Fuente: ISO/IEC 9000-3.

CAPITULO III. MARCO METODOLÓGICO

3.1. Tipo de Investigación

Tomando en cuenta la naturaleza y características del problema estudiado y en función de los objetivos propuestos, el tipo de investigación utilizado fue Investigación Aplicada.

Según Arias (2012), la investigación aplicada es aquella que produce un nuevo conocimiento, el cual puede ser aplicado inmediatamente en la solución de problemas prácticos. Sabino (1992) afirma que la investigación aplicada persigue fines más directos e inmediatos

3.2. Diseño de la Investigación

El diseño de la presente investigación fue un Estudio de Caso. Según Hernández, Fernández y Baptista (2006), los estudios de caso son: "estudios que al utilizar los procesos de investigación cuantitativa, cualitativa o mixta; analizan profundamente una unidad para responder al planteamiento del problema, probar hipótesis y desarrollar alguna teoría".

3.3. Unidad de Análisis

Las unidades de análisis fueron los proyectos de desarrollo de software realizados por el departamento de Fábrica de Software de Grupo CIS de Venezuela.

Los proyectos considerados para efectos del presente trabajo tienen las siguientes características:

- Las especificaciones funcionales son entregadas por el cliente.

- Se sigue una metodología de desarrollo SCRUM con iteraciones sucesivas.
- Existe un registro de los pases a los ambientes de certificación usuaria y producción.
- Existe un registro de los errores detectados en el ambiente de certificación.

3.4. Técnicas y Herramientas de Recolección e Interpretación

Para el presente caso de estudio se utilizaron las siguientes herramientas de recolección de datos:

- Análisis documental.
- Observación directa no estructurada y no participante.
- Cuestionarios.
- Entrevistas no estructuradas.

La información recolectada, se clasificó y analizó, y permitió evidenciar las debilidades del proceso actual de gestión de a calidad, sirviendo de insumo para la propuesta del Modelo de Gestión de Calidad.

La población a la que se le aplicaron los instrumentos de recolección de datos fue:

- Consultores de Sistemas de Información
- Líderes de Proyecto
- Gerente de Desarrollo

3.5. Fases de la Investigación

La investigación se desarrolló bajo la modalidad de Gestión de Proyectos y contempla las cinco (5) fases establecidas por la norma UNE-ISO 21500, a saber: Inicio, Planificación, Implementación, Control y Cierre. En la fase de Inicio se realizó el Planteamiento de la Investigación y el Marco Teórico, en la fase de Planificación se realizó el Marco Metodológico y el Marco Referencial, en la fase de ejecución se realizó el Análisis Situacional y el Desarrollo de la Propuesta, en la fase de Seguimiento y Control se realizó el Seguimiento de la Planificación, el Seguimiento de Costos y el Control de Calidad. Para la fase de cierre se realizó la Impresión del Documento, la Presentación de la Investigación, se obtuvo la Aprobación y se solicitaron las Solvencias correspondientes.

a. **Procedimientos por objetivos**

Tabla 3. Procedimiento por Objetivos

Objetivo Específico	Actividades	Técnicas / Herramientas	Entregables
Realizar un análisis situacional, en cuanto a las actividades de calidad de los proyectos e identificar mejoras en las diferentes etapas del Ciclo de Vida de los proyectos	<ul style="list-style-type: none"> - Diseñar un instrumento basado en ISO25000 con preguntas cerradas (Si – No). - Aplicar instrumento. - Identificar las mejoras en el proceso de calidad. 	<ul style="list-style-type: none"> - Análisis documental. - Cuestionarios 	Informe del Análisis Situacional.
Evaluar Modelos de Calidad de Software.	<ul style="list-style-type: none"> - Revisar el informe de análisis situacional. - Revisar alternativas de Modelos de Calidad. 	<ul style="list-style-type: none"> - Análisis documental. 	Cuadro Comparativo de Modelos de Calidad de Software
Definir un Modelo de Gestión de Calidad para el Ciclo de Vida de los proyectos de desarrollo de software.	<ul style="list-style-type: none"> - Seleccionar el modelo de Gestión de la Calidad 	<ul style="list-style-type: none"> - Análisis documental. 	Modelo de Gestión de la Calidad
Diseñar el Plan de Ejecución del Proyecto de la Implementación del Modelo de Gestión de Calidad propuesto.	<ul style="list-style-type: none"> - Preparar el Plan de Ejecución del Proyecto 	<ul style="list-style-type: none"> - Análisis documental. 	Propuesta de Plan de Ejecución del Proyecto.

b. Estructura Desagregada de Trabajo



Figura 4. Estructura Desagregada de Trabajo

3.6. Operacionalización de las Variables

Tabla 4. Operacionalización de Variables

Objetivo General	Objetivo Específico	Variables	Indicadores	Técnicas	Fuente
Proponer un Plan de Implementación del Proyecto de Aplicación de un Modelo de Gestión de Calidad para el Desarrollo de Software en Grupo CIS de Venezuela, que permita mejorar la calidad del producto entregado al cliente.	Realizar un análisis situacional, en cuanto a las actividades de calidad de los proyectos e identificar mejoras en las diferentes etapas del Ciclo de Vida de los proyectos.	Adecuación Funcional, Seguridad, Interoperabilidad, Fiabilidad, Usabilidad, Eficiencia de Desempeño, Mantenibilidad, Portabilidad	- Informe de análisis Situacional	-Cuestionario -Revisión Documental.	Personal de Grupo CIS de Venezuela, Bases Académicas, ISO/IEC 25000 SQUARE, PMI (2013).
	Evaluar Modelos de Calidad de Software.	Adecuación Funcional, Seguridad, Interoperabilidad, Fiabilidad, Usabilidad, Eficiencia de Desempeño, Mantenibilidad, Portabilidad	Cuadro Comparativo de Modelos de Calidad de Software	- Revisión Documental.	Bases Académicas, ISO/IEC 25000 SQUARE, ISO/IEC 9000-3
	Definir un Modelo de Gestión de Calidad para el Ciclo de Vida de los proyectos de desarrollo de software.	Adecuación Funcional, Seguridad, Interoperabilidad, Fiabilidad, Usabilidad, Eficiencia de Desempeño, Mantenibilidad, Portabilidad	- Modelo de Gestión de Calidad de Software.	-Revisión Documental	Cuadro Comparativo de Modelos de Calidad de Software
	Diseñar el Plan de Ejecución del Proyecto de la Implementación del Modelo de Gestión de Calidad propuesto.	Alcance, Tiempo, Costos, Riesgos, Involucrados, Calidad, Comunicaciones,	-Plan de Ejecución del Proyecto	-Revisión Documental	Buenas prácticas en la gestión de Proyectos, Bases Académicas.

3.7. Aspectos Éticos de la Investigación

a. Código de ética del Colegio de Ingenieros de Venezuela (1996)

La presente investigación se realizó de acuerdo con lo establecido por el Código de Ética Profesional del Colegio de Ingenieros de Venezuela, quien plantea un grupo de artículos orientados a:

- Virtudes
- Ilegalidad
- Conocimiento
- Seriedad
- Dispensa
- Remuneración
- Firma
- Obra
- Licitaciones
- Influencia
- Ventajas
- Reputación
- Intereses
- Justicia
- Al ambiente
- Extranjeros
- Autoría
- Secretos
- Experimentación y servicios no necesarios
- Publicidad indebida
- Actuación gremial

b. Código de ética del PMI (2006)

La presente investigación se realizó de acuerdo con lo establecido por el Código de Ética y Conducta Profesional establecido por el Project Management Institute. El código establece normas con respecto a Responsabilidad, Respeto, Equidad y Honestidad.

CAPITULO IV. MARCO REFERENCIAL

4.1. RESEÑA HISTÓRICA

(Grupo CIS de Venezuela, 2017), Grupo CIS de Venezuela, C.A., es una empresa venezolana, fundada en el año 2005 que nace como resultado de la asociación de un grupo de ingenieros, con sólidos conocimientos en el área de tecnologías de información y experiencia en la conducción de proyectos exitosos.

Grupo CIS de Venezuela, C.A. nace con la misión de proveer al mercado regional, sistemas informáticos orientados a instituciones financieras, desarrollados con la última tecnología, multiplataforma, de bajo coste y con actualización y servicio permanente.

Cinco años después de creada Grupo CIS de Venezuela, C.A. aumenta su portafolio, expande su área de operación, crea alianzas con empresas de tecnología y formaliza la internacionalización en Latinoamérica.

En el año 2014 realiza su primera participación como expositor en el CLAB Panamá 2014 organizado por FELABAN (Federación Latinoamericana de Bancos) y en 2015 participó en la edición realizada en Miami – USA.

4.2. DIRECTIVA

La directiva está conformada por un Presidente, un Vicepresidente y un Administrador.

4.3. MISIÓN

Somos un equipo de consultores tecnológicos con experiencia en el área de integración de sistemas informáticos, especializados para el sector bancario e industrias relacionadas. Comprometidos con la entrega de productos de excelente calidad y valor, que brinden soluciones integrales a necesidades específicas de nuestros clientes. Proporcionando ventajas competitivas y asegurando una excelente relación costo/beneficio.

4.4. VISIÓN

Ser reconocidos como una organización con excelentes prácticas de negocio y soluciones innovadoras y proveer al mercado regional Sistemas Informáticos orientados a áreas financieras, desarrollados con la última tecnología, multi-plataforma, de bajo costo, y con actualización y servicio permanente.

4.5. VALORES

Somos una empresa en la cual nuestra máxima es generar valor a nuestros clientes mediante la entrega de soluciones empresariales usando para ello nuestra vasta experiencia en el campo de Ingeniería de Software combinado con el conocimiento del negocio.

Somos reconocidos en el medio por la confiabilidad y oportunidad en el desarrollo de nuestros productos y servicios, así como en el compromiso que asumimos con nuestros clientes, construyendo relaciones perdurables, contando para ello con un equipo de trabajo con altas capacidades.

Aseguramos a nuestro potencial humano crecimiento personal y profesional, en medio de un agradable ambiente laboral, donde la confianza en el trabajo del otro es nuestro mayor activo y a los clientes les aseguramos satisfacción en el producto terminado.

Fomentamos los valores en el ámbito laboral, familiar y social.

4.6. ORGANIGRAMA DE LA UNIDAD ESTRATÉGICA DE NEGOCIOS

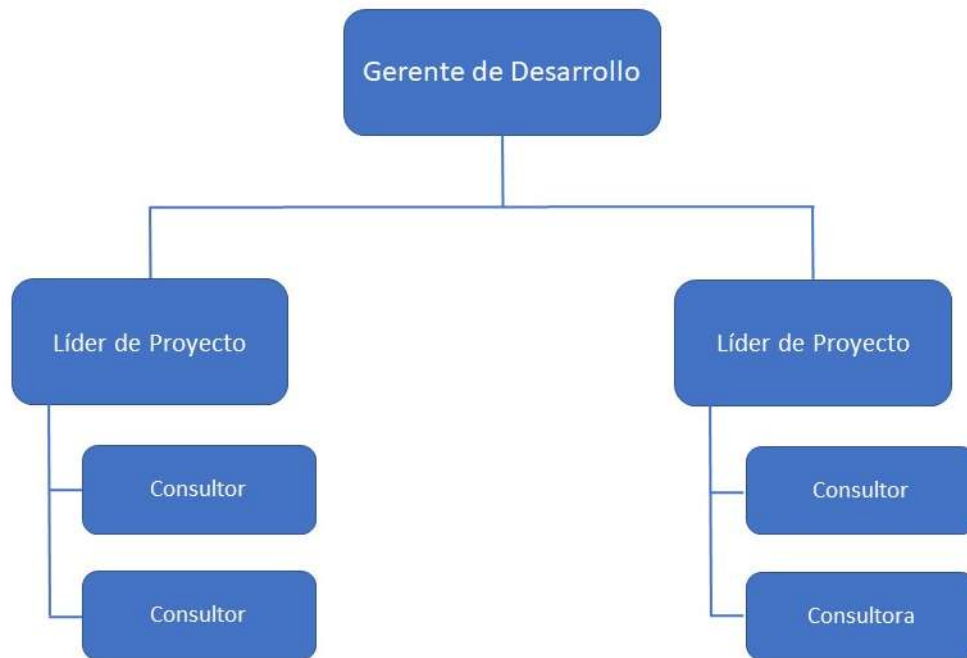


Figura 5. Organigrama de la Unidad de Negocio

Fuente: Grupo CIS de Venezuela (2017).

4.7. DESCRIPCIÓN DE LA UNIDAD ESTRATÉGICA DE NEGOCIOS

El departamento de “Fábrica de Software” tiene como objetivo principal el producir software basado en un conjunto de requerimientos solicitados por el cliente. Ejecutan proyectos de desarrollo de aplicaciones, así como la venta de horas/hombres de desarrollo.

Con Grupo CIS usted recibe servicios de Desarrollo de Software bajo las mejores prácticas, con niveles de calidad y productividad de primera clase que demostramos al cliente con cifras objetivas y medibles. En un ambiente de total transparencia entre cliente y proveedor, usted puede utilizar las métricas que acordemos para monitorear el rendimiento de los recursos que desplegamos en su empresa, sabiendo siempre lo que usted recibe por su inversión.

Al utilizar el servicio de Fábrica de Software, usted obtiene acceso a un grupo de Ingenieros excelentemente calificados los cuales pueden mejorar drásticamente su productividad en el desarrollo y mantenimiento de aplicaciones, disminuir sus costos de operación, y aumentar la velocidad de despliegue de nuevas herramientas informáticas que potencian su negocio.

Al contratar a Grupo CIS, usted puede estar tranquilo, sabiendo que trabajamos bajo las más estrictas prácticas de seguridad de la información cobijadas por el estándar ISO 27001 (2005), bajo el cual tenemos personal adiestrado. Por un lado, ISO 27001 protege su información de ataques informáticos y filtraciones. Por otro lado, provee mecanismos de contingencia para asegurar que los servicios desplegados por Grupo CIS continúen operando aun en caso de eventos mayores.

CAPITULO V. ANÁLISIS DE LA INVESTIGACIÓN DE CAMPO

A fin de realizar un análisis situacional, se procedió a aplicar un cuestionario a los empleados de la Fábrica de Software de Grupo CIS de Venezuela. Los empleados que participaron en la actividad cubren los perfiles de Consultor de Servicios de Información, Líder de Proyecto y Gerente de Desarrollo.

Bernal (2010), afirma que el cuestionario es un conjunto de preguntas diseñadas para generar los datos necesarios, con el propósito de alcanzar los objetivos del proyecto de investigación. Se trata de un plan formal para recabar información de la unidad de análisis objeto de estudio y centro del problema de investigación (p. 250). Para la elaboración del cuestionario se consideraron las características del modelo de calidad del producto de software que establece la norma ISO/IEC 25010 (2010); a saber, Adecuación Funcional, Eficiencia de Desempeño, Compatibilidad, Usabilidad, Fiabilidad, Seguridad, Mantenibilidad y Portabilidad. Las preguntas utilizadas fueron cerradas del tipo “si - no” y la hipótesis establecida es que existe un problema con la calidad del producto de software desarrollado por el departamento de Fábrica de Software de la empresa Grupo CIS de Venezuela.

La norma ISO/IEC 25010 (2010) define las características de calidad de la siguiente manera:

Adecuación Funcional: Representa la capacidad del producto software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones especificadas.

Eficiencia de Desempeño: Esta característica representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones.

Compatibilidad: Capacidad de dos o más sistemas o componentes para intercambiar información y/o llevar a cabo sus funciones requeridas cuando comparten el mismo entorno hardware o software.

Usabilidad: Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones.

Fiabilidad: Capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa bajo unas condiciones y periodo de tiempo determinados.

Seguridad: Capacidad de protección de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos.

Mantenibilidad: Esta característica representa la capacidad del producto software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas.

Portabilidad: Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno hardware, software, operacional o de utilización a otro.

Finalizada la fase de recolección de datos, se organizó la información y a continuación se presentan los resultados obtenidos:

4.1. Adecuación Funcional

Tabla 5. Preguntas – Adecuación Funcional

Pregunta	SI	NO
¿Se verifica que el producto desarrollado contenga todas las funcionalidades requeridas?	3	4
¿Se verifica que los valores emitidos por el producto desarrollado se correspondan con los esperados por el usuario final?	4	3
Total	7	7

Los resultados indican que los participantes entienden el concepto consultado y es considerado parcialmente durante el proceso de desarrollo del software y se realizan validaciones que garanticen la misma. Algunos de los participantes

manifestaron que se preocupaban por la funcionalidad asignada, más no conocían verificaciones sobre la funcionalidad global.

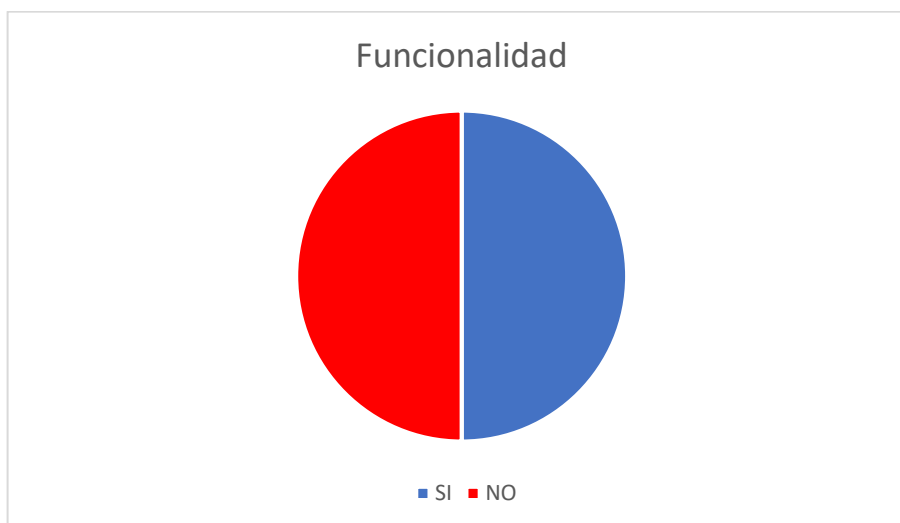


Gráfico 1– Respuestas Adecuación Funcional

4.2. Seguridad

Tabla 6. Preguntas – Seguridad

Pregunta	SI	NO
¿Se implementan controles de acceso al producto desarrollado?	7	0
¿Se implementan auditorías al producto desarrollado?	7	0
Total	14	0

Los resultados indican los participantes entienden el concepto consultado y es considerado durante el proceso de desarrollo del software y se realizan validaciones que garanticen la misma. Los participantes indican que no hay ningún procedimiento formal al respecto.



Gráfico 2– Respuestas Seguridad

4.3. Compatibilidad

Tabla 7. Preguntas – Compatibilidad

Pregunta	SI	NO
¿Se realizan pruebas de conectividad con sistemas externos?	2	5
Total	2	5

Los resultados indican que los participantes entienden el concepto consultado y no es mayormente considerado durante el proceso de desarrollo del software, los participantes manifiestan que las pruebas de interoperabilidad, en la mayoría de los casos, son desarrolladas al momento de las pruebas integrales en el cliente.

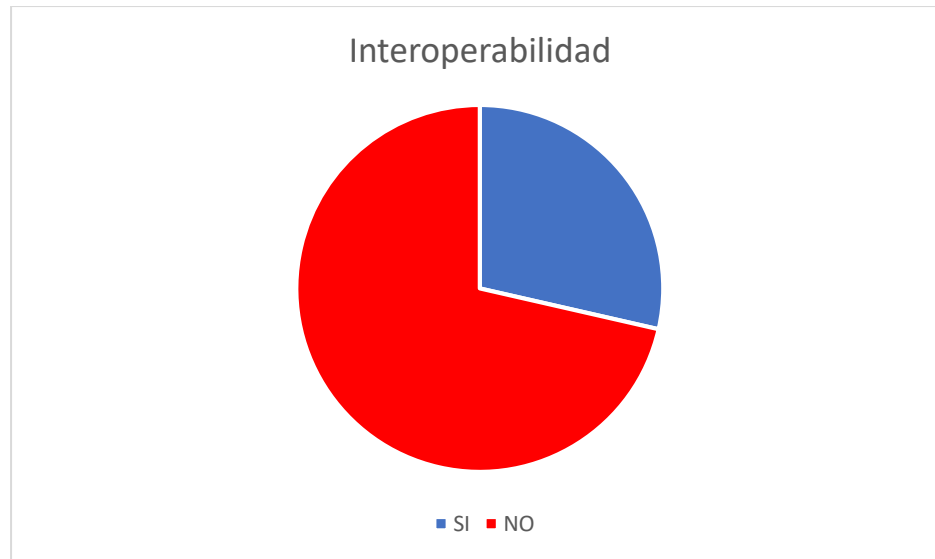


Gráfico 3 – Respuestas Compatibilidad

4.4. Fiabilidad

Tabla 8. Preguntas – Fiabilidad

Pregunta	SI	NO
¿Se desarrollan planes de prueba para el producto desarrollado?	4	3
¿Se realizan prueba para el producto desarrollado?	7	0
¿Se implementan validaciones para manejo de errores en el producto desarrollado?	7	0
¿Se implementan procesos o políticas de recuperación, en caso de fallas/caídas del producto desarrollado?	2	5
Total	20	8

Los resultados indican que los participantes entienden el concepto consultado y es considerado durante el proceso de desarrollo del software y se realizan validaciones que garanticen la misma. Algunos de los participantes manifiestan que los planes de prueba no se encuentran establecidos formalmente y que generalmente no se implementan procedimientos de recuperación automática en casos de falla.

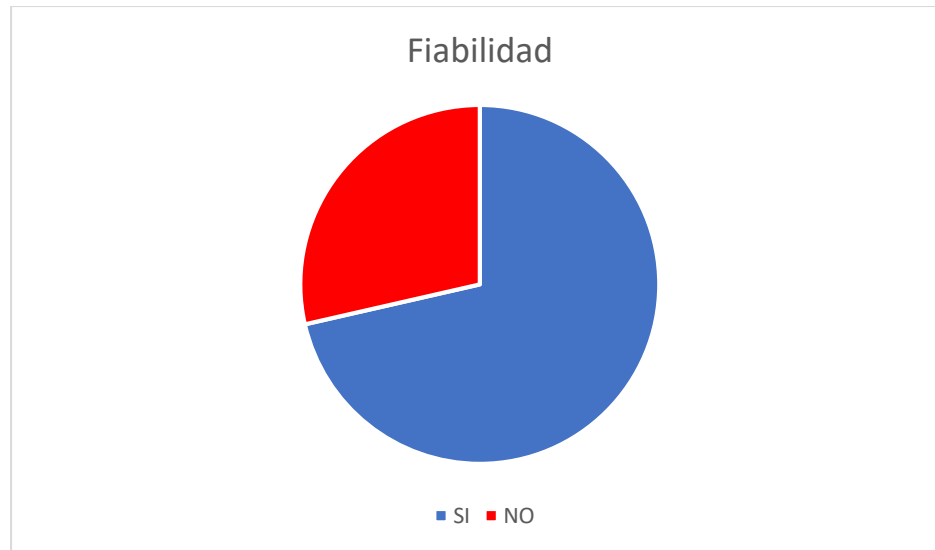


Gráfico 4 – Respuestas Fiabilidad

4.5. Usabilidad

Tabla 9. Preguntas – Usabilidad

Pregunta	SI	NO
¿Se realizan pruebas de uso y funcionalidad?	4	3
Total	4	3

Los resultados indican que los participantes entienden el concepto consultado y es considerado parcialmente durante el proceso de desarrollo del software. Algunos de los participantes manifiestan que las pruebas con el usuario final no están establecidas formalmente y su ejecución depende del analista o líder de proyecto asignado.

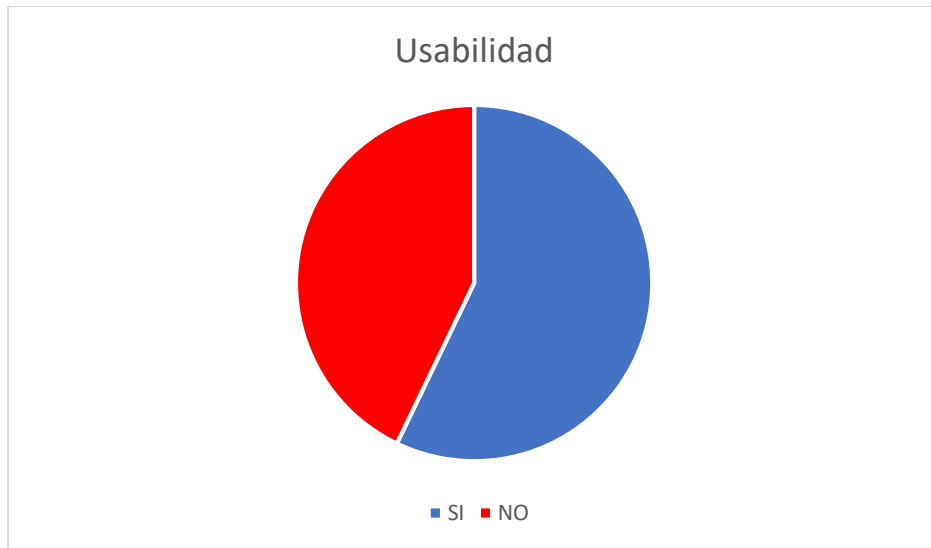


Gráfico 5 – Respuestas Usabilidad

4.6. Eficiencia de Desempeño

Tabla 10. Preguntas – Eficiencia de Desempeño

Pregunta	SI	NO
¿Se realizan pruebas de rendimiento/desempeño a nivel de tiempo de respuesta?	2	5
¿Se realizan pruebas de rendimiento/desempeño a nivel de consumo de recursos?	2	5
Total	9	5

Los resultados indican que los participantes entienden el concepto consultado y no es mayormente considerado durante el proceso de desarrollo del software. Algunos de los participantes manifiestan que las pruebas de desempeño son realizadas de acuerdo con el criterio del Líder de Proyecto o en caso de que el cliente manifieste alguna disconformidad.



Gráfico 6 – Respuestas Eficiencia de Desempeño

4.7. Mantenibilidad

Tabla 11. Preguntas – Mantenibilidad

Pregunta	SI	NO
¿Se implementan políticas de desarrollos modulares/reusables?	3	4
¿Se implementan políticas de documentación de código?	4	3
¿Se realizan validaciones al código desarrollado (complejidad ciclomática)?	2	5
Total	9	12

Los resultados indican que los participantes entienden el concepto consultado y es parcialmente considerado durante el proceso de desarrollo del software. Algunos de los participantes manifiestan no tener políticas formales para la codificación de las aplicaciones y las mismas dependen del criterio del Líder de Proyecto o Analista Funcional.

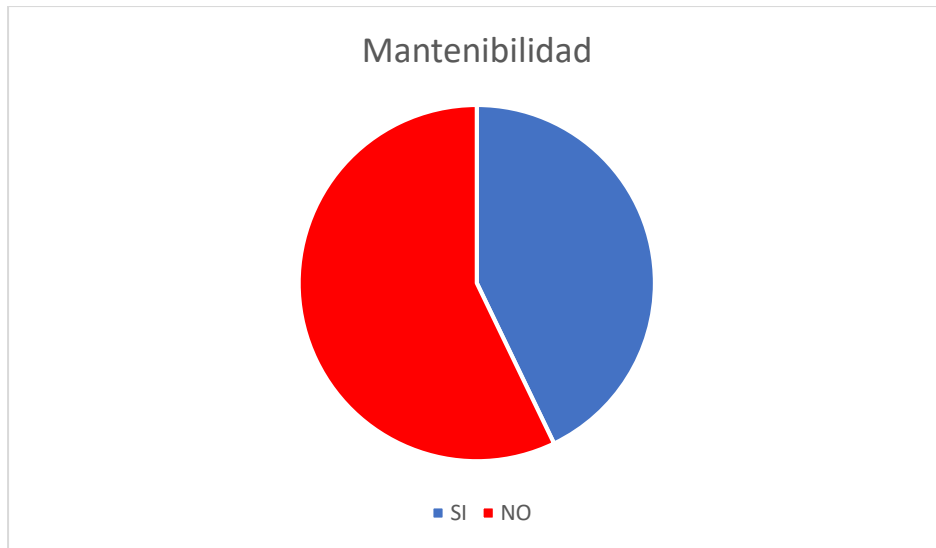


Gráfico 7 – Respuestas Mantenibilidad

4.8. Portabilidad

Tabla 12. Preguntas – Portabilidad

Pregunta	SI	NO
¿Se realizan pruebas de ambiente, conforme al ambiente de producción?	3	4
¿Se desarrollan instaladores?	3	4
¿Se validan los tiempos de instalación del producto desarrollado?	3	4
Total	9	12

Los resultados indican que los participantes entienden el concepto consultado y es parcialmente considerado durante el proceso de desarrollo del software. Algunos de los participantes manifiestan no tener políticas formales para realizar pruebas en ambientes similares a producción, desarrollo de instaladores y tiempos de instalación del producto.

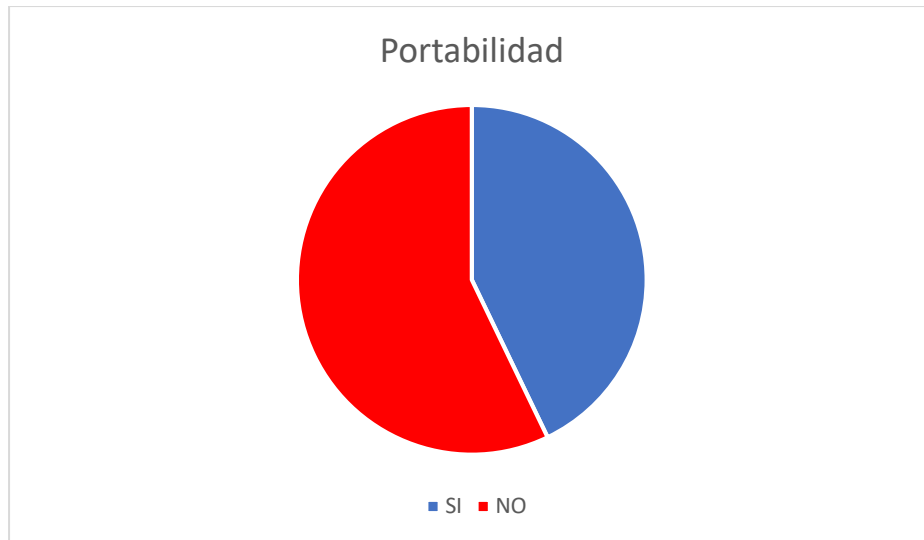


Gráfico 8 – Respuestas Portabilidad

4.9. Relación de las Preguntas con Respecto a los Objetivos Propuestos

Las preguntas realizadas en el cuestionario se relacionan en su totalidad con el objetivo 1: “Realizar un Análisis Situacional, en cuanto a las Actividades de Calidad e Identificar Mejoras en las Diferentes Etapas del Ciclo de Vida de los Proyectos”. Los hallazgos obtenidos, agrupados por características, son los siguientes:

Adecuación Funcional: con respecto a este tópico se evidenció que la funcionalidad del software en desarrollo se considera. Del mismo modo se evidencia que, dependiendo del perfil, algunos participantes manifiestan solo conocer su funcionalidad asignada o parcial, más no la globalidad.

Seguridad: con respecto a este tópico, los participantes manifiestan considerar la seguridad en la totalidad de los desarrollos. Adicionalmente informaron que no existe ningún procedimiento formal al respecto.

Compatibilidad: el resultado obtenido sobre este tópico es que en su mayoría la compatibilidad no es considerada durante las pruebas del producto desarrollado. En

la mayoría de los casos las pruebas de interoperabilidad son desarrolladas al momento de las pruebas de integrales con el cliente.

Fiabilidad: en cuanto a este tópico se pudo evidenciar que en general se considera la fiabilidad en los productos desarrollados. Adicionalmente algunos participantes manifestaron la no existencia de procedimientos formales de prueba y la inexistencia de procedimientos de recuperación automática.

Usabilidad: este tópico obtuvo respuestas fraccionadas, se evidenció que no existe un procedimiento formal de pruebas con el usuario final y la ejecución de las mismas depende del analista o líder de proyecto asignado.

Eficiencia de Desempeño: el resultado evidenció que las pruebas de eficiencia no son realizadas generalmente y dependen del criterio del Líder de Proyecto o un reporte de falla por parte del cliente.

Mantenibilidad: se pudo evidenciar que este tópico es parcialmente considerado en los proyectos. Los participantes indicaron que no existen políticas formales para la codificación de las aplicaciones desarrolladas y en general dependen del Líder de Proyecto.

Portabilidad: El resultado evidencia que este tópico es considerado parcialmente. Algunos participantes manifiestan no tener políticas formales en cuanto a pruebas en ambientes similares a producción, desarrollo de instaladores y tiempos de instalación del producto.

CAPÍTULO VI. DESARROLLO DE LA INVESTIGACIÓN

6.1. Objetivo 1 - Realizar un Análisis Situacional, en cuanto a las Actividades de Calidad e Identificar Mejoras en las Diferentes Etapas del Ciclo de Vida de los Proyectos.

Del cuestionario aplicado, la observación de campo y el análisis documental de los procesos involucrados en el desarrollo de productos de la Fábrica de Software de Grupo CIS de Venezuela, se evidenciaron diferentes causas de baja calidad por cada característica evaluada, los cuales se señalan en la siguiente figura.

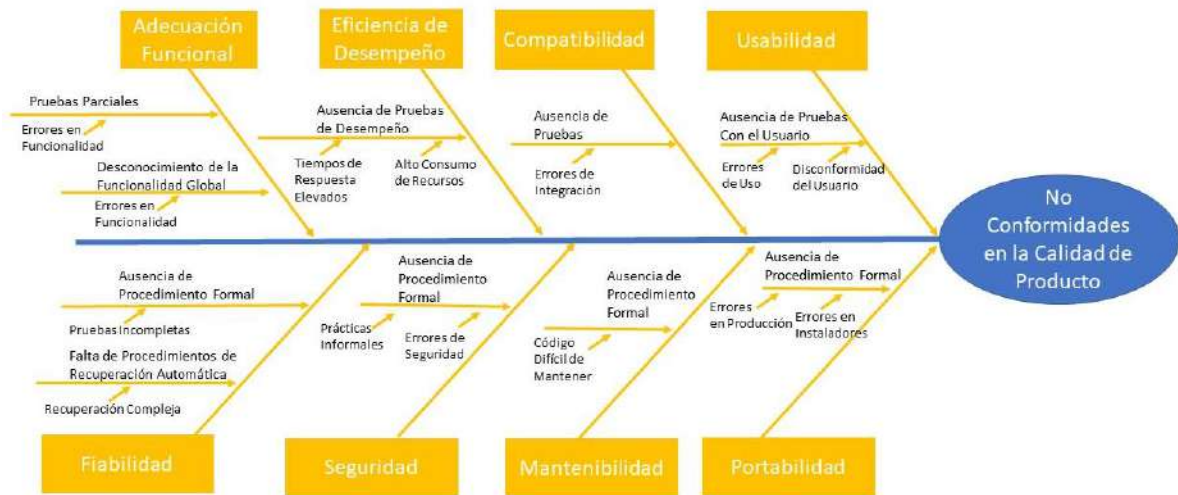


Figura 6. Diagrama Ishikawa – Análisis Situacional

Fuente: Adaptado de Ishikawa (1943).

Para el área de Adecuación Funcional se evidenciaron errores en la funcionalidad esperada, producto de pruebas parciales y desconocimiento de la funcionalidad global por parte de algunos consultores que provocaban errores luego de la integración. Con respecto a la Eficiencia de Desempeño se reportó la ausencia de pruebas relacionadas y como consecuencia problemas con los tiempos de

respuesta y alto consumo de recursos por parte de los productos desarrollados. En el área de Compatibilidad y Usabilidad se encontró la falta de pruebas tanto locales como con el usuario, esto generando errores de integración, de uso y la correspondiente disconformidad por parte del usuario.

En las áreas de Fiabilidad, Seguridad, Mantenibilidad y Portabilidad se encontraron deficiencias con respecto a la ausencia de procedimientos formales. Muchas de las actividades se realizaban, pero sin un documento formal asociado. Esto generaba problemas tales como: pruebas incompletas, falta de procedimientos de recuperación, prácticas informales, errores de seguridad, dificultades para mantener el código elaborado, errores de instalación y errores en producción.

Se realizó una revisión documental de los proyectos ejecutados en el año 2016 y por cada fase del proyecto se relacionaron los problemas de calidad encontrados.

Tabla 13. Proyectos Desarrollados en 2016

Proyecto	Inicio	Planificación	Implementación	Control	Cierre
Desarrollo de Mantenimientos SARA	- Disponibilidad de Recursos	- Ausencia de Plan de Pruebas.	- Ausencia de pruebas unitarias e integrales. - Ausencia de pruebas con el usuario.	- Falta de verificación de pruebas.	- Reportes de fallas de usuario.
Desarrollo de EFact POS – Gasolinera	- Falta de información sobre requerimientos particulares.	- Ausencia de Plan de Pruebas.	- Ausencia de pruebas unitarias e integrales. - Ausencia de pruebas con el usuario.	- Falta de verificación de pruebas.	- Reportes de fallas de usuario.
Desarrollo Datacheck – Banco de Venezuela	- Falta de información sobre requerimientos particulares.	- Ausencia de Plan de Pruebas.	- Ausencia de estándares de codificación. - Ausencia de Instaladores. - Ausencia de pruebas integrales. - Ausencia de procedimientos de recuperación automática.	- Falta de verificación de pruebas. - Falta de verificación de estándares de codificación.	- Reportes de fallas de usuario.
Desarrollo de Componente de Encriptación POS - Guatemala	- Falta de información sobre requerimientos particulares. - Disponibilidad de Recursos	- Ausencia de Plan de Pruebas.	- Ausencia de pruebas con el usuario.	- Falta de verificación de pruebas.	- Reportes de fallas de usuario.

Fuente: Grupo CIS (2017)

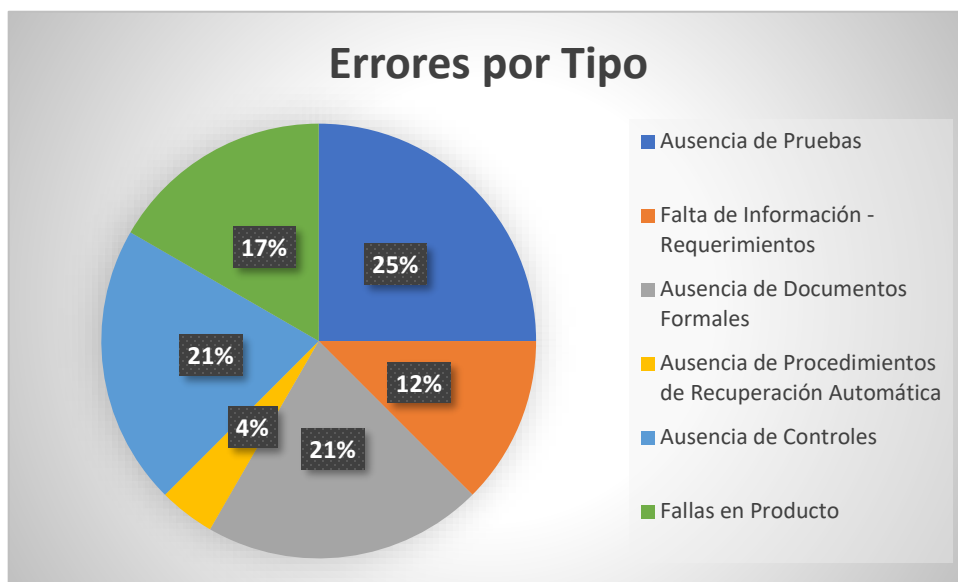


Gráfico 9 – Errores por Tipo

Al agrupar los errores por tipo, se puede evidenciar que los porcentajes más grandes se encuentran en la Ausencia de Pruebas, la Ausencia de Documentos formales y la Ausencia de Controles, todo esto generando errores en el producto final, que representan un 17% del total de errores reportados, pero individualmente representan fallas en todos los entregables que recibió el usuario.

En el siguiente gráfico se refleja el impacto de los errores por área de conocimiento:

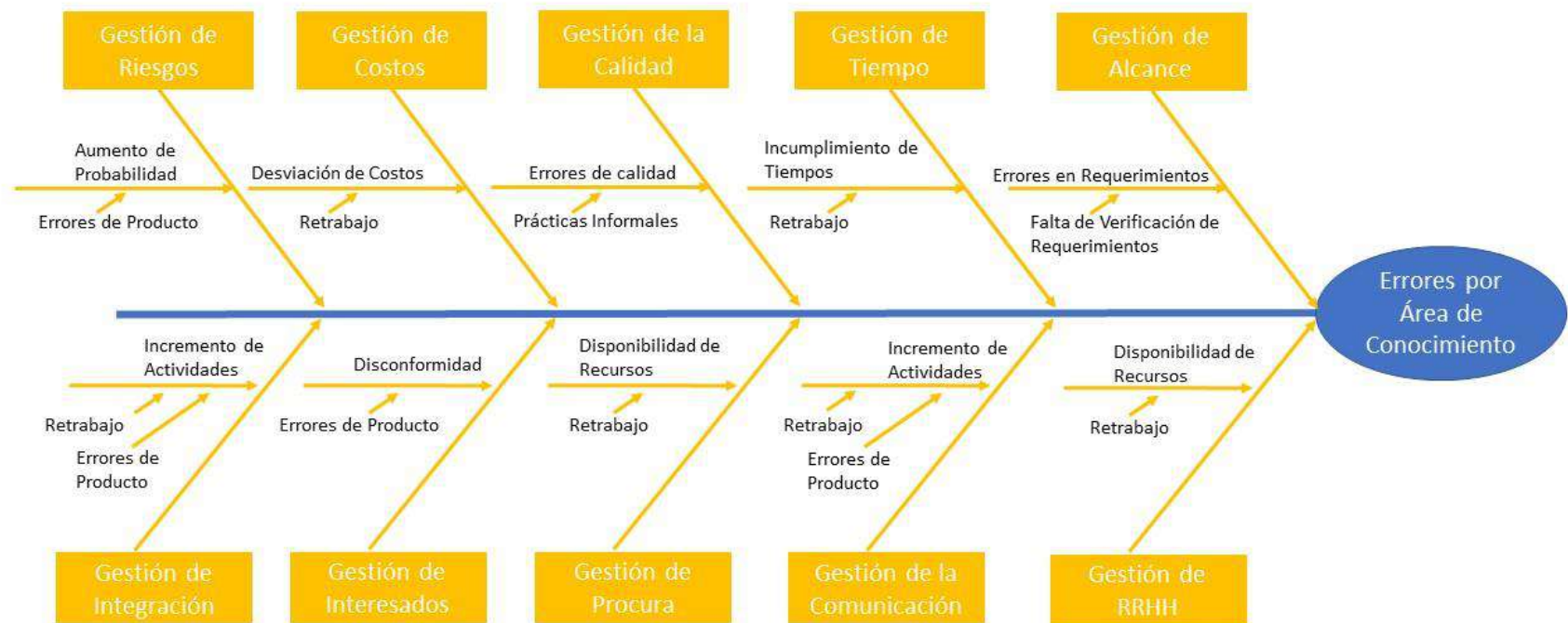


Figura 7. Diagrama Ishikawa – Errores por Área de Conocimiento
 Fuente: Adaptado de Ishikawa (1943).

Tabla 14. Matriz de Riesgos – Análisis Situacional

ID	Descripción	Probabilidad	Impacto	Pxl	Acción para Mitigarlo	Responsable	Fecha	Observaciones
1	Desviaciones Negativas en Cronograma	0.50	0.50	0.25	Actividades de Aseguramiento de Calidad	Gerente de Proyecto		
2	Disponibilidad de Recursos	0.40	0.40	0.16	Actividades de Aseguramiento de Calidad	Gerente de Proyecto		
3	Desviaciones Negativas de Costos	0.40	0.40		Actividades de Aseguramiento de Calidad	Gerente de Proyecto		
4	No conformidades en la Calidad del Producto	0.5	0.5	0.25	Actividades de Aseguramiento de Calidad Actividades de Control de Calidad	Gerente de Proyecto		

En la matriz de riesgo precedente se puede observar la probabilidad y el impacto muy altos para cada uno de los riesgos planteados, todos y cada uno de ellos, consecuencia de una mala Gestión de la Calidad.

6.2. Objetivo 2 - Evaluar los Modelos de Calidad de Software.

La calidad del producto de software es un tema que afecta a las empresas de desarrollo en la actualidad, pero que tiene muchos años siendo tratado. En el año 1977 McCall creó un modelo de calidad del producto software basado en factores de calidad tenían que ver con la operación, revisión y transición de los productos software. En 1978 Boehm publicó un modelo similar, pero que difería del modelo de McCall, en que agregaba el rendimiento y la “utilidad” del producto. En el año 1991, fue publicada la norma ISO 9126, que unifica criterios de los modelos anteriores y es cuando los modelos de calidad del producto software empiezan realmente a ser una herramienta para las empresas.

La ISO 9126 (2001), al igual que los modelos de McCall y Boehm, propone una jerarquía de atributos de calidad. La norma ISO 9126, ahora bajo el proyecto SQuaRE (Software Product Quality Requirements and Evaluation) que desarrolla la norma ISO 25000 (2014), establece un modelo de calidad basado en modelos de calidad propuestos por los investigadores durante los últimos 30 años.

En la tabla Nro. 16 se presentan las características generales de los tres modelos mencionados.

Luego de revisar las características de cada modelo descrito se puede establecer lo siguiente:

- El modelo ISO/IEC 25000 es el que presenta actualizaciones, en algunos de sus componentes, del año 2017.

- El modelo ISO/IEC 25000 es el más flexible y en consecuencia adaptable a los diferentes tamaños de desarrollos emprendidos por la Fábrica de Software de Grupo CIS de Venezuela.
- Es el que presenta el mayor número de métricas a evaluar.
- Las normas ISO constituyen las más usadas a nivel mundial.

Al revisar las fallas de calidad reportadas en los proyectos mencionados en la tabla 4, se encontraron algunas a nivel de documentación del proyecto que afectan posteriormente a la calidad del producto final. Se realizó una consulta a un grupo focal de alto nivel, para determinar en que fases de los proyectos se requerían prácticas que aseguraran la calidad del producto y adicionalmente se colocaron modelos de calidad de proceso y producto a fin de conocer la aplicabilidad de los mismos, de acuerdo con la operativa del departamento y la dinámica de la empresa.

Tabla 15. Grupo Focal – Evaluación de Modelos de Calidad

Modelo/Norma	Inicio			Planificación			Implementación			Control			Cierre		
	R1	R2	R3	R1	R2	R3	R1	R2	R3	R1	R2	R3	R1	R2	R3
ISO/IEC 9000-3	x	x	x	x	x	x									
CMMI															
ISO/IEC 25000							x	x	x	x	x	x	x	x	x
McCall															
Bohem															

Los participantes del grupo focal manifestaron su preferencia por adoptar prácticas parciales de la norma ISO/IEC 9000-3 a fin de cubrir las necesidades de aseguramiento, y el modelo propuesto por la norma ISO/IEC 25000 para las necesidades de calidad del producto. Con respecto a la calidad del proceso la decisión se basó en la complejidad de los modelos, el menor número de horas de preparación y en consecuencia menores costos asociados. Con respecto a la calidad del producto, la flexibilidad y posibilidad de establecer prioridades dependiendo del tipo de proyecto, hicieron que se inclinaran por la ISO/IEC 25000.

Tabla 16. Comparación de Modelos de Calidad

Nombre	Tipo	Características	Año	Se Enfoca	Factores	Métricas
McCall	Fijo	(+) Reusable (-) Rígido	1977	Operación: características de operación. Revisión: habilidad para ser cambiado. Transición: adaptabilidad al nuevo ambiente	Corrección, Confiabilidad, Usabilidad, Integridad o Seguridad, Eficiencia o Performance, Facilidad, Mantenimiento, Flexibilidad, Facilidad de Prueba, Grado, Portabilidad, Reusabilidad, Interoperabilidad	Facilidad de auditoría, Exactitud, Normalización de las Comunicaciones, Completitud, Concisión, Consistencia, Estandarización en los datos, Tolerancia de errores, Eficiencia en la ejecución, Facilidad de expansión, Generalidad, Independencia del hardware, Instrumentación, Modularidad, Facilidad de operación, Seguridad, Auto-Documentación
Bohem	Fijo	(+) Reusable (-) Rígido	1987	Sus características operativas. Su capacidad para soportar los cambios. Su adaptabilidad a nuevos entornos. La evaluación del desempeño del hardware.	Portabilidad, Confiabilidad, Eficiencia, Usabilidad, Chequeabilidad, Comprensibilidad, Modificabilidad	Independencia, Completitud, Consistencia, Eficiencia, Accesibilidad, Comunicatividad, Estructuración, Concisión, Legibilidad, Expansividad, Exactitud, Autodescriptividad
ISO/IEC 25000	Mixto	(+) Reusable (+) Flexible	2005	Calidad interna, calidad externa, y calidad en uso.	Adecuación Funcional	Completitud Funcional, Corrección Funcional, Pertinencia Funcional
					Eficiencia de Desempeño	Comportamiento Temporal, Utilización de recursos, Capacidad
					Compatibilidad	Coexistencia, Interoperabilidad
					Usabilidad	Inteligibilidad, Aprendizaje, Operabilidad, Protección frente a errores de usuario, Estética, Accesibilidad
					Fiabilidad	Madurez, Disponibilidad, Tolerancia a fallos, Capacidad de recuperación
					Seguridad	Confidencialidad, Integridad, No repudio, Autenticidad, Responsabilidad
					Mantenibilidad	Modularidad, Reusabilidad, Analizabilidad, Capacidad de ser modificado, Capacidad de ser probado
Portabilidad	Adaptabilidad, Facilidad de instalación, Capacidad de ser reemplazado					

Al comparar los modelos de calidad de producto, se puede confirmar que permitirían mitigar los problemas de calidad reportados y el modelo ISO/IEC 25000, en particular, cubriría el requerimiento de flexibilidad en su aplicación. Punto importante que considerar, debido a la diversidad de proyectos que se implementan en el Departamento de Fábrica de Software de Grupo CIS de Venezuela.

En el Canvas que se muestra en la figura 8 se muestran las implicaciones de la implantación de un Modelo de Gestión de la Calidad para la organización en estudio.

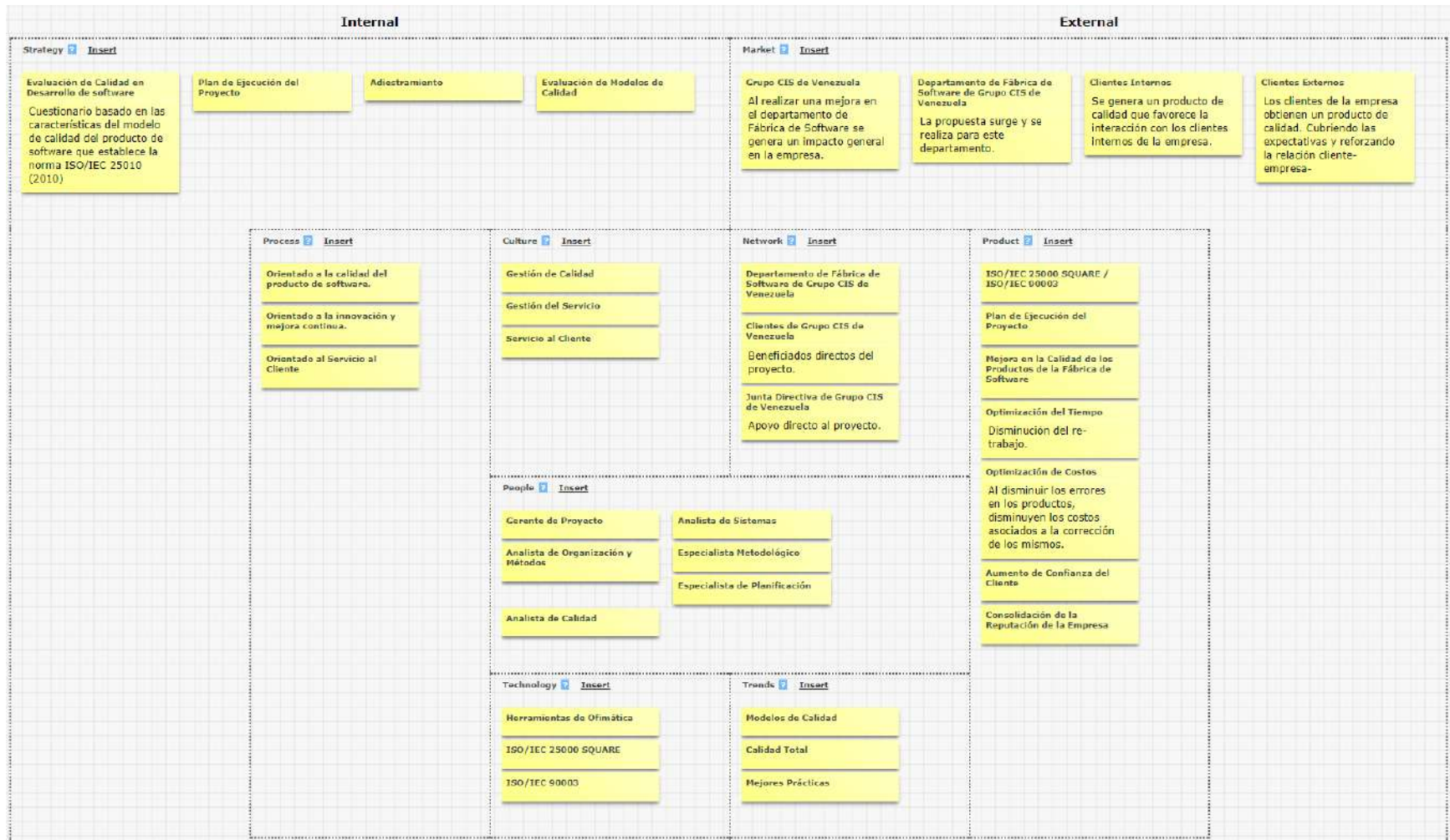


Figura 8. Open Innovation Canvas – Implementación de un Modelo de Calidad

A continuación, se detallan los elementos mencionados en el Open Innovation Canvas – Implementación de un Modelo de Calidad, desde el punto de vista interno:

Estrategia: a nivel de estrategia, la implementación de un modelo de calidad se realizaría luego de realizar un análisis situacional, evaluar modelos de calidad y elaborar un Plan de Ejecución del Proyecto.

Proceso: A nivel de procesos se estarían alineando a la Calidad del Producto de software, innovación y mejora continua y al servicio al cliente.

Cultura: A nivel cultural dentro de la organización se estarían desarrollando los conceptos de Gestión de la Calidad, Gestión de Servicio y Servicio al Cliente.

Personas: El personal involucrado a nivel de la organización incluiría un Gerente de Proyecto y un Analista de Sistemas.

Tecnología: En cuanto a la tecnología, se estarían adoptando las normas ISO/IEC 25000 SQUARE y la ISO/IEC 9000-3, en conjunto con herramientas ofimáticas.

Desde el punto de vista externo:

Mercado: La implantación de un modelo de calidad tiene un mercado interno, ya que una mejora en la calidad de los productos generados por el departamento de Fábrica de Software generaría un impacto positivo en la organización. Adicionalmente se estaría afectando a los clientes externos de la organización que reciben los entregables de los proyectos ejecutados.

Redes: Para la implementación del proyecto la red estaría compuesta por el departamento de Fábrica de Software, La Junta Directiva y los clientes de la organización.

Producto: Los productos involucrados en la implantación de un modelo de gestión de la calidad incluirían: ISO/IEC 25000 SQUARE. ISO/IEC 00003 y Plan de Ejecución del Proyecto. Del mismo modo, estos productos generarían mejoras en la calidad de los entregables finales, disminución del re-trabajo, optimización de costos, aumento de la confianza del cliente y como consecuencia la consolidación de la reputación de la empresa.

Personas: A nivel de personal externo, estarían involucrados un Analista de Organización y Métodos, un Analista de Calidad, un Especialista Metodológico y un Especialista en Planificación.

Tendencias: Con respecto a las tendencias involucradas, se pueden mencionar: Modelos de Calidad, Calidad Total y Mejores Prácticas en la ejecución de proyectos.

6.3. Objetivo 3 - Definir un Modelo de Gestión de Calidad para el Ciclo de Vida de los Proyectos de Desarrollo de Software.

Basado en la evaluación realizada en el objetivo 2, el modelo propuesto contendrá partes de la norma ISO/IEC 9000-3 para las fases de inicio y planificación del proyecto y para las fases de implementación, control y cierre el contenido en la norma ISO/IEC 25000 y en particular en el Modelo de Calidad ISO/IEC 25010.

Para las fases de inicio y planificación se proveerán una serie de formatos que permitan asegurar la generación de la documentación necesaria que dará formalidad a

Considerando que cada desarrollo tiene características particulares, el modelo plantea una ponderación de porcentaje de las características de calidad, para estimar el grado de satisfacción del sistema a evaluarse.

Como resultado del procedimiento realizado se establece una matriz de calidad y su procedimiento correspondiente, la cual se utilizará para realizar la evaluación de la calidad del producto de software de una manera precisa y adecuada.

6.3.1. Planificación de la Realización del Producto

6.3.1.1. Planificación de la Calidad

La planificación de la calidad se trata de que el proyecto se cumpla con los requisitos de calidad tales como: documentación, responsabilidades y procesos. Como resultado de este proceso se debe obtener un documento con la propuesta de planificación de la calidad, debidamente aprobado por el

Gerente/Líder del Proyecto. Dicho documento debe contener como mínimo los siguientes aspectos:

- a. Descripción del proyecto del software.
 1. Antecedentes.
 2. Objetivos.
 3. Resultados esperados.
 4. Definición de actividades de revisión, verificación, validación.
 5. Definición de actividades de monitoreo y medición.
 6. Gestión del cambio.

- b. Estudio de los requisitos del software a desarrollar.
 1. Evaluación de necesidades.
 2. Ámbito del sistema.
 3. Definición del equipo de trabajo y sus respectivas necesidades de formación.
 4. Modelo de calidad del producto (ISO/IEC 25000) y requisitos de calidad.

- c. Diseño conceptual del software a desarrollar.
 1. Plan para el diseño y desarrollo.
 2. Diseño de la base de datos.
 3. Diseño de la tecnología (hardware y software).
 4. Diseño de pruebas.

- d. Planificación de la implementación del software a desarrollar.
 1. Plan para la implementación.
 2. Mecanismos de seguimiento y mejora.

6.3.2. Procesos Relacionados con el Cliente

6.3.2.1. Descripción de requisitos según el ciclo de vida el software

Los requisitos los puede suministrar el cliente como necesidades iniciales, la organización debe hacer una especificación de los mismos a través de métodos para el levantamiento, la comprobación, el registro y la trazabilidad de los mismos. Deben compilarse en un documento operativo, deben quedar expresados en términos claros sin ambigüedades, que faciliten la validación y la aceptación del producto.

6.3.2.2. Revisión de los requisitos relacionados con el producto

La revisión de los requisitos debe asegurar que:

1. Están definidos los requisitos del producto de software.
2. Se han resuelto las diferencias entre los anteriores requisitos y los plasmados en un contrato.
3. La organización está en capacidad de cumplir con los requisitos.

6.3.3. Diseño y Desarrollo

6.3.3.1. Planificación del Diseño y Desarrollo

La planificación del diseño y desarrollo de software a desarrollar debe abordar los siguientes elementos:

1. Datos de entrada requeridos.
2. Cronograma de Actividades.

3. Actividades de diseño y desarrollo, codificación, integración, pruebas, instalación, soporte y gestión para la aceptación del producto de software.

6.3.3.2. Verificación del Diseño y Desarrollo

La verificación del diseño y desarrollo del software a desarrollar está orientada a asegurar que el resultado de las actividades de diseño y desarrollo cumplen con los requisitos del sistema.

6.3.3.3. Validación del Diseño y Desarrollo

La validación del diseño y desarrollo del software a desarrollar se realiza antes de ofrecer el producto para aceptación por parte del cliente. Debe incluir diferentes tipos de pruebas:

1. Pruebas unitarias.
2. Pruebas de integración.
3. Pruebas de certificación y de aceptación del software completo.

6.3.4. Definición de Nivel de Importancia

Las características de calidad establecidas por la norma aplican a todos los sistemas de software, sin embargo, dependiendo del tipo de sistema que se esté desarrollando, las características tendrán un grado de importancia mayor o menor para cada caso en particular de acuerdo al criterio del evaluador. En la tabla 17 se definen los niveles de importancia a utilizar en el modelo propuesto.

Tabla 17. Niveles de Importancia

Nivel de Importancia	Simbología	Porcentaje Referencial	Descripción
Alto	A	70%-100%	El grado de importancia de la característica/subcaracterística es alto y será medido.
Medio	M	25%-69%	La característica/subcaracterística no es tan relevante, pero puede o no ser medida de acuerdo al criterio del evaluador.
Bajo	B	1%-24%	La característica/subcaracterística no tiene relevancia y no será medida.
No Aplica	NA	0%	La característica/subcaracterística no puede ser medida.

6.3.5. Aplicación de Niveles de Importancia a Características de Calidad

A continuación, se ejemplifica la aplicación de los niveles de importancia a las características de calidad interna, externa y de uso.

Tabla 18. Ejemplo de Niveles de Importancia para Características de Calidad Interna

Características de Calidad Interna	
Característica	Nivel de Importancia
Adecuación Funcional	A
Fiabilidad	A
Eficiencia de Desempeño	M
Facilidad de Uso	B
Seguridad	B
Compatibilidad	B
Mantenibilidad	M
Portabilidad	NA

Tabla 19. Ejemplo de Niveles de Importancia para Características de Calidad Externa

Características de Calidad Externa	
Característica	Nivel de Importancia
Adecuación Funcional	A
Fiabilidad	M
Eficiencia de Desempeño	B
Facilidad de Uso	B
Seguridad	B
Compatibilidad	B
Mantenibilidad	M
Portabilidad	NA

Tabla 20. Ejemplo de Niveles de Importancia para Características de Calidad en Uso

Características de Calidad en Uso	
Característica	Nivel de Importancia
Efectividad	A
Eficiencia	A
Satisfacción	M
Libertad de Riesgo	B
Cobertura de Contexto	B

6.3.6. Aplicación de Niveles de Importancia a Subcaracterísticas y Atributos

A continuación, se ejemplifica la aplicación de los niveles de importancia a las subcaracterísticas y atributos de calidad interna y externa de un producto de software. Al igual que para las características, cada caso deberá ser analizado y establecido por el criterio del evaluador.

Tabla 21. Ejemplo de Niveles de Importancia para Subcaracterísticas y Atributos de Calidad Interna

Subcaracterísticas y Atributos de Calidad Interna		
Característica	Subcaracterística	Nivel de Importancia
Adecuación Funcional	Compleitud Funcional	A
	Exactitud Funcional	A
Fiabilidad	Madurez	B
	Disponibilidad	B
	Tolerancia a Fallos	B
	Recuperabilidad	B
	Comportamiento Temporal	A

Subcaracterísticas y Atributos de Calidad Interna		
Eficiencia en el Desempeño	Utilización de Recursos	B
	Capacidad	B
Facilidad de Uso	Capacidad de Reconocer su Adecuación	B
	Capacidad de Ser Entendido	M
	Operatividad	A
	Protección Frente a Errores de Usuario	A
	Estética de la Interfaz de Usuario	M
	Accesibilidad Técnica	M
Seguridad	Confidencialidad	M
	Integridad	B
	No Repudio	B
	Responsabilidad	M
	Autenticidad	A
Compatibilidad	Coexistencia	B
	Interoperabilidad	B
Mantenibilidad	Modularidad	M
	Reusabilidad	M
	Capacidad de Ser Analizado	M
	Capacidad de Ser Modificado	M
	Capacidad de Ser Probado	M
Portabilidad	Adaptabilidad	NA
	Facilidad de Instalación	NA
	Capacidad de ser Reemplazado	NA

Tabla 22. Ejemplo de Niveles de Importancia para Subcaracterísticas y Atributos de Calidad Externa

Subcaracterísticas y Atributos de Calidad Externa		
Característica	Subcaracterística	Nivel de Importancia
Adecuación Funcional	Complejidad Funcional	A
	Exactitud Funcional	A
Fiabilidad	Madurez	A
	Disponibilidad	A
	Tolerancia a Fallos	B
	Recuperabilidad	B
Eficiencia en el Desempeño	Comportamiento Temporal	A
	Utilización de Recursos	A
	Capacidad	B
Facilidad de Uso	Capacidad de Reconocer su Adecuación	B
	Capacidad de Ser Entendido	M
	Operatividad	A
	Protección Frente a Errores de Usuario	A
	Estética de la Interfaz de Usuario	M
	Accesibilidad Técnica	M
Seguridad	Confidencialidad	M
	Integridad	B
	No Repudio	B
	Responsabilidad	M
	Autenticidad	A
Compatibilidad	Coexistencia	B
	Interoperabilidad	B
Mantenibilidad	Modularidad	NA

Subcaracterísticas y Atributos de Calidad Externa		
	Reusabilidad	NA
	Capacidad de Ser Analizado	M
	Capacidad de Ser Modificado	M
	Capacidad de Ser Probado	A
Portabilidad	Adaptabilidad	NA
	Facilidad de Instalación	NA
	Capacidad de ser Reemplazado	NA

Tabla 23. Ejemplo de Niveles de Importancia para Subcaracterísticas y Atributos de Calidad en Uso

Subcaracterísticas y Atributos de Calidad en Uso		
Característica	Subcaracterística	Nivel de Importancia
Efectividad	Efectividad	A
Eficiencia	Eficiencia	A
Satisfacción	Utilidad	A
Libertad de Riesgo	Libertad del Riesgo Económico	B
	Libertad del Riesgo de Salud y Seguridad	B
	Mitigación del Riesgo Ambiental	B
Cobertura de Contexto	Integridad de Contexto	B
	Flexibilidad	B

6.3.7. Métricas de Calidad Interna y Externa

A continuación, se definen las métricas para evaluar la calidad interna y externa de los productos de software a ser evaluados:

Tabla 24. Métricas de Calidad Interna/Externa para Adecuación Funcional

Métricas para la característica de calidad: Adecuación Funcional								
Subcaracterística	Métrica	Fase de Calidad	Propósito	Método de Aplicación	Fórmula	Valor Deseado	Tipo de Medida	Recursos Utilizados
Complejidad Funcional	Complejidad de Implementación Funcional	Interna / Externa	¿Cuán completa es la implementación de acuerdo a la especificación de requerimientos?	Contar el número de las funciones indicadas en la especificación de requerimientos y el número de funciones que faltan o están incorrectas	$X=A/B$ A = número de funciones incorrectas o no implementadas. B = Número de las funciones establecidas en la especificación de requisitos. Donde $B>0$.	$0 \leq X \leq 1$ El más cercano a 0 es el mejor.	X=Contable / Contable A=Contable B=Contable	Especificación de requerimientos, Código fuente, Desarrollador y Pruebista.
Exactitud Funcional	Exactitud	Interna / Externa	¿Cuánto del estándar requerido de exactitud se cumple?	Contar el número de elementos de datos implementados con el estándar	$X=A/B$ A = número de elementos de datos implementados	$0 \leq X \leq 1$ El más cercano a 1 es el mejor.	X=Contable / Contable A=Contable B=Contable	Especificación de requerimientos, Código fuente,

Métricas para la característica de calidad: Adecuación Funcional

				específico de exactitud y el número total de elementos de datos implementados	con el estándar específico de exactitud. B = Número total de elementos de datos implementados. Donde B>0.			Desarrollador y Pruebista
	Precisión Computacional	Interna / Externa	¿Con que frecuencia ocurren resultados inexactos?	Contar el número de cálculos inexactos encontrados y tomar el tiempo de operación	$X=A/T$ A = Número de cálculos inexactos encontrados. T = Tiempo de operación. Donde T>0	El más cercano a 0/t es el mejor. Donde el peor caso es $\geq 10/t$	X=Contable / Tiempo A=Contable B=Tiempo	Código fuente, Desarrollador y Pruebista

Tabla 25. Métricas de Calidad Interna/Externa para Fiabilidad

Métricas para la característica de calidad: Fiabilidad								
Subcaracterística	Métrica	Fase de Calidad	Propósito	Método de Aplicación	Fórmula	Valor Deseado	Tipo de Medida	Recursos Utilizados
Madurez	Eliminación de Errores	Interna / Externa	¿Cuántos errores detectados han sido corregidos?	Contar el número de fallas corregidas en la fase de diseño / codificación / pruebas y el número de fallas detectadas en las pruebas	$X=A/B$ A = número de fallas corregidas en la fase de diseño / codificación / pruebas. B = Número de fallas detectadas en las pruebas. Donde $B>0$.	$0 \leq X \leq 1$ El más cercano a 1 es el mejor.	X=Contable / Contable A=Contable B=Contable	Especificación de requerimientos, Documento de casos de prueba, Código fuente, Desarrollador y Pruebista.
	Cobertura de Pruebas	Interna / Externa	¿Cuántos casos de prueba requeridos han sido ejecutados durante la etapa de pruebas?	Contar el número de casos de prueba realizados en un escenario de operación durante la prueba y el número de casos de prueba a ser realizados para cubrir los requerimientos	$X=A/B$ A = número de casos de prueba realizados en un escenario de operación durante la prueba. B = número de casos de prueba a ser realizados para cubrir los requerimientos. Donde $B>0$.	$0 \leq X \leq 1$ El más cercano a 1 es el mejor.	X=Contable / Contable A=Contable B=Contable	Especificación de requerimientos, Documento de casos de prueba, Código fuente, Desarrollador y Pruebista.

Métricas para la característica de calidad: Fiabilidad

	Tiempo Medio entre Fallos	Externa	¿Cuál es la frecuencia en que el sistema falla en operación?	Tomar el tiempo de operación y contar el número total de fallas detectadas actualmente	$X=A/T$ A = número total de fallas detectadas actualmente. T = Tiempo de operación. Donde $T>0$.	El más cercano a 0/T es el mejor.	$X=Contable / Tiempo$ $A=Contable$ $T=Tiempo$	Especificación de requerimientos, Desarrollador y Pruebista.
Disponibilidad	Tiempo de servicio	Externa	¿Cuál es el tiempo de servicio del sistema que proporciona realmente?	Tomar el tiempo de servicio del sistema que se proporciona actualmente y tomar el tiempo de servicio del sistema regulado en el cronograma operacional	$X = A/B$ A = tiempo de servicio del sistema que se proporciona actualmente. B = tiempo de servicio del sistema regulado en el cronograma operacional. Donde $B>0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor.	$X=Tiempo / Tiempo$ $A=Tiempo$ $T=Tiempo$	Especificación de requerimientos, Desarrollador y Pruebista.
	Tiempo medio de inactividad	Externa	¿Cuál es el tiempo promedio que el sistema está inactivo después de que ocurre un fallo?	Tomar el tiempo total de inactividad y contar el número de fallos observados	$X = A/T$ A = Número de fallos observados T = Tiempo total de inactividad Dónde: $T > 0$	$X = A/T$ El más cercano a 0/t es el mejor	$X=Contable/ Tiempo$ $A=Tiempo$ $B=Contable$	Especificación de requerimientos, Desarrollador y Pruebista.

Métricas para la característica de calidad: Fiabilidad

Tolerancia a fallos	Prevención de fallas	Externa	¿Cuántas fallas iniciales estuvieron bajo control para evitar fallas serias y críticas?	Contar el número de ocurrencia de fallas serias y críticas evitadas contra los casos de pruebas de fallas iniciales y el número de casos de pruebas de fallas iniciales ejecutados durante las pruebas	$X = A/B$ A = Número de ocurrencia de fallas evitadas contra los casos de pruebas de fallas iniciales. B = Número de casos de pruebas de fallas iniciales ejecutados durante las pruebas Dónde: $B > 0$	$0 \leq X \leq 1$ Cuanto más se acerque a 1 es lo mejor	X= Contable / Contable A= Contable B= Contable	Especificación de requerimientos, Desarrollador, Pruebista
	Redundancia	Interna / Externa	¿Cuántos tipos de componentes / sistemas del son instalados de forma redundante para evitar un fallo en el sistema?	Contar el número total de tipos de componentes y el número de tipos de componentes instalados de forma redundante	$X = A / B$ A= Número componentes / sistemas instalados de forma redundante B = Número total de componentes/sistemas instalados Dónde: $B > 0$	$0 \leq X \leq 1$ Cuanto más se acerque a 1 es lo mejor	X= Contable/ Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista
	Anulación de operación incorrecta	Interna	¿Cuántas funciones son implementadas con capacidad	Contar el número de funciones implementadas que evitan fallas	$X = A/B$ A = Número de operaciones	$0 \leq X \leq 1$ Cuanto más se acerque	X= Contable/ Contable	Especificación de requerimientos,

Métricas para la característica de calidad: Fiabilidad								
			de anular operaciones incorrectas?	críticas y serias causadas por operaciones incorrectas y contar el número de operaciones incorrectas presentadas	incorrectas presentadas B = Número total de funciones implementadas para anular operaciones incorrectas Dónde: B > 0	a 0 es lo mejor	A= Contable B= Contable	Código fuente, Desarrollador
Recuperabilidad	Tiempo medio de recuperación	Interna / Externa	¿Cuál es el tiempo promedio que toma el sistema en recuperarse completamente después un fallo?	Tomar el tiempo que le tomó al sistema en recuperarse y contar el número de casos en los cuales se ha observado que el sistema entró en recuperación	$X = A / T$ A = Número de casos en los cuales se ha observado que el sistema entró en recuperación T = Tiempo que le tomó al sistema en recuperarse Dónde: T > 0	$X = A/T$ El más cercano a 0/t es el mejor. Donde el peor caso es $\geq 10/t$.	X= Contable /Tiempo A= Contable B= Tiempo	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista

Tabla 26. Métricas de Calidad Interna/Externa para Eficiencia en el Desempeño

Métricas para la característica de calidad: Eficiencia en el Desempeño								
Subcaracterística	Métrica	Fase de Calidad	Propósito	Método de Aplicación	Fórmula	Valor Deseado	Tipo de Medida	Recursos Utilizados
Comportamiento del tiempo	Tiempo de respuesta	Interna / Externa	¿Cuál es el tiempo estimado para completar una tarea?	Tomar el tiempo desde que se envía la petición hasta obtener la respuesta	$X = B - A$ A= Tiempo de envío de petición B = Tiempo en recibir la primera respuesta	$0 \leq X \leq 1$ El más cercano a 0 es el mejor Donde el peor caso es $\geq 15t$	X= Tiempo / Tiempo A= Tiempo B= Tiempo	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista
	Tiempo de espera	Interna / Externa	¿Cuál es el tiempo desde que se envía una instrucción, para que inicie un trabajo, hasta que lo completa?	Tomar el tiempo cuando se inicia un trabajo y el tiempo en completar el trabajo	$X = B - A$ A= Tiempo cuando se inicia un trabajo B = Tiempo en completar el trabajo	$0 \leq X \leq 1$ El más cercano a 0 es el mejor. Donde el peor caso es $\geq 15t$	X= Tiempo / Tiempo A= Tiempo B= Tiempo	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista
	Rendimiento	Interna / Externa	¿Cuántas tareas pueden ser procesadas por unidad de tiempo?	Contar el número de tareas completadas en un intervalo de tiempo	$X = A/T$ A= Número de tareas completadas T= Intervalo de tiempo Dónde: $T > 0$	$X = A/T$ El más lejano a $0/t$ es el mejor Donde el mejor caso es $\geq 10/t$	X= Contable/ Tiempo A= Contable T= Tiempo	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista

Métricas para la característica de calidad: Eficiencia en el Desempeño								
Utilización de Recursos	Líneas de código	Interna	¿Cuántas líneas de código existen por cada función implementada?	Contar el número de líneas de código (sin tomar en cuenta espacios ni comentarios) que existen en una determinada función	$X = A$ A = Número de líneas de código	$1 \leq X \leq 50$ El más cercano a 1 es el mejor Donde el peor caso es ≥ 50 líneas de código	X= Contable A= Contable	Código fuente
	Utilización de CPU	Interna / Externa	¿Cuánto tiempo de CPU es usado para realizar una tarea dada?	Tomar el tiempo de operación y la cantidad de tiempo de CPU que se usa para realizar una tarea	$X = B - A$ A= La cantidad de tiempo de CPU que realmente es usado para realizar una tarea B = Tiempo de operación Dónde: $B > 0$	$0 \leq X \leq 1$ Cuanto más se acerque a 0 es lo mejor. Donde el peor caso es $\geq 15t$	X= Tiempo / Tiempo A= Tiempo B= Tiempo	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista
	Utilización de la memoria	Interna / Externa	¿Cuánto espacio de memoria es usado para realizar una tarea dada?	Medir la cantidad total de espacios de memoria y la cantidad de espacios de memoria que realmente es usado para	A = Cantidad de Espacios de memoria que realmente es usado para realizar una tarea	$0 \leq X \leq 15$ El más cercano a 0 es el mejor	X= Tamaño / Tamaño A= Tamaño B= Tamaño	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista

Métricas para la característica de calidad: Eficiencia en el Desempeño								
				realizar una tarea	B = Cantidad total de espacios de memoria Dónde: B > 0			
	Utilización de los dispositivos de E/S	Interna / Externa	¿Cuánto tiempo los dispositivos de E/S utilizan para realizar una tarea?	Tomar el tiempo de operación y el tiempo que los dispositivos de E/S pasan ocupados para realizar la tarea	X = B - A A = Tiempo que los dispositivos de E/S pasan ocupados para realizar la tarea B = Tiempo de operación Dónde: B > 0	0 <= X <= 15 El más cercano a 0 es el mejor	X = Tiempo/ Tiempo A = Tiempo B = Tiempo	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista
Capacidad	Número de peticiones online	Interna / Externa	¿Cuántas peticiones online pueden ser procesadas por unidad de tiempo?	Contar el número máximo de peticiones online procesadas y tomar el tiempo de operación	X = A/T A = Número máximo de peticiones online procesada T = Tiempo de operación Dónde: T > 0	X = A/T El más lejano a 0/t es el mejor Donde el mejor caso es >=10/t.	X = Contable/ Tiempo A = Contable T = Tiempo	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista
	Número de accesos simultáneos	Interna / Externa	¿Cuántos usuarios pueden acceder al sistema simultáneamente	Contar el número máximo de accesos simultáneos y tomar el tiempo de operación	X = A/T A = Número máximo de accesos simultáneos T = Tiempo de	X = A/T El más lejano a 0/t es el mejor Donde el	X = Contable/ Tiempo A = Contable B = Tiempo	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista

Métricas para la característica de calidad: Eficiencia en el Desempeño								
			en un cierto tiempo?		operación Dónde: $T > 0$	mejor caso es $\geq 10/t$		
	Sistema de transmisión de ancho de banda	Externa	¿Cuánto es el valor límite absoluto de transmisión necesaria para cumplir con las funciones?	Contar la cantidad máxima de transmisión de datos y tomar el tiempo de operación	$X = A/T$ A= Cantidad máxima de transmisión de datos B = Tiempo de operación Dónde: $T > 0$	$X = A/T$ El más lejano a 0/t es el mejor . Donde el mejor caso es $\geq 10/t$	$X =$ Tamaño / Tiempo $A =$ Tamaño $B =$ Tiempo	

Tabla 27. Métricas de Calidad Interna/Externa para Facilidad de uso

Métricas para la característica de calidad: Facilidad de uso								
Subcaracterística	Métrica	Fase de Calidad	Propósito	Método de Aplicación	Fórmula	Valor Deseado	Tipo de Medida	Recursos Utilizados
Capacidad de reconocer su adecuación	Integridad de descripción	Interna / Externa	¿Qué cantidad de funciones (o tipos de funciones) son descritas como entendibles en la descripción del producto?	Contar el número de funciones (o tipos de funciones) descritas como entendibles en la descripción del producto y contar el número total de unciones (o tipos de funciones)	$X = A/B$ A = Número de funciones (o tipos de funciones) descritas como entendibles en la descripción del producto B = Número total de funciones (o tipos de funciones) Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista
	Capacidad de demostración	Interna / Externa	¿Qué cantidad de funciones tienen la capacidad de demostración?	Contar el número de funciones implementadas con capacidad de demostración y contar el número	$X = A/B$ A = Número de funciones implementadas con capacidad de demostración	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B=	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista

Métricas para la característica de calidad: Facilidad de uso

				total de funciones que requieren capacidad de demostración	B = Número total de funciones que requieren capacidad de demostración Dónde: B > 0		Contable	
Capacidad para ser entendido	Funciones evidentes	Interna	¿Qué cantidad de funciones del producto son evidentes al usuario?	Contar el número de funciones que son evidentes al usuario y comparar con el número total de funciones.	X = A / B A= Número de funciones (o tipo de funciones) evidentes al usuario B = Número total de funciones (o tipo de funciones) Dónde: B > 0	0<=X<=1 El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Código fuente
	Efectividad de la documentación del usuario o ayuda del sistema	Interna / Externa	¿Qué cantidad de funciones están descritas correctamente en la documentación del usuario o ayuda en línea?	Contar el número de funciones descritas correctamente y contar el número total de funciones implementadas	X = A / B A= Número de funciones descritas correctamente B = Número total de funciones implementadas Dónde: B > 0	0<=X<=1 El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista

Métricas para la característica de calidad: Facilidad de uso

Operatividad	Recuperabilidad de error operacional	Interna	¿Qué cantidad de funciones pueden tolerar errores de usuario?	Contar el número de funciones implementadas con tolerancia de error de usuarios y el número total de funciones requeridas con capacidad de tolerancia.	$X = A / B$ A= Número de funciones implementadas con tolerancia de error de usuarios B = Número total de funciones requeridas con capacidad de tolerancia. Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Código fuente
	Claridad del mensaje	Interna / Externa	¿Qué cantidad de mensajes son auto explicativo?	Contar el número de mensajes implementados con explicaciones claras y el número total de mensajes implementados	$X = A / B$ A= Número de mensajes implementados con explicaciones claras B = Número total de mensajes implementados Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista
	Consistencia operacional	Interna / Externa	¿Cuántas operaciones similares pueden llevarse a cabo consecuentemente?	Contar el número de operaciones que se comportan de manera incoherente y el número total de operaciones que se comportan de forma normal	$X = A / B$ A= Número de de operaciones que se comportan de manera incoherente B = Número total de operaciones que se comportan de forma normal Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista
Protección contra errores del usuario	Posibilidad de personalización	Interna / Externa	¿Cuántas funciones y procedimientos	Contar el número de funciones implementadas	$X = A / B$ A = Número de funciones	$0 \leq X \leq 1$ El más	X= Contable	Especificación de

Métricas para la característica de calidad: Facilidad de uso								
			operacionales puede un usuario modificar para su conveniencia?	que pueden ser personalizados durante la operación y el número de funciones que requieran la capacidad de personalización	implementadas que pueden ser personalizados durante la operación B = Número de funciones que requieran la capacidad de personalización Dónde: B > 0	cercano a 1 es el mejor	/ Contable A= Contable B= Contable	requerimientos, Código fuente, Desarrollador, Pruebista
	Verificación de entradas válidas	Interna / Externa	¿Qué cantidad de ítems de entrada son validados?	Contar el número de ítems de entrada que son validados y el número de ítems que necesitan ser validados	X = A/B A= Número de ítems de entrada que son validados B = Número de ítems que necesitan ser validados Dónde : B > 0	0<=X<=1 El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista
	Prevención del uso incorrecto	Interna / Externa	¿Cuántas funciones tienen la capacidad de evitar operaciones incorrectas?	Contar el número de funciones implementadas para evitar fallos de funcionamiento provocados por un uso incorrecto y el número total de operaciones iniciales incorrectas	X = A/B A = Número operaciones iniciales incorrectas B = Número de funciones implementadas para evitar fallos de funcionamiento provocados por un uso incorrecto Dónde: B > 0	0<=X<=1 El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista
Estética de la Interfaz del usuario	Personalización de la apariencia	Interna / Externa	¿Qué cantidad de los elementos de	Contar el número de tipos de elementos de	X = A/B A= Número de elementos de	0<=X<=1 El más	X= Contable	Especificación de

Métricas para la característica de calidad: Facilidad de uso								
	de la interfaz del usuario		la interfaz de usuario pueden ser personalizados en apariencia?	interfaz que pueden ser personalizados y contar el número total de tipos de elementos de interfaz	interfaz que pueden ser personalizados B = Número total de elementos de interfaz Dónde: B > 0	cercano a 1 es el mejor	/ Contable A= Contable B= Contable	requerimientos, Código fuente, Desarrollador, Pruebista
Accesibilidad técnica	Accesibilidad física	Interna / Externa	¿A qué cantidad de funciones puede acceder un usuario con discapacidades físicas?	Contar el número de funciones a las que pueden acceder personas con discapacidad y contar el número total de funciones implementadas	$X = A/B$ A = Número de funciones a las que pueden acceder personas con discapacidad B = Número total de elementos de interfaz Dónde: B > 0	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista

Tabla 28. Métricas de Calidad Interna/Externa para Seguridad

Métricas para la característica de calidad: Seguridad								
Subcaracterística	Métrica	Fase de Calidad	Propósito	Método de Aplicación	Fórmula	Valor Deseado	Tipo de Medida	Recursos Utilizados
Confidencialidad	Capacidad de control de acceso	Interna / Externa	¿Qué tan controlable son los accesos al sistema?	Contar el número de diferentes tipos de operaciones ilegales detectados y el número de tipos de operaciones ilegales en la especificación	$X = A / B$ A = Número de diferentes tipos de operaciones ilegales detectados B = Número de tipos de operaciones ilegales en la especificación Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista
	Encriptación de datos	Interna / Externa	correctamente es la implementación de encriptación / descriptación de datos de acuerdo a la especificación de requerimientos?	Contar el número de elementos de datos encriptados/ descriptados correctamente y el número de elementos de datos que requiere encriptación / descriptación	$X = A / B$ A = Número de elementos de datos encriptados/ descriptados correctamente B = Número de elementos de datos que requiere encriptación / descriptación Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista

Métricas para la característica de calidad: Seguridad								
Integridad	Prevención de corrupción de datos	Interna / Externa	¿Hasta qué punto se puede prevenir la corrupción de datos?	Contar el número de casos de corrupción de datos ocurridos en la actualidad y el número de accesos donde se espera que ocurran daños de datos	$X = A / B$ A = Número de casos de corrupción de datos ocurridos en la actualidad B = Número de accesos donde se espera que ocurran daños de datos Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista
No repudio	Utilización de firma digital	Interna / Externa	Contar el número de eventos procesados usando firma digital y el número de eventos que requieran la propiedad de no repudio	Contar el número de eventos procesados usando firma digital y el número de eventos que requieran la propiedad de no repudio	$X = A / B$ A = Número de eventos procesados usando firma digital B = Número de eventos que requieran la propiedad de no repudio Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista
Responsabilidad	Capacidad de auditoría de acceso	Interna / Externa	¿Qué tan completa es la pista de auditoría en relación al acceso de los	Contar el número de accesos al sistema y los datos registrados en el log del sistema y el número de	$X = A / B$ A = Número de accesos ocurridos en la realidad B = Número de accesos al sistema	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable / Contable A= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista

Métricas para la característica de calidad: Seguridad

			usuarios al sistema y a los datos?	accesos ocurridos en la realidad	y los datos registrados en el log del sistema Dónde: $B > 0$		B= Contable	
Autenticidad	Métodos de autenticación	Interna / Externa	¿Qué tan bien El sistema autentica la identidad de un sujeto o recurso?	Contar el número de métodos de autenticación previstos	$X = A$ A = Número de métodos de autenticación previstos	$X \geq 0$ Donde X es mayor a 0, siendo X el mejor igual o mayor a 2	X= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista

Tabla 29. Métricas de Calidad Interna/Externa para Compatibilidad

Métricas para la característica de calidad: Seguridad								
Subcaracterística	Métrica	Fase de Calidad	Propósito	Método de Aplicación	Fórmula	Valor Deseado	Tipo de Medida	Recursos Utilizados
Co-existencia	Co-Existencia disponible	Interna / Externa	¿Qué tan adaptable es el sistema en compartir su entorno con otros sistemas sin causar efectos adversos?	Contar el número de entidades con las que el producto puede coexistir y el número de entidades en el entorno de operación que requieren de coexistencia	$X = A / B$ A = Número de entidades con las que el producto puede coexistir B = Número de entidades en el entorno de operación que requieren de coexistencia Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista
Interoperatividad	Conectividad con sistemas externos	Interna / Externa	¿Qué tan correctamente se ha implementado los protocolos de interfaz externa?	Contar el número de interfaces implementadas con otros sistemas y el número total de interfaces externas	$X = A/B$ A= Número de interfaces implementadas con otros sistemas B = Número total de Interfaces externas	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista

Métricas para la característica de calidad: Seguridad

	Capacidad de intercambiar datos	Interna / Externa	¿Qué tan exacto es el intercambio de datos entre el sistema otros sistemas de enlace?	Contar el número de datos que se han intercambiado sin problemas con otro sistema y el número total de datos que se intercambiaran	$X = A/B$ A= Número de datos que se han intercambiado sin problemas con otro sistema B = Número total de datos que se intercambiaran Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista
--	---------------------------------	-------------------	---	--	--	---	--	---

Tabla 30. Métricas de Calidad Interna/Externa para Mantenibilidad

Métricas para la característica de calidad: Mantenibilidad								
Subcaracterística	Métrica	Fase de Calidad	Propósito	Método de Aplicación	Fórmula	Valor Deseado	Tipo de Medida	Recursos Utilizados
Modularidad	Capacidad de condensación	Interna	¿Qué tan fuerte es la relación entre los componentes del sistema?	Contar el número de componentes que no son afectados por cambios de otros componentes y el número total de componentes específicos	$X = A / B$ A = Número de componentes que no son afectados por cambios de otros componentes B = Número total de componentes específicos Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Código fuente
	Acoplamiento de clases	Interna	¿Qué tan fuerte es la relación entre una función del sistema con otras clases implementadas?	Contar el número de relaciones que tiene una función con respecto a otras clases	$X = A$ A = Número de relaciones que tiene una función con respecto a otras clases	$1 \leq X \leq 4$ El más cercano a 1 es el mejor	X= Contable A= Contable	Código fuente
Reusabilidad	Ejecución de reusabilidad	Interna	¿Cuántos elementos pueden ser reutilizados?	Contar el número de elementos reutilizados y el número total de	$X = A / B$ A = Número de elementos reutilizados	$0 \leq X \leq 1$ El más	X= Contable / Contable A=	Código fuente

Métricas para la característica de calidad: Mantenibilidad								
				elementos de la biblioteca reutilizable	B = Número total de elementos de la biblioteca reutilizable Dónde: B > 0	cercano a 1 es el mejor	Contable B= Contable	
Capacidad de ser analizado	Capacidad de pistas de auditoría	Interna / Externa	¿Los usuarios pueden identificar fácilmente la operación específica que causó el fallo?	Contar el número de datos realmente grabadas durante la operación y el número de datos previstos a grabar se para controlar el estado del sistema durante la operación	X = A / B A = Número de datos realmente grabadas durante la operación B = Número de datos previstos a grabar se para controlar el estado del sistema durante la operación Dónde: B > 0	0<=X<=1 El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Código fuente
	Diagnóstico de funciones suficientes	Interna / Externa	¿Hasta qué punto las funciones de diagnóstico están preparadas o hasta qué punto funcionan para el análisis causal?	Contar el número de funciones de diagnóstico implementadas y contar el número de funciones de diagnóstico requeridas en la especificación	X = A/B A = Número de funciones de diagnóstico implementadas B = Número de funciones de diagnóstico requeridas en la especificación	0<=X<=1 El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista

Métricas para la característica de calidad: Mantenibilidad								
				de requerimientos	de requerimientos Dónde: $B > 0$			
Capacidad de ser modificado	Complejidad ciclomática	Interna	¿Cuál es la complejidad estructural de un código fuente?	Contar las instrucciones condicionales, bucles, salidas de métodos y cláusulas AND y OR dentro de los condicionales.	$X = A + 1$ A = Número de instrucciones condicionales que tiene una función	$1 \leq X < 15$ El más cercano a 1, es el mejor	X= Contable A= Contable	Código fuente
	Profundidad de herencia	Interna	¿Qué tan profunda es la jerarquía de la herencia de las clases involucradas en una determinada función?	Contar las jerarquías empleadas en una determinada función o método	$X = A$ A = Número de jerarquías empleadas para una determinada función	$0 \leq X \leq 4$ El más cercano a 0 es el mejor	X= Contable A= Contable	Código Fuente
	Grado de localización de corrección de impacto	Interna / Externa	¿Hasta qué punto los problemas causados pueden tener como	Contar el número de fallas aparecidas después que se ha resuelto un fallo y contar el	$X = A/B$ A = Número de fallas aparecidas después que se ha resuelto un fallo B = Número de	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B=	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista

Métricas para la característica de calidad: Mantenibilidad

			consecuencia un mantenimiento?	número de fallas resultas	fallas resueltas Dónde: $B > 0$		Contable	
	Complejidad de modificación	Externa	¿Con qué facilidad el desarrollador puede modificar el software para resolver problemas?	Tomar el tiempo de trabajo que le toma al desarrollador modificar y contar el número de modificaciones	$X = A/T$ A = Número de modificaciones B = Tiempo de trabajo que le toma al desarrollador modificar Dónde: $T > 0$	$X = A/T$ El más Lejano a $0/t$ es el mejor	X= Contable / Tiempo A= Contable B= Contable	Desarrollador
	Índice de éxito de modificación	Externa	¿Hasta qué punto puede el sistema ser operado sin fallas después del mantenimiento?	Contar el número de problemas dentro de un determinado período antes de Mantenimiento y contar el número de problemas en el mismo período después del mantenimiento	$X = A/B$ A = Número de problemas dentro de un determinado período antes de mantenimiento B = Número de problemas en el mismo período después del mantenimiento Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Desarrollador
Capacidad de ser probado	Complejidad funcional de funciones de pruebas	Interna	¿Son las funciones de prueba completas y	Contar el número de funciones de prueba implementadas	$X = A/B$ A = Número de funciones de prueba implementadas	$0 \leq X \leq 1$ El más	X= Contable / Contable A=	Código fuente, Pruebista

Métricas para la característica de calidad: Mantenibilidad

			fáciles de implementar?	y contar el número de funciones de prueba requeridas	B = Número de funciones de prueba requeridas Dónde: $B > 0$	cercano a 1 es el mejor	Contable B= Contable	
Capacidad de prueba autónoma	Interna		¿Qué tan independiente es el software al ser probado?	Contar el número de pruebas que están dependiendo de otros sistemas y contar el número total de pruebas dependientes con otros sistemas	$X = A/B$ A = Número de pruebas que están dependiendo de otros sistemas B = Número total de pruebas dependientes con otros sistemas Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Código fuente, Pruebista
Capacidad de reinicio de pruebas	Externa		¿Con qué facilidad se pueden llevar a cabo las pruebas nuevamente después del mantenimiento?	Contar el número de casos en los cuales el mantenedor puede pausar y restaurar las pruebas y contar el número de casos de pausa en la ejecución de pruebas	$X = A/B$ A = Número de casos en los cuales el mantenedor puede pausar y restaurar las pruebas B = Número de casos de pausa en la ejecución de pruebas Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Desarrollador, Pruebista

Tabla 31. Métricas de Calidad Interna/Externa para Portabilidad

Métricas para la característica de calidad: Portabilidad								
Subcaracterística	Métrica	Fase de Calidad	Propósito	Método de Aplicación	Fórmula	Valor Deseado	Tipo de Medida	Recursos Utilizados
Adaptabilidad	Adaptabilidad en entorno hardware	Interna / Externa	¿Es el sistema lo suficientemente capaz de adaptarse al entorno de hardware?	Contar el número funciones operativas de las tareas que no se hayan completado durante las pruebas operativas con el entorno de hardware y contar el número total de funciones las cuales han sido probadas	$X = A / B$ A = Número funciones operativas de las tareas que no se hayan completado durante las pruebas operativas con el entorno de hardware B = Número total de funciones que han sido probadas. Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista
	Adaptabilidad en entorno de software	Interna / Externa	¿Es el sistema lo suficientemente capaz de adaptarse al entorno del sistema de software?	Contar el número funciones operativas de las tareas que no se hayan completado durante las pruebas operativas con el sistema y contar el	$X = A/B$ A = Número de funciones operativas de las tareas que no se hayan completado durante las pruebas operativas con el sistema	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista

Métricas para la característica de calidad: Portabilidad

				número total de funciones las cuales han sido probadas	B = Número total de funciones que han sido probadas Dónde: B > 0			
	Adaptabilidad en entorno empresarial	Interna / Externa	¿Es el sistema lo suficientemente capaz de adaptarse al entorno operacional?	Contar el número funciones operativas de las tareas que no se hayan completado durante las pruebas operativas con usuarios del entorno empresarial y contar el número total de funciones las cuales han sido probadas	X = A/B A = Número de funciones operativas de las tareas que no se hayan completado durante las pruebas operativas con usuarios del entorno empresarial B = Número total de funciones que han sido probadas Dónde: B > 0	0<=X<=1 El más cercano a 0 es el mejor	X= Contable / Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista
Capacidad de ser Instalado	Eficiencia en el tiempo de instalación	Externa	¿Cuánto tiempo es requerido para realizar una instalación?	Contar el tiempo total transcurrido al instalar el sistema y contar el número de reintentos al instalar el sistema	X = A/B A = Número casos en que los usuarios tuvieron éxito al instalar el sistema cambiando proceso de instalación para su conveniencia	X = A/T El más lejano a 0/t es el mejor		

Métricas para la característica de calidad: Portabilidad

					B = Número total de casos en que los usuarios han intentado cambiar el proceso de instalación para su conveniencia			
	Facilidad de instalación	Externa	¿Puede fácilmente el usuario o el desarrollador instalar el software en un entorno operacional?	Contar el número de casos en que los usuarios tuvieron éxito al instalar el sistema cambiando el proceso de instalación para su conveniencia y contar el número total de casos en que los usuarios han intentado cambiar el proceso de instalación para su conveniencia		$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Desarrollador, Pruebista

Métricas para la característica de calidad: Portabilidad

Capacidad de ser Reemplazado	Consistencia en la función de soporte al usuario	Interna / Externa	¿Cuán consistente es el nuevo componente con la interfaz de usuario existente?	Contar el número de nuevas funciones que son consideradas como no consistentes por el usuario y contar el número de nuevas funciones	$X = A/B$ A = Número de nuevas funciones que son consideradas como no consistentes por el usuario B = Número de nuevas funciones Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0 es el mejor	X= Contable / Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Pruebista
	Inclusividad Funcional	Externa	¿Pueden fácilmente las funciones ser utilizadas después de ser cambiadas a por otras similares?	Contar el número de funciones que producen resultados similares con anterioridad y que no se han exigido cambios y contar el número de funciones probadas que son similares a las funciones proporcionadas por otro software para ser reemplazado	$X = A/B$ A = Número de funciones que producen resultados similares con anterioridad y que no se han exigido cambios B = Número de funciones probadas que son similares a las funciones proporcionadas por otro software para ser reemplazado	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Desarrollador, Pruebista

Métricas para la característica de calidad: Portabilidad

					Dónde: $B > 0$			
	Uso continuo de datos	Externa	¿Pueden los datos fácilmente ser utilizados después de reemplazar el software por otro similar?	Contar el número de datos que son continuamente utilizables por el software a ser reemplazado y contar el número de datos que son continuamente reutilizables por el software a ser reemplazado	$X = A/B$ A = número de datos que son continuamente solo utilizables por el software a ser reemplazado B = Número de datos que son reutilizables por el software a ser reemplazado Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Desarrollador, Pruebista

6.3.8. Métricas de Calidad en Uso

Tabla 32. Métricas de Calidad en Uso para Efectividad

Métricas para la característica de calidad: Efectividad								
Subcaracterística	Métrica	Fase de Calidad	Propósito	Método de Aplicación	Fórmula	Valor Deseado	Tipo de Medida	Recursos Utilizados
Efectividad	Compleitud de la tarea	Uso	¿Qué cantidad de tareas son completadas correctamente?	Contar el número de tareas completadas y el número total de tareas intentadas	$X = A / B$ A = Número de tareas completadas B = Número total de tareas intentadas Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Usuario
	Efectividad de la tarea	Uso	¿Qué cantidad de los objetivos de la tarea se realiza completamente?	Tomar el valor proporcional de cada componente faltante o incorrecto en la salida de la tarea	$X = A/B$ A= Cantidad de Objetivos completados por la tarea. B= Cantidad de objetivos planeados que realice la tarea	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Usuario
	Frecuencia de error	Uso	¿Cuál es la frecuencia de los errores cometidos por el usuario en comparación con lo planeado?	Contar el número de errores cometidos por los usuarios y contar el número de tareas	$X = A/B$ A = Número de errores cometidos por los usuarios B = Número de tareas Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable / Contable A= Contable B= Contable	Usuario

Fuente: ISO/IEC 25023.

Tabla 33. Métricas de Calidad en Uso para Eficiencia

Métricas para la característica de calidad: Eficiencia								
Subcaracterística	Métrica	Fase de Calidad	Propósito	Método de Aplicación	Fórmula	Valor Deseado	Tipo de Medida	Recursos Utilizados
Eficiencia	Tiempo de la tarea	Uso	¿Cuánto tiempo se tarda en completar una tarea en comparación con lo planeado?	Tomar el tiempo planeado y el tiempo actual	$X = A / B$ A= Tiempo actual B = Tiempo planeado Dónde: $A > 0$	$0 \leq X \leq 1$ Si $A \leq B$ el más cercano a 0 es lo mejor. Si $A > B$ será considerado como el peor caso	X= Tiempo / Tiempo A= Tiempo B= Tiempo	Usuario
	Tiempo relativo de la tarea	Uso	¿Cuánto tiempo necesita un usuario normal en completar una tarea en comparación con un experto?	Tomar el tiempo que completa una tarea un usuario normal y el tiempo que completa una tarea un usuario experto	$X = A / B$ A = Tiempo que completa una tarea un usuario experto B = Tiempo que completa una tarea un usuario normal Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, el mejor	X= Tiempo / Tiempo A= Tiempo B= Tiempo	Usuario
	Eficiencia de la tarea	Uso	¿Qué tan eficientes son los usuarios?	Contar el número de tareas efectivas y tomar el	$X = A/T$ A= Número de tareas efectivas	$0 \leq X \leq 1$ El más cercano a 1 es lo mejor	X= Contable / Contable	Usuario, Especificación de requerimientos

Métricas para la característica de calidad: Eficiencia

				tiempo de la tarea	T= Tiempo de la tarea Dónde: T > 0		A= Contable B= Contable	
Eficiencia relativa de la tarea	Uso	¿Qué tan eficiente es un usuario comparado con lo planeado?	Contar el número de tareas eficientes realizadas por un usuario ordinario y contar el número de tareas eficientes planeadas	X = A/B A = Número de tareas eficientes realizadas por un usuario ordinario B = Número de tareas eficientes planeadas Dónde: B > 0	0<=X<=1 El más cercano a 1 es lo mejor	X= Contable / Contable A= Contable B= Contable	Usuario, Especificación de requerimientos	
Productividad económica	Uso	¿Qué tan rentable es el usuario?	Contar el número de tareas efectivas y tomar el costo total de las tareas	X = A/B A = Número de tareas efectivas B = Numero de tareas totales Dónde: B > 0	0<=X<=1 El más cercano a 1 es lo mejor	X= Contable / Contable A= Contable B= Contable	Usuario, Especificación de requerimientos	
Porcentaje productivo	Uso	¿Cuál es el porcentaje de tiempo que el usuario realiza	Tomar el tiempo de productividad y el tiempo de la	X = A/B A = Tiempo de la tarea B = Tiempo de	0<=X<=1 Si A<=B el más cercano	X= Tiempo / Tiempo A= Tiempo B= Tiempo	Usuario	

Métricas para la característica de calidad: Eficiencia

Métricas para la característica de calidad: Eficiencia								
			acciones de productividad?	tarea	productividad. Dónde: $B > 0$	a 0 es lo mejor. Si $A > B$ será considerado como el peor caso		
	Numero relativo de acciones del usuario	Uso	¿Cuál es el número de acciones mínimas necesarias que realizan los usuarios?	Contar el número de acciones realizadas por los usuarios y contar el número de acciones necesarias actualmente	$X = A/B$ A = Número de acciones realizadas por los usuarios B = Número de acciones necesarias actualmente Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es lo mejor	X= Contable/ Contable A= Contable B= Contable	Usuario

Tabla 34. Métricas de Calidad en Uso para Satisfacción

Métricas para la característica de calidad: Satisfacción								
Subcaracterística	Métrica	Fase de Calidad	Propósito	Método de Aplicación	Fórmula	Valor Deseado	Tipo de Medida	Recursos Utilizados
Utilidad	Nivel de satisfacción	Uso	¿Qué tan satisfecho está el usuario?	Realizar un cuestionario sobre el nivel de satisfacción sobre el sistema.	$X = A / B$ A= Numero de preguntas con respuestas satisfactorias B = Número total de preguntas realizadas en el cuestionario. Dónde: $A > 0$	$0 \leq X \leq 1$ El más cercano a 1 es lo mejor	X= Contable/ Contable A= Contable B= Contable	Usuario
	Uso discrecional de las funciones	Uso	¿Qué porcentaje de los usuarios optan por utilizar las funciones del sistema?	Observación de uso	$X = A/B$ A= Número de Funciones específicas del software que se utilizan B= Número total de funciones que están destinados a ser usados Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es lo mejor	X= Contable/ Contable A= Contable B= Contable	Usuario

Métricas para la característica de calidad: Satisfacción

	Porcentaje de quejas de los clientes	Uso	¿Cuál es el porcentaje de quejas realizadas por los clientes?	Contar el número de clientes que se quejan y contar el número total de clientes	$X = A/B$ A = Número de clientes que se quejan B = Número total de clientes Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0 es lo mejor	X= Contable/ Contable A= Contable B= Contable	Usuario
--	--------------------------------------	-----	---	---	---	---	---	---------

Tabla 35. Métricas de Calidad en Uso para Libertad de riesgo

Métricas para la característica de calidad: Libertad de riesgo								
Subcaracterística	Métrica	Fase de Calidad	Propósito	Método de Aplicación	Fórmula	Valor Deseado	Tipo de Medida	Recursos Utilizados
Libertad del riesgo económico	Retorno de la Inversión (ROI)	Uso	¿Cuál es el retorno de la inversión?	Consultar los beneficios obtenidos y el capital invertido	$X = A / B$ A=Beneficios obtenidos B=Beneficios esperados. Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es lo mejor	X= Contable/ Contable A= Contable B= Contable	Cliente
	Tiempo para lograr el retorno de la inversión	Uso	¿El retorno de la inversión es logrado en un tiempo aceptable?	Tomar el tiempo para lograr el ROI y tomar el tiempo aceptable para lograr el ROI	$X = A/B$ A= Tiempo real para lograr el ROI B= Tiempo aceptable para lograr el ROI Dónde: $B > 0$	$0 \leq X \leq 1$ Si $A \leq B$ el más cercano a 0 es lo mejor. Si $A > B$ será considerado como el peor caso	$X =$ Tiempo / Tiempo A = Tiempo B = Tiempo	Usuario
	Rendimiento relativo de negocios	Uso	¿Qué tan comparable es el rendimiento del negocio con otras empresas de primera clase	Consultar el monto de la inversión de TI o de las ventas de la empresa y el monto de	$X = B/A$ A = Monto de inversión de TI o las ventas planeadas de la empresa para la	$0 \leq X \leq 1$ Si $B \leq A$ el más cercano a 1 es lo mejor. Si $B > A$		Cliente

Métricas para la característica de calidad: Libertad de riesgo

			en la industria o en la misma empresa	inversión de TI o las ventas planeadas de la empresa para la comparación	comparación B= Monto real de la inversión de TI o de las ventas de la empresa Dónde: $B > 0$	será considerado como el mejor caso		
Balanced Score Card	Uso	Los beneficios de la inversión en IT evaluados utilizando los Balanced Score Card para cumplir los objetivos		Consultar el resultado del BSC y el BSC planeado	$X = A/B$ A = Resultado del BSC B = BSC planeado Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, mejor		Cliente
Tiempo de entrega	Uso	¿Cuál es el tiempo de entrega para cumplir los con objetivos?		Consultar el tiempo de entrega actual o retrasos en las entregas y el tiempo de entrega planeado o retrasos en las entregas	$X = A/B$ A = Tiempo de entrega planeado o retrasos en las entregas B = Tiempo de entrega actual o retrasos en las	$0 \leq X \leq 1$ Si $A \leq B$ el más cercano a 0 es lo mejor. Si $A > B$ será considerado como el peor caso	X= Tiempo/ Tiempo A= Tiempo B= Tiempo	Cliente
Ganancias para cada	Uso	Las ganancias de cada		Consultar los ingresos reales	$X = A/B$	$0 \leq X \leq 1$ Si $A \leq B$ el	X= Contable/	Cliente

Métricas para la característica de calidad: Libertad de riesgo

	cliente		cliente al cumplir con sus objetivos	de un cliente y los ingresos planeados de un cliente	A = Ingresos reales de un cliente B = Ingresos planeados de un cliente Dónde: $B > 0$	más cercano a 1 es lo mejor. Si $A > B$ será considerado como el mejor caso	Contable A= Contable B= Contable	
	Errores con consecuencias económicas	Uso	La frecuencia de errores humanos o del sistema con consecuencias económicas	Contar el número de errores con consecuencias económicas y contar número total de situaciones de uso	$X = A/B$ A = Número de errores con consecuencias económicas B = Número total de situaciones de uso Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0 es el mejor	X= Contable/ Contable A= Contable B= Contable	Cliente
	Corrupción del software	Uso	La frecuencia de corrupción del software resultado de errores humanos o del sistema	Contar el número de ocurrencias de corrupción del software y contar número total de situaciones de uso	$X = A/B$ A = Número de ocurrencias de corrupción del software B = Número total de situaciones de uso Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0 es el mejor	X= Contable/ Contable A= Contable B= Contable	Cliente

Métricas para la característica de calidad: Libertad de riesgo

Libertad del riesgo de salud y seguridad	Frecuencia de problemas en la salud y seguridad del usuario	Uso	La frecuencia de problemas de salud entre los usuarios del producto	Contar el número de usuarios que notificaron problemas de salud y contar el número total de usuarios	$X = A/B$ A = Número de usuarios que notificaron problemas de salud B = Número total de usuarios Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0 es el mejor	X= Contable/ Contable A= Contable B= Contable	Usuario
	Impacto en la salud y seguridad del usuario	Uso	El impacto en la salud y la seguridad en los usuarios del producto	Contar el número de personas afectadas, tomar el tiempo y el grado de importancia	$X = A/T$ A = Número de personas afectadas T = Tiempo	$0 \leq X \leq 5$ El más cercano a 0 es el mejor		
	Seguridad de las personas afectadas por el uso del sistema	Uso	La incidencia de riesgo para las personas afectadas por el uso del sistema	Contar el número de personas puestas en peligro y contar el número total de personas potencialmente afectadas por el sistema	$X = A/B$ A = Número de personas puestas en peligro B = Número total de personas potencialmente afectadas por el sistema	$0 \leq X \leq 1$ El más cercano a 0 es el mejor	X= Contable/ Contable A= Contable B= Contable	Cliente

Métricas para la característica de calidad: Libertad de riesgo								
					Dónde: $B > 0$			
Libertad del riesgo ambiental	Impacto Ambiental	Uso	El impacto ambiental de la elaboración y el uso del sistema	Estimar el impacto ambiental y el impacto ambiental aceptable	$X = A/B$ $A = \text{Impacto ambiental aceptable}$ $B = \text{Impacto Ambiental real}$ Dónde: $B > 0$	$0 \leq X \leq 1$ Si $A \leq B$ el más cercano a 0 es lo mejor. Si $A > B$ será considerado como el peor caso	$X =$ Contable/ Contable $A =$ Contable $B =$ Contable	Usuario, cliente, desarrollador

Tabla 36. Métricas de Calidad en Uso para Cobertura de contexto

Métricas para la característica de calidad: Cobertura de contexto								
Subcaracterística	Métrica	Fase de Calidad	Propósito	Método de Aplicación	Fórmula	Valor Deseado	Tipo de Medida	Recursos Utilizados
Complejidad de Contexto	Complejidad de Contexto	Uso	Porcentaje en que el producto puede utilizarse con facilidad en contextos de uso	Contar el número de contextos con la facilidad de uso inaceptable y el número total de distintos contextos de uso	$X = A/B$ A= Número de distintos contextos de uso inaceptables B = Número total de distintos contextos de uso Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es lo mejor	X= Contable/ Contable A= Contable B= Contable	Cliente
Flexibilidad	Función flexible del diseño	Uso	Grado en que el producto puede adaptarse para satisfacer las diferentes necesidades de los usuarios	Contar el número de características diseñadas con completa flexibilidad y contar el número total de características de diseño	$X = A/B$ A= Número de características diseñadas con completa flexibilidad B = Número total de características de diseño Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable/ Contable A= Contable B= Contable	Usuario

6.3.9. Modelo de Indicadores y Métricas

6.3.9.1. Ponderación en porcentaje de las características de calidad interna, externa en uso más relevantes para el producto software

La ponderación que se otorgara a las características de calidad interna, externa y en uso dependerá del nivel de importancia que fue asignado al producto software. Tomando en cuenta que las ponderaciones que serán asignadas dependerán del criterio del evaluador y del tipo de producto a evaluarse, además estas ponderaciones deben ser divididas entre las características que fueron aplicadas, y la sumatoria no debe pasar del 100%.

Tabla 37. Ejemplo de ponderación en porcentajes para la calidad interna

Características de Calidad Externa		
Característica	Nivel	Importancia
Adecuación funcional	M	25%
Fiabilidad	B	0%
Eficiencia en el desempeño	M	15%
Facilidad de uso	M	15%
Seguridad	M	15%
Compatibilidad	B	0%
Mantenibilidad	A	30%
Portabilidad	N/A	0%

6.3.9.2. Niveles de puntuación final para la calidad interna, externa y en uso

La escala de medición que se presenta en la Tabla 29 se utilizará para analizar el resultado final de las características de calidad interna, externa y en uso. El valor total obtenido establecerá el nivel de puntuación final que se le asignará al producto software después de su evaluación.

Tabla 38. Niveles de puntuación final para la calidad interna, externa y en uso

Escala de medición	Niveles de puntuación	Grado de satisfacción
8.75 - 10	Cumple con los requisitos	Muy satisfactorio
5 – 8.74	Aceptable	Satisfactorio
2.75 – 4.9	Mínimamente aceptable	Insatisfactorio
0 – 2.74	Inaceptable	

Como resultado de la evaluación se obtendrá una matriz como la detallada en la tabla 30.

Tabla 39. Niveles de puntuación final para la calidad interna, externa y en uso

Calidad	Calidad del Sistema	Nivel de Puntuación	Grado de Satisfacción
Interna	7.5	Aceptable	Satisfactorio
Externa	8.6	Aceptable	Satisfactorio
Uso	9.07	Cumple con los Requisitos	Muy Satisfactorio
Total	8.39	Aceptable	Satisfactorio

6.4. Objetivo 4 - Diseñar el Proyecto de Ejecución del Plan de Implementación del Modelo de Gestión de Calidad Propuesto.

6.4.1. Plan de Gestión del Alcance

6.4.1.1. Objetivos del Proyecto

El objetivo del presente proyecto es la implementación del Modelo de Calidad PLANGECA en el departamento de Fábrica de Software de Grupo CIS de Venezuela, C.A., en un lapso de tres (3) meses, logrando la aceptación del modelo.

6.4.1.2. Descripción del Alcance del Proyecto

El alcance del presente proyecto abarca la implementación del Modelo de Calidad PLANGECA en el departamento de Fábrica de Software de Grupo CIS de Venezuela, en el lapso de tres (3) meses. El modelo a implementar debe permitir establecer la importancia de cada característica de calidad a evaluar, dependiendo del proyecto a implementar.

6.4.1.3. Estructura Desagregada de Trabajo

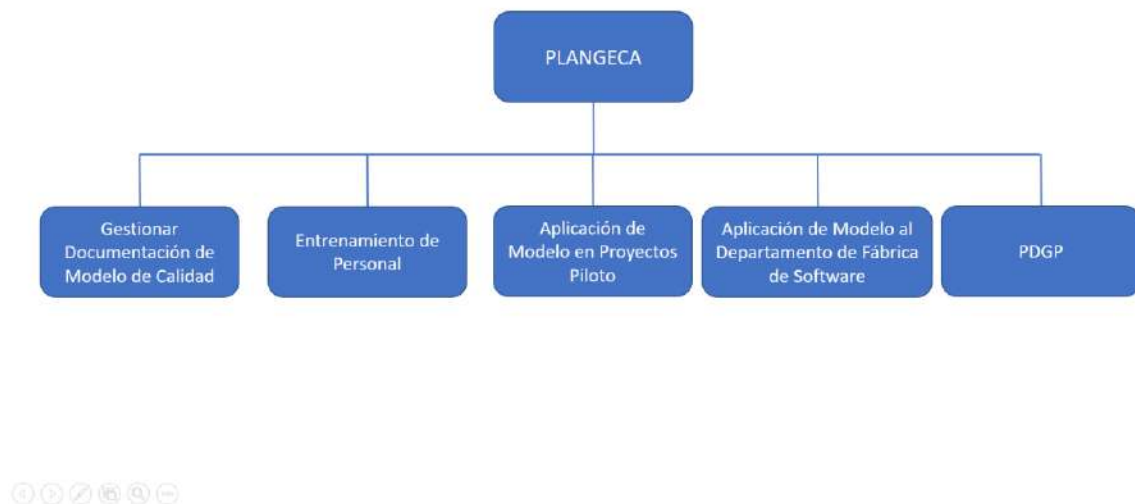


Figura 9. EDS - PLANGECA

6.4.1.4. Criterios de Aceptación

1. PLANGECA se encuentre implementado en el Departamento de Fábrica de Software de Grupo CIS de Venezuela.
2. Se hayan culminado con las actividades descritas en el cronograma de trabajo y el Departamento de Fábrica de Software haya firmado la carta de aceptación correspondiente.
3. Hayan transcurrido quince (15) días de presentados los informes de seguimiento, sin haber recibido comunicación formal por parte del Departamento de Fábrica de Software relacionando las observaciones necesarias para su aceptación.
4. PLANGECA se encuentre en producción.

6.4.1.5. Criterios de Calidad

1. El modelo de calidad implementado debe permitir establecer la importancia de cada característica de calidad a evaluar para cada proyecto a implementar.
2. El modelo de calidad debe cubrir todas las etapas del proceso de desarrollo del software.

6.4.1.6. Restricciones del Proyecto

1. El modelo de calidad debe ser implementado únicamente en el Departamento de Fábrica de Software.
2. El tiempo de implementación del proyecto es de tres (3) meses.

6.4.1.7. Organigrama Inicial

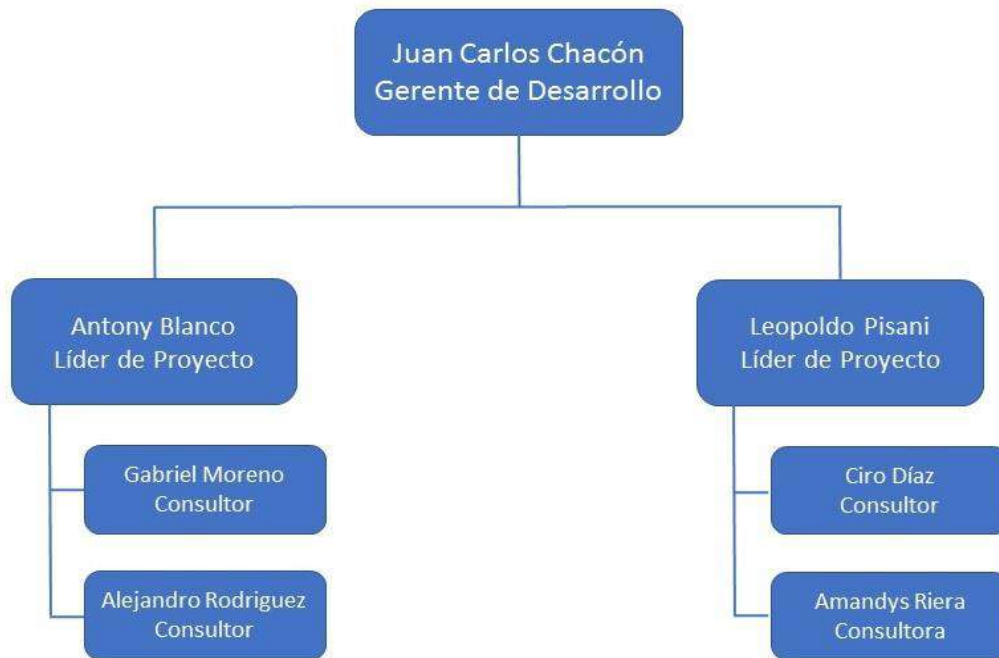


Figura 10. Organigrama Inicial A

Fuente: Grupo CIS de Venezuela (2017).

6.4.1.8. Riesgos Iniciales

1. Existan dificultades de asignación de recursos al proyecto
2. Existan dificultades para añadir horas de servicio para la gestión de la calidad, en los proyectos negociados o que se encuentren en ejecución.

6.4.1.9. Estimación de Costo

La estimación inicial de costos, considerando un (1) recurso a tiempo completo y un (1) recurso a medio tiempo es de Bs. 4.800.000,00.

6.4.1.10. Fechas de Inicio y Fin Requeridas

El proyecto debe ser implementado en sesenta (60) días hábiles comprendidos entre el 01/09/2017 y el 30/11/2017.

6.4.2. Plan de Gestión del Cronograma

6.4.2.1. Cronograma

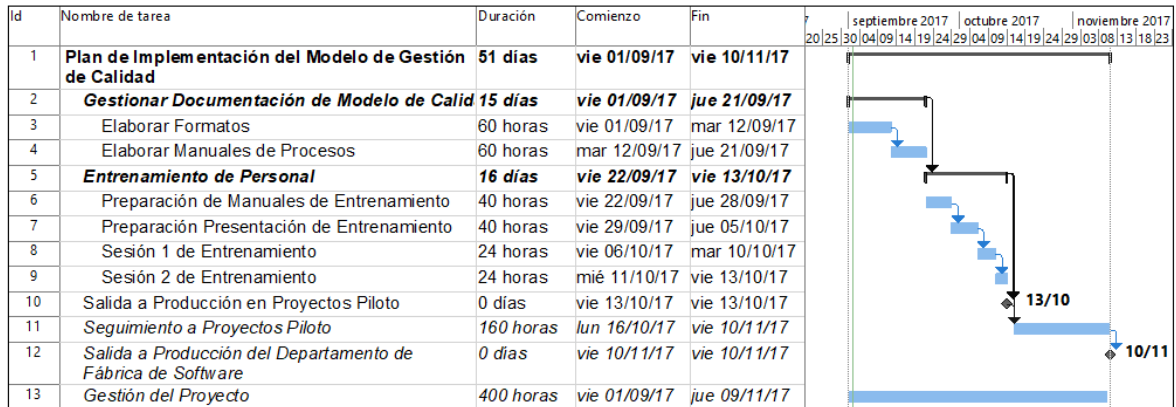


Figura 11. Cronograma del Plan de Implementación

6.4.2.2. Camino Crítico

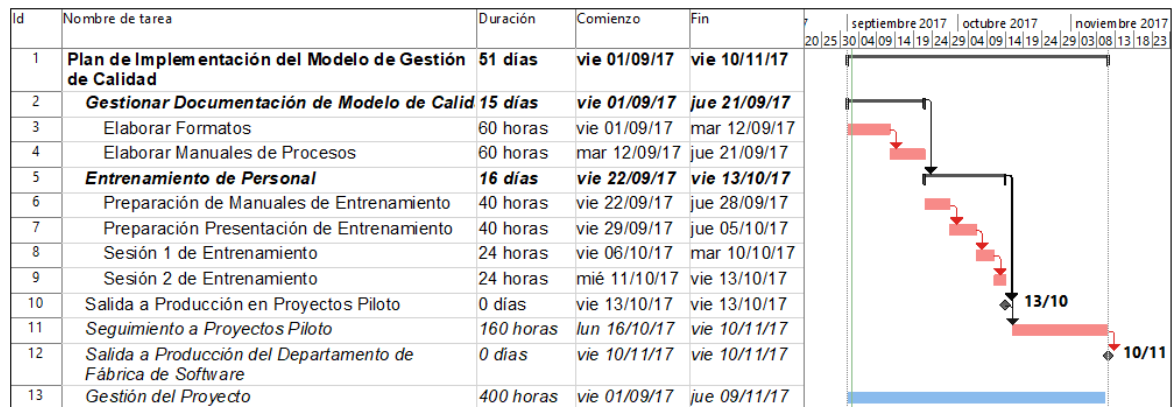


Figura 12. Camino Crítico del Cronograma

6.4.3. Plan de Gestión de Costos

6.4.3.1. Perfil de Progreso Financiero

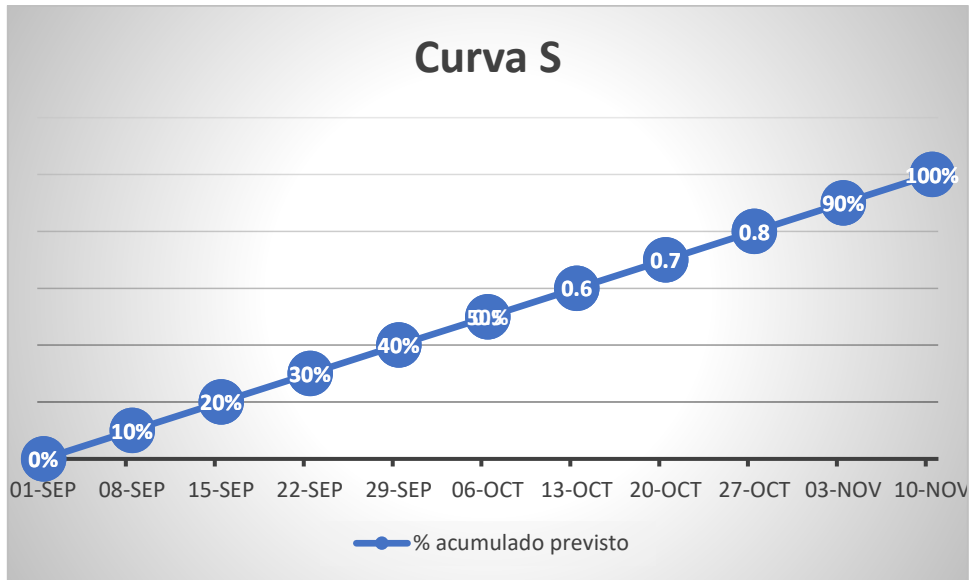


Gráfico 10 – Perfil de Progreso Financiero

6.4.4. Plan de Gestión de Calidad

6.4.4.1. Matriz de Actividades de Calidad

Tabla 40. Matriz de Actividades de Calidad

Paquete de Trabajo	Métrica Utilizada	Actividades de Prevención	Actividades de Control
Gestionar Documentación de Modelo de Calidad	Documento de Requerimientos.	Revisar que los documentos generados cumplan con el Modelo de Calidad establecido.	Validar los documentos generados.
Entrenamiento de Personal	Documento de Requerimientos.	Revisar que el material didáctico cubra el contenido del Modelo de Calidad establecido.	Validar el material didáctico.
Entrenamiento de Personal	Documento de Requerimientos.	Revisar que el entrenamiento cubra el contenido del Modelo de Calidad establecido.	Validar el contenido del entrenamiento.
Aplicación de Modelo en Proyectos Piloto	Documento de Requerimientos.	Revisar que el Modelo de Calidad sea aplicado a los proyectos Piloto	Validar que los proyectos piloto están utilizando el Modelo de Calidad.

6.4.5. Plan de Gestión de Recursos Humanos

6.4.5.1. Histograma de Recursos



Gráfico 11 – Histograma de Recursos

6.4.5.2. Presupuesto de Recursos

Tabla 41. Presupuesto de Recursos

Recurso	Horas	Tarifa x Hora	Total
Gerente de Proyecto	400	Bs. 10.000,00	Bs. 4.000.000,00
Analista de Sistemas	208	Bs. 10.000,00	Bs. 2.080.000,00
Analista de O y M	200	Bs. 10.000,00	Bs. 2.000.000,00
Total	808		Bs. 8.080.000,00

6.4.6. Plan de Gestión de Comunicaciones

Tabla 42. Matriz de Comunicaciones

Nombre Involucrado	Prioridad	Frecuencia	Tipo de Comunicación	Responsable de Elaborar	Forma de Entrega
Gerente de Proyecto	Alta	Semanal	Informe de Seguimiento	Gerente de Proyecto	PDF - Correo Electrónico
Analista de Sistemas	Alta	Diario	Estatus de Actividades	Analista de Sistemas	PDF – Correo Electrónico
Analista de O y M	Alta	Diario	Estatus de Actividades	Analista de O y M	PDF – Correo Electrónico
Gerente de Fábrica de Software	Media	Siempre que sea necesario	Control de Cambio	Gerente de Fábrica de Software	PDF – Correo Electrónico

6.4.7. Plan de Gestión de Riesgos

Tabla 43. Matriz de Riesgos

ID	Descripción	Probabilidad	Impacto	Pxl	Acción para Mitigarlo	Responsable	Fecha	Observaciones
1	Dificultades de asignación de recursos al proyecto	0.20	0.40	0.08	Realizar una solicitud de personal con dos (2) meses de antelación al comienzo del proyecto.	Gerente de Proyecto	02/09/16	
2	Dificultades para añadir horas de servicio para la gestión de la calidad, en los proyectos negociados o que se encuentren en ejecución.	0.30	0.40	0.12	Realizar la gestión del Control de Cambios con el Cliente y la Gerencia de los Proyectos en Ejecución.	Gerente de Proyecto	02/09/16	

6.4.8. Plan de Gestión de Procura

La procura del proyecto se enfocará en la asignación del personal interno, necesario para la ejecución del mismo.

Tabla 44. Matriz de Procura

Perfil	Área	Fecha
Gerente de Proyecto	PMO	01/10/2017
Analista de Sistemas	Fábrica de software	01/10/2017
Analista de O y M	Organización y Métodos	06/10/2017

CAPITULO VII. ANÁLISIS DE RESULTADO

La implementación del Modelo de Calidad propuesto permitirá obtener un producto de calidad, que satisfaga las necesidades del cliente y genere un mayor ingreso en dinero y en confianza hacia la empresa.

Como resultado de la implementación del modelo se verán afectadas las diferentes etapas del ciclo de vida de los proyectos emprendidos por el departamento de Fábrica de Software de Grupo CIS de Venezuela.

A continuación, se detalla la afectación del modelo en cada una de las etapas:

Inicio: Durante la etapa de inicio se realizarán procesos de calidad asociados a la gestión del personal participante.

Planificación: Durante la etapa de Planificación se realizarán actividades relacionadas al Plan de Calidad, Descripción y Revisión de Requisitos, Planificación, Verificación y Validación del Diseño y Desarrollo. Adicionalmente se establecerán los niveles de importancia para las Características y Subcaracterísticas de Calidad, de acuerdo con el proyecto a implementar.

Implementación: Durante la etapa de implementación se realizará la aplicación sistemática de las actividades planificadas relativas a la Calidad, asociadas a los paquetes de trabajo establecidos en la EDT (Figura 9). Asegurando de este modo que el Proyecto utilice todos los procesos necesarios para cumplir con los requisitos de Calidad establecidos.

Control: Durante la etapa control del proyecto se realizarán las mediciones de las Métricas de Calidad Interna, Externa y en Uso, detalladas en las tablas 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38 y 39.

Cierre: Durante la etapa de inicio no se realizarán procesos de calidad.

Para facilitar el impacto de la propuesta para la organización, se muestra a continuación un canvas basado en el Plan de Implementación del Modelo de Gestión de la Calidad:

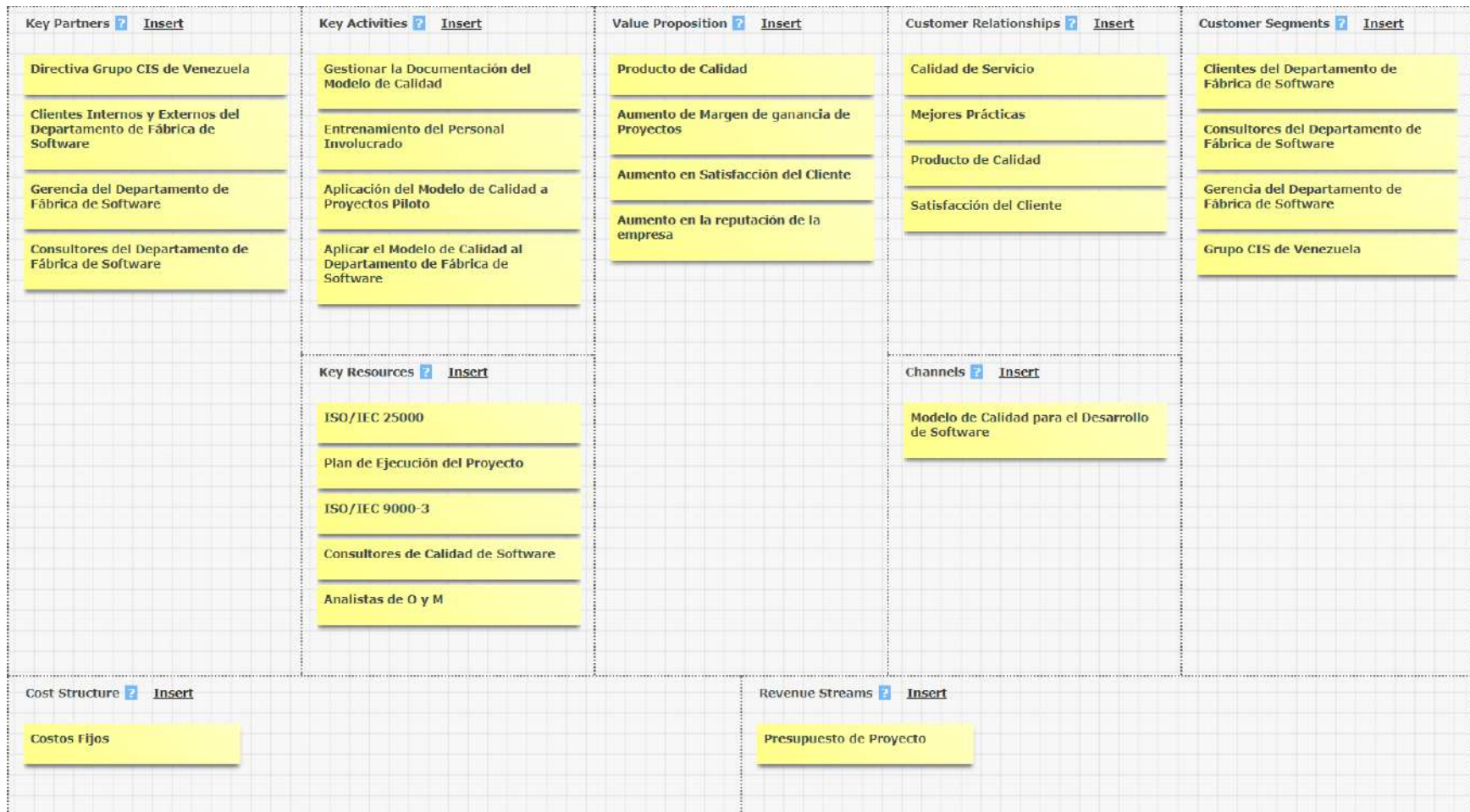


Figura 13. Canvas Modelo de Negocio: Plan de Implementación del Modelo de Gestión de la Calidad

A continuación, se detallan los elementos mencionados en el Canvas Modelo de Negocio – Plan de Implementación del Modelo de Gestión de la Calidad:

Socios Clave: Los socios más importantes para la ejecución del proyecto son la Directiva de la empresa y los relacionados al Departamento de Fábrica de Software, los clientes internos, externos, los consultores y el gerente.

Actividades Clave: las actividades clave comprenden la gestión de documentos y formatos asociados al modelo de calidad a implantar, el entrenamiento del personal involucrado, la ejecución del modelo en proyectos piloto y la implantación final en el Departamento de Fábrica de Software.

Propuestas de Valor: las propuestas de valor que genera el modelo son muchas y entre las más importantes se pueden mencionar: la generación de un producto de calidad, el incremento del margen de ganancia en los proyectos, el aumento de la satisfacción del cliente y por último la mejora en la reputación de la empresa.

Relaciones con Clientes: La relación con los clientes se verá enmarcadas en la calidad de servicio, las mejores prácticas para el desarrollo de software, productos de calidad y la satisfacción del consumidor final.

Segmentos de Cliente: El nicho donde se desarrollará el modelo de negocio planteado será la empresa Grupo CIS de Venezuela, en particular su Junta Directiva y los consultores y gerente del Departamento de Fábrica de Software.

Recursos Clave: los activos más importantes a manejar son: ISO/IEC 25000, el Plan de Ejecución del Proyecto, ISO/IEC 9000-3, el consultor de calidad de software y el analista de Organización y Métodos.

Canales: el canal a través del cual se implementará el negocio es el Modelo de Gestión de la Calidad propuesto.

Estructura de Costos: los costos asociados están representados por los desembolsos fijos del personal interno, externos y los insumos de operación.

Fuentes de Ingresos: los ingresos serán cubiertos por el presupuesto asociado al proyecto.

CAPITULO VIII. LECCIONES APRENDIDAS

El PMI (2013, p.550), establece que las lecciones aprendidas corresponden a “el conocimiento adquirido durante un proyecto el cual muestra cómo se abordaron o deberían abordarse en el futuro los eventos del proyecto, a fin de mejorar el desempeño futuro”. A continuación, se detallan las enseñanzas generadas en las diferentes etapas del proyecto implementado.

7.1. Diccionario de la Estructura Desagregada de Trabajo

Tabla 45. Diccionario de la EDT

ID	Descripción	Entregable	Duración	Costo
1.1.1	Planteamiento de la Investigación	Capítulo I	40h	Bs. 400.000,00
1.1.2	Marco Teórico	Capítulo II	40h	Bs. 400.000,00
1.2.1	Marco Metodológico	Capítulo III	80h	Bs. 800.000,00
1.2.2	Marco Referencial	Capítulo IV	16h	Bs. 160.000,00
1.3.1	Diagnóstico de la Situación Actual	Capítulo V	40h	Bs. 400.000,00
1.3.2	Desarrollo de la Propuesta	Capítulo VI	80h	Bs. 800.000,00
1.4.1	Seguimiento de la Planificación	Cronogramas actualizados, SV, SPI y SV%	8h	Bs. 80.000,00
1.4.2	Seguimiento de Costos	CV, CPI y CV%	8h	Bs. 80.000,00

1.4.3	Control de Calidad	Mediciones del Control de Calidad del TEG	16h	Bs. 160.000,00
1.5.1	Impresión de Documentos	TEG Impreso	2h	Bs. 20.000,00
1.5.2	Presentación/Defensa	Presentación del TEG	8h	Bs. 80.000,00
1.5.3	Aprobación	Aprobación del TEG	1h	Bs. 10.000,00
1.5.4	Solicitud de Solvencia	Solvencia de Aprobación de TEG	1h	Bs. 10.000,00

Los paquetes de trabajo establecidos en el comienzo del proyecto fueron respetados en su mayoría, salvo pequeños ajustes entre el alcance de las actividades. La variación más grande se presentó en el desarrollo de la propuesta, la cual terminó siendo un Plan de Ejecución de proyecto, más que un Modelo de Gestión de la Calidad de Software.

7.2. Factibilidad de La Investigación

a. Cronograma

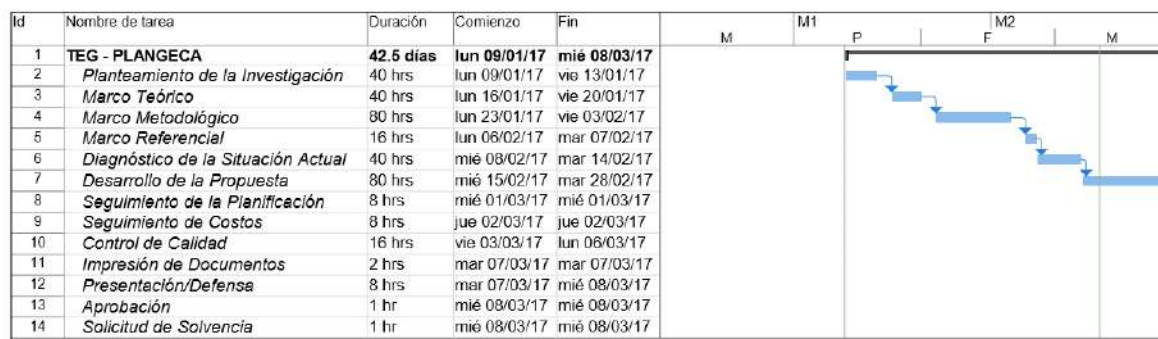
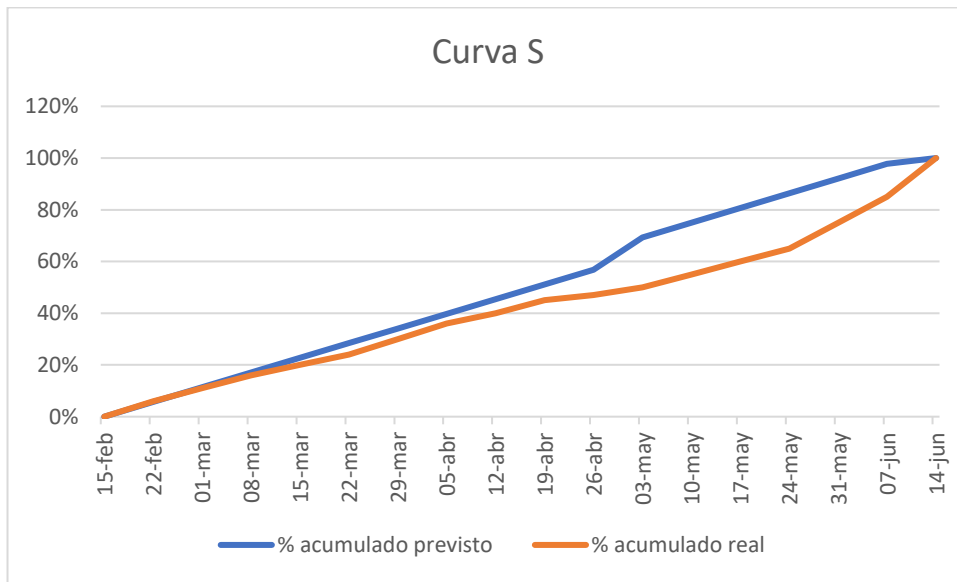


Figura 14. Cronograma Original del Proyecto

El cronograma establecido inicialmente debió ser ajustado a la disponibilidad de los recursos que participaban en el proyecto y en paralelo formaban parte de proyectos que se encontraban en ejecución al momento de realizar la

investigación. La duplicidad de asignaciones provocaba que se les diera prioridad a los proyectos en desarrollo por la Fábrica de Software y en segunda instancia al trabajo de grado. Si bien se respetaron el número de horas presupuestadas, las fechas y horario de ejecución, debieron adaptarse a la disponibilidad de los recursos involucrados.

Gráfico 12 – Curva S de Esfuerzo



En la curva S se pueden apreciar las variaciones de esfuerzo requerido a lo largo del proyecto y básicamente se refleja el incremento de actividades hacia la fecha de finalización. Las actividades de diagnóstico presentaron retrasos motivados por la disponibilidad de recursos y los proyectos a los cuales se encontraban asignados.

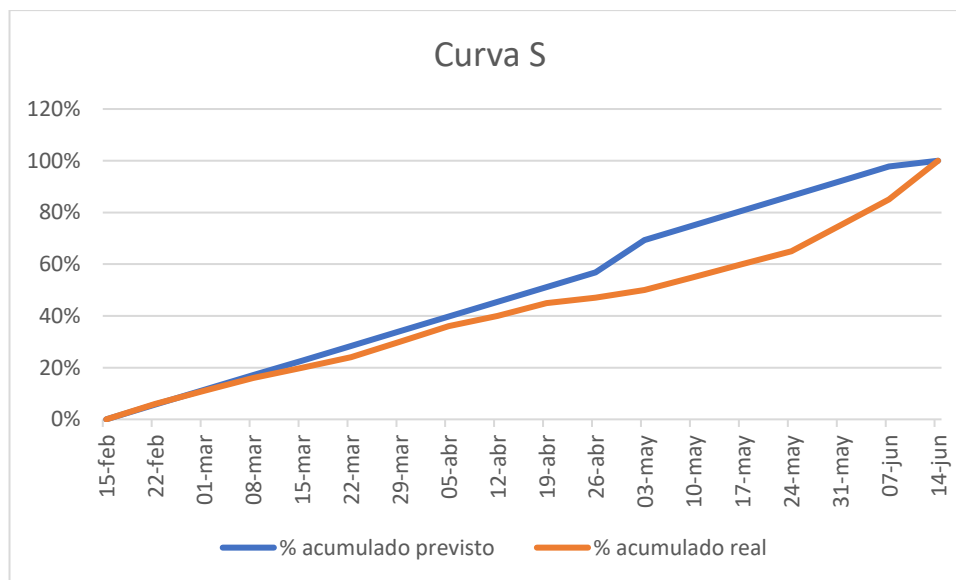
b. Presupuesto de Recursos

Tabla 46. Presupuesto de Recursos

Recurso	Horas	Tarifa x Hora	Total
Gerente de Proyecto	80	Bs. 10.000,00	Bs. 800.000,00
Analista de Sistemas	40	Bs. 10.000,00	Bs. 400.000,00
Analista de O y M	60	Bs. 10.000,00	Bs. 600.000,00
Especialista Metodológico	60	Bs. 10.000,00	Bs. 600.000,00
Especialista en Planificación	60	Bs. 10.000,00	Bs. 600.000,00
Analista de Calidad	40	Bs. 10.000,00	Bs. 400.000,00
Total	340		Bs. 3.400.000,00

El presupuesto en horas fue respetado y los costos, establecidos con fines académicos, se mantuvieron.

Gráfico 13 – Curva S de Recursos del Proyecto



En la gráfica anterior se puede observar cómo se realizó el desembolso del dinero a lo largo del proyecto. A raíz del retraso en la etapa de diagnóstico en la situación actual, los desembolsos más fuertes se realizaron en las dos últimas semanas del proyecto.

CAPITULO IX. CONCLUSIONES Y RECOMENDACIONES

9.1. Conclusiones

La gestión de la calidad de software debe involucrar el aseguramiento de la calidad y el control de la calidad. Al comienzo de la investigación se planteó aplicar un modelo de calidad de producto, y durante la ejecución del proyecto surgió la necesidad de establecer actividades de aseguramiento que generaren la calidad a ser controlada posteriormente.

Entregar un producto de calidad genera un grupo de beneficios que mejoran y optimizan el funcionamiento de la empresa. Se producen mejoras a nivel de costos, tiempos, horas trabajadas, calidad de servicio, aceptación del cliente, reputación de la empresa, entre otros.

Del proyecto ejecutado se lograron cumplir los objetivos establecidos, obteniendo como conclusiones las siguientes:

Objetivo Específico 1. Realizar un análisis situacional, en cuanto a las actividades de calidad de los proyectos e identificar mejoras en las diferentes etapas del Ciclo de Vida de los proyectos: El análisis situacional realizado en la Fábrica de Software de Grupo CIS de Venezuela, permitió conocer las diferentes posibilidades de mejora a nivel de calidad de producto y adicionalmente surgieron actividades de aseguramiento de calidad. Entre los puntos más importantes, se pueden destacar la falta de formalidad en diferentes procesos del desarrollo de software, la ausencia de procedimientos de aseguramiento de calidad y la ausencia de procedimientos de control de calidad.

Objetivo Específico 2: Evaluar los Modelos de Calidad de Software: La evaluación de tres (3) de los modelos más conocidos para la gestión de la calidad

de Software, permitió establecer el modelo que más se adaptaba a la operación del departamento objeto del estudio y cubriera los puntos de mejora definidos. La flexibilidad en la aplicación del modelo de calidad fue una de las características diferenciadora entre los modelos evaluados, y que permitió escoger el modelo a utilizar.

Objetivo Específico 3: Definir un Modelo de Gestión de Calidad para el Ciclo de Vida de los proyectos de desarrollo de software: Una vez analizados los modelos de calidad de producto y los hallazgos obtenidos del análisis situacional, se estableció el modelo a implementar. En base a las necesidades y características particulares de la empresa, el modelo definido abarca actividades de aseguramiento y control de calidad, adicionalmente permite utilizarlo de manera flexible dependiendo del tipo de proyecto a implementar.

Objetivo Específico 4: Diseñar el Plan de Ejecución del Proyecto de la Implementación del Modelo de Gestión de Calidad propuesto: Luego de establecer el modelo a implementar, se estableció el Plan de Implementación que guiará el proceso de adopción del Modelo de Calidad por parte del departamento de Fábrica de Software. El plan propuesto incluye la Gestión de Alcance, Gestión del Tiempo, Gestión de Costos, Gestión de Calidad, Gestión de Recursos Humanos, Gestión de Comunicaciones, Gestión de Riesgos y Gestión de Adquisiciones.

9.2. Recomendaciones

- Al realizar un proyecto de investigación que no se corresponda con la actividad natural de la empresa patrocinante y objeto del estudio, se debe procurar que la asignación de recursos sea exclusiva. Esto permitirá una ejecución ajustada al cronograma del proyecto y un rendimiento acorde de los recursos involucrados. Más importante aún, permitirá no interferir con los proyectos que se encuentren en ejecución.

- Ahondar en el concepto de Calidad del Proceso. Si bien se incluyeron al modelo propuesto actividades de aseguramiento de la calidad, se recomienda realizar una investigación más profunda para verificar la factibilidad de implementar un Modelo de Calidad del Proceso de Desarrollo de Software.
- Realizar jornadas informativas que permitan dar a conocer la importancia de la Calidad en el Desarrollo de Software, a nivel de personal medio y bajo del departamento de Fábrica de Software.
- Tener en cuenta que la implementación de un Modelo de Calidad implica cambios de cultura organizacional, de cultura del departamento, de procesos y nuevas competencias y habilidades que deben ser consideradas.

REFERENCIAS BIBLIOGRÁFICAS

- Arias, F. (2012). El Proyecto de Investigación. Venezuela: Editorial Episteme.
- Baldeón, E. (2015). Método para la evaluación de calidad de software basado en ISO/IEC 25000. Trabajo de Grado de Maestría. Universidad de Sant Martín de Porres. Lima.
- Balseca, E. (2014). Evaluación de calidad de productos de software en empresas de desarrollo de software aplicando la norma ISO/IEC 25000. Trabajo de Grado. Escuela Politécnica Nacional. Quito.
- Barzola, C. y Henríquez, H. (2014). Diseño de una metodología de certificación de productos de software orientado al sector público. Trabajo de Grado de Maestría. Universidad de Sant Martín de Porres. Lima.
- Basso, D. (2014). Propuesta de Métricas para Proyectos de Explotación de Información. Trabajo de Grado de Maestría. Universidad Tecnológica Nacional. Buenos Aires.
- Berlinches, A. (2002) Calidad. España: S.A. EDICIONES PARANINFO.
- Bernal, C. (2010). Metodología de la Investigación. Colombia: Prentice Hall.
- Chamoun, Y. (2002). Administración profesional de proyectos: la guía/Yamal Chamoun. México: Mc Graw Hill.
- Colegio de Ingenieros de Venezuela. (2017). Código de Ética. Caracas: Colegio de Ingenieros de Venezuela. Recuperado de www.civ.net.ve/uploaded_pdf/cep.pdf.
- Crosby, P. (1979). Quality is free. Nueva York: McGraw Hill.
- Delgado, M. (2014). Elaboración de un modelo de gestión de riesgos basado en la metodología de gestión de proyectos para el proyecto de implementación de software financiero, fiduciario y de inversión en la empresa Gestor iNC, C.A. Trabajo de Grado de Maestría. Universidad de las Fuerzas Armadas. Sangolqui.
- DEMING, W. E. (1989). Calidad, Productividad y Competitividad la Salida de la Crisis (1a. ed.). Madrid: Díaz De Santos.
- Diccionario de informática (1993). Ediciones Díaz de Santos. Madrid: 2ª edición.
- FEIGENBAUM, A. V. (2009). Control Total de la Calidad (3a. ed., 11a. reimp.). Ciudad Juarez (Chihuahua): Grupo Editorial Patria.

Freedman, A. (1984). Glosario de computación. ¡Mucho más que un glosario! México: McGraw Hill (1ª edición).

Grupo CIS de Venezuela. (2017). Reseña Histórica. Grupo CIS de Venezuela. Recuperado de www.grupocis.net.

IEEE (1990). IEEE Standard Glossary of Software Engineering Terminology. IEEE Std.

ISO (2008) ISO/IEC 9001 Quality management systems – Requirements, International Organization for Standardization.

ISO (2001) ISO/IEC 9126 Software engineering — Product quality, International Organization for Standardization.

ISO (2001) ISO/IEC 14598-6 Software engineering -- Product evaluation -- Part 6: Documentation of evaluation modules, International Organization for Standardization.

ISO (2014) ISO/IEC 25000 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE, International Organization for Standardization.

ISO (2014) ISO/IEC 25001 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Planning and management, International Organization for Standardization.

ISO (2011) ISO/IEC 25010 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models, International Organization for Standardization.

ISO (2008) ISO/IEC 25012 Software engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- Data quality model, International Organization for Standardization.

ISO (2007) ISO 25020 Software and system engineering--Software Product Quality Requirements and Evaluation (SQuaRE)--Measurement reference model and guide, International Organization for Standardization.

ISO (2012) ISO 25021 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Quality measure elements, International Organization for Standardization.

ISO (2016) ISO 25022 Systems and software engineering -- Systems and software quality requirements and evaluation (SQuaRE) -- Measurement of quality in use, International Organization for Standardization.

ISO (2016) ISO 25023 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Measurement of system and software product quality, International Organization for Standardization.

ISO (2016) ISO 25024 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Measurement of data quality, International Organization for Standardization.

ISO (2007) ISO 25030 Software engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- Quality requirements, International Organization for Standardization.

ISO (2011) ISO 25040 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Evaluation reference model and guide, International Organization for Standardization.

ISO (2011) ISO 25041 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Evaluation guide for developers, acquirers and independent evaluators, International Organization for Standardization.

ISO (2011) ISO 25042 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Evaluation modules, International Organization for Standardization.

ISO (2011) ISO 25045 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Evaluation module for recoverability, International Organization for Standardization.

ISO (2005) ISO/IEC 27001 Information technology - Security techniques - Information security management systems - Requirements, International Organization for Standardization.

Lledó, P (2011). Director de proyectos: como aprobar el examen PMP® sin morir en el intento. Victoria, BC, Canadá: el autor.

Meza, W (2014). Marco de Trabajo para la Gestión de la Calidad en Proyectos de Desarrollo de software basado en PMBOK y CMMI DEV. Trabajo de Grado de Maestría. Escuela Politécnica Nacional. Quito.

Office of Government Commerce (2009). Managing successful projects with PRINCE2 (5th edición). Reino Unido.

Pressman, R (1998). INGENIERÍA DEL SOFTWARE UN ENFOQUE PRACTICO. España: McGraw Hill.

Project Management Institute. (2013). A guide to the project management body of knowledge (PMBOK guide). Newtown Square, Pa: Project Management Institute.

Project Management Institute. (2017). Código de Ética. Newtown Square: Project Management Institute. Recuperado de https://www.pmi.org/-/media/pmi/documents/public/pdf/ethics/pmi-code-of-ethics.pdf?sc_lang_temp=es-ES

Ramos, D (2016). Diseño de un Modelo de Evaluación de la Calidad de Productos de Software, Basado en Métricas Externas y Usabilidad Aplicado a un Caso de Estudio. Trabajo de Grado de Maestría. Escuela Politécnica Nacional. Quito.

Sabino, C. (1992). El Proceso de Investigación. Caracas: Panapo.

Sampieri, R. y Fernández, C. y Baptista, P. (2006). Metodología de la Investigación. México: McGraw Hill.