



MAESTRÍA EN GERENCIA DE SISTEMAS Y
TECNOLOGÍAS DE INFORMACIÓN
FACULTAD DE POSGRADOS

ADAPTACIÓN DE MARCOS DE REFERENCIA DE CALIDAD
A LA INDUSTRIA DE DESARROLLO DE SOFTWARE
EN ECUADOR

“Trabajo de Titulación presentado en conformidad a los requisitos establecidos
para optar por el título de Magister en Gerencia de Sistemas y
Tecnologías de Información”

Profesor guía

Dip. Manolo Giovanni Sangoquiza Socasi

Autor

Ing. Paula Natalia Cadena Loayza

Año

2015

DECLARACIÓN PROFESOR GUÍA

“Declaro haber dirigido este trabajo a través de reuniones periódicas con el estudiante, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”

Manolo Giovani, Sangoquiza Socasi

Diplomado

1711931566

DECLARACIÓN DE AUTORÍA DEL MAESTRANTE

“Declaro que este trabajo es original de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.”

Paula Natalia, Cadena Loayza

Ingeniera

1719567149

AGRADECIMIENTOS

Agradezco en primer lugar a Dios por haberme permitido cumplir con una más de mis metas que me he propuesto a lo largo de mi vida, y que me ha guiado en el transcurso de mi maestría.

A mis padres y hermanos quienes me han apoyado en la elaboración de este trabajo y durante toda mi vida estudiantil.

A mi enamorado y amigo, Daniel Jarrín, quién ha estado a mi lado durante toda la elaboración de la tesis, y me ha brindado su apoyo y ayuda en momentos difíciles de mi carrera estudiantil.

A mis queridos amigos y compañeros Julio Nuñez y Diana Casanova quienes me han sabido incentivar y ayudar cuando me he sentido derrotada.

A la Universidad de las Américas, en especial a nuestros profesores, quienes nos han formado en el transcurso de estos años, proporcionándonos grandes conocimientos, que sin duda, nos han servido y nos servirán en el futuro. Un profundo agradecimiento a mi tutor y corrector del trabajo de titulación.

DEDICATORIA

A toda mi familia que me ha brindado su apoyo y ha estado a mi lado en el transcurso de mi carrera. A mis padres que en cada momento difícil de mi carrera me dieron su apoyo incondicional y me dieron fuerzas para no dejarme derrotar. A todos aquellos que en algún momento de mi vida me brindaron su ayuda, amistad y comprensión.

RESUMEN

En el Ecuador, los requerimientos de negocio en los proyectos de software cambian constantemente por lo que las exigencias de calidad son cada vez mayores. El aseguramiento de calidad es algo esencial en cualquier empresa generadora de productos, ya que estos van a ser utilizados por terceras personas, y siempre se espera que sean de la mejor calidad y se ajusten a las necesidades del cliente.

Los estándares de calidad generan a las empresas una gran ventaja frente a sus competidores y permite que las mismas puedan realizar una mejora continua en la calidad de los procesos, servicios y productos de software. Actualmente algunas empresas de desarrollo de software presentan problemas al momento de realizar el control de calidad de sus productos, ocasionando que la empresa tenga que gestionar nuevamente el proyecto y se tenga que invertir altos costos en la mejora de estos problemas de software.

El presente proyecto tiene como propósito adaptar, a la realidad ecuatoriana, los estándares y marcos de referencia ya definidos en las empresas de desarrollo de software en base a un análisis de los principales estándares, procedimientos y métodos para el aseguramiento de la calidad utilizados en las normas ISO y CMMI.

ABSTRACT

In Ecuador, the business requirements for software projects change constantly and for that reason the need for quality is becoming more and more demanding. Quality assurance is essential in any product development enterprise. Since the products will be used by a third person it is always expected that they provide a good quality standard and that they adjust to the customer's needs.

Setting quality standards generate a great advantage for companies against their competitors and allows them to implement continuous quality improvements for their processes, services and software products. These days some software development companies have problems at the moment of conducting the quality assurance control of their products, causing the company the need to re-work a project and to spend high amounts of money to fix software issues.

The purpose of this project is to adapt the standards and guidelines to the reality of Ecuadorian software development business, based on an analysis of the main industry standards, procedures and methods used for quality assurance in the ISO and CMMI norms.

ÍNDICE

INTRODUCCIÓN	1
OBJETIVOS	2
OBJETIVOS GENERALES	2
OBJETIVOS ESPECÍFICOS	2
1. Capítulo I. La Industria de Software en el Ecuador	3
1.1 Exportación por Regiones	4
1.2 PYMES en el Ecuador	4
1.3 Empresas de Desarrollo de Software en el Ecuador	6
1.4 Ingresos totales del sector de Software en el Ecuador	6
1.5 Estándares de Calidad en el Ecuador	7
2. Capítulo II. Sistemas de Gestión de Calidad en el Desarrollo de Software	9
2.1 Gestión de la Calidad del Software	10
2.2 Beneficios de los Sistemas de Calidad	11
2.2.1 Punto de Vista Interno	11
2.2.2 Punto de Vista Externo	11
2.3 Calidad del Software	12
2.3.1 Modelos y/o Estándares de Calidad	13
2.3.2 Calidad del Producto y del Proceso	14
3. Capítulo III. Análisis Crítico de los Principales Marcos de Referencia y Estándares de Calidad existentes para el Desarrollo de Software	16

3.1 Antecedentes.....	16
3.2 Los Marcos de Referencia y su Aplicabilidad Universal	17
3.3 Fundamentos principales de los Marcos de Referencia en el Ciclo de Vida del Software	18
3.4 Características de las Normas ISO y CMMI.....	21
3.4.1 Normas ISO para el desarrollo de software	21
3.4.2 CMMI.....	25
3.5 Comparativa de Normas ISO (9001:2008, 9000-3:2004) y CMMI en el control de calidad del software (Ventajas, desventajas y características)	29
3.5.1 Ámbito comparativo	29
3.5.2 Estudio de las características	30
3.5.3 Ventajas y desventajas	31
4. Capítulo IV. Adaptación y Propuesta de un nuevo Marco de Referencia de Calidad para el Desarrollo de Software.....	35
4.1 Lineamientos del marco de referencia	36
4.2 Estructura del marco de referencia propuesto	37
4.2.1 Obtención de calidad en el producto a desarrollar	38
4.2.2 Descripción de las fases propuestas	39
4.2.2.1 Levantamiento de la necesidad.....	39
4.2.2.2 Estructura técnica del software.....	52
4.2.2.3 Modelamiento y diseño.....	55
4.2.2.4 Diseño de la implantación	59
4.2.2.5 Procesos de evaluación del cumplimiento	62

4.2.2.6 Mejora Continua (Retroalimentación)	68
4.3 Planificación integral de desarrollo	68
5. Capítulo V. Conclusiones y Recomendaciones	70
5.1 Conclusiones	70
5.2 Recomendaciones	72
GLOSARIO Y LISTA DE ABREVIATURAS.....	74
REFERENCIAS.....	77

ÍNDICE DE TABLAS

Tabla 1. PYMES en el Ecuador	5
Tabla 2. Empresas ecuatorianas en el área de Software por ciudad.....	8
Tabla 3. Familiaridad y Estado de Uso de estándares de calidad de las empresas ecuatorianas desarrolladoras de software.....	8
Tabla 4. Fundamentos de los marcos de referencia en el desarrollo de software.....	19
Tabla 5. Normas Internacionales ISO.....	23
Tabla 6. Características de los Marcos de Referencia Estudiados.....	30
Tabla 7. Cuadro de semejanzas y diferencias.....	34
Tabla 8. Descripción de la fase de levantamiento de la necesidad	40
Tabla 9. Delimitación del entorno	41
Tabla 10. Matriz para expresión de datos	46
Tabla 11. Tabla de interpretación de los resultados.....	48
Tabla 12. Cálculo de rangos por variable	48
Tabla 13. Cálculo de relación entre variables.....	51
Tabla 14. Estructura técnica del software.....	52
Tabla 15. Organización para el desarrollo del software	54
Tabla 16. Estructura modular.....	56
Tabla 17. Codificación de objetos en el software	57
Tabla 18. Nomenclatura de relación.....	58
Tabla 19. Levantamiento de recursos disponibles	60
Tabla 20. Formulario de registro de prueba del software.....	65
Tabla 21. Proceso de evaluación de funcionamiento.....	66
Tabla 22. Indicadores.....	67
Tabla 23. Modelo de planificación	69

ÍNDICE DE GRÁFICOS

Gráfico 1. Exportación por Regiones	4
Gráfico 2. Empresas de Desarrollo de Software en el Ecuador.....	6
Gráfico 3. Ingresos totales por venta del sector de Software en el Ecuador	7
Gráfico 4. Componentes de la Gestión de la Calidad Total	10
Gráfico 5. Estructura de un Modelo de Calidad del Software	13
Gráfico 6. Calidad del Producto y del Proceso.....	15
Gráfico 7. Cobertura de los fundamentos de los marcos de referencia en el desarrollo de software.....	20
Gráfico 8. Modelo de un sistema de gestión de calidad basado en procesos.....	24
Gráfico 9. Cinco niveles de los procesos de madurez CMMI.....	27
Gráfico 10. Lineamientos del marco de referencia.....	36
Gráfico 11. Estructura del marco de referencia propuesto.....	38
Gráfico 12. Macro procesos.....	43
Gráfico 13. Conceptos de diagramas	44
Gráfico 14. Diagramación de procesos	45
Gráfico 15. Procesos de implantación del software	59
Gráfico 16. Medición de la satisfacción del cliente.....	63

INTRODUCCIÓN

Actualmente las empresas de desarrollo de software están formadas principalmente por MIPYMES (micro, pequeñas y medianas empresas) en donde la calidad en el diseño y desarrollo software, es una parte esencial para mejorar el rendimiento de los procesos y satisfacer los requerimientos de los usuarios.

En nuestro país la mayoría de empresas dedicadas al sector de desarrollo de software conocen la existencia de normas de calidad internacionales, sin embargo, el establecimiento de modelos de control de calidad en MIPYMES es difícil debido a que requiere de grandes inversiones de dinero, tiempo y recursos que no están al alcance de las empresas, por lo cual en base a las necesidades han creado sus propios métodos que han sido desarrollados en base a su línea de negocio y a la experiencia adquirida en el transcurso del tiempo, lo que repercute en que la calidad de sus productos no cumpla con las expectativas del cliente y esto involucre que la empresa tenga que gestionar y corregir el producto desarrollado dedicando tiempo y dinero en la mejora de estos problemas de software.

Con el fin de ayudar a las empresas a utilizar las buenas prácticas en desarrollo de software, se han creado varios estándares, en respuesta a las necesidades del mercado actual. En el presente trabajo se realizará una comparación de dos sistemas de calidad de desarrollo de software: La norma ISO 9001 y el Modelo de Madurez de Capacidad Integral (CMMI). Se unificará criterios para la adaptabilidad de estándares, metodologías y procesos de calidad con los métodos que han ido desarrollando las micro, pequeñas y medianas empresas de desarrollo de software ecuatoriano.

Mediante la recolección y análisis de marcos de referencia existentes, se planteará una propuesta de mejora que se adapte al tamaño y tipo de negocio que tienen las MIPYMES Ecuatorianas, para esto se desarrollará iniciativas de integración de las metodologías y procedimientos que se utilizan en las normas de calidad, buscando mejorar la calidad de sus productos a través de la mejora de sus procesos, lo cual permitirá fortalecer la producción de software en nuestro país haciendo que tenga un nivel competitivo frente a otros países.

OBJETIVOS

OBJETIVOS GENERALES

Desarrollar una propuesta para un marco de referencia adaptable al ambiente de trabajo ecuatoriano en base a un análisis de los principales estándares, procedimientos y métodos para el aseguramiento de la calidad utilizados en las normas ISO y CMMI.

OBJETIVOS ESPECÍFICOS

- Recolectar y analizar los marcos de referencia que se utilizan actualmente en empresas de desarrollo de software ecuatorianas.
- Identificar y comparar los principales estándares de control de calidad para la gestión del desarrollo de software.
- Elaborar la propuesta del nuevo marco de referencia para el control de calidad en el Ecuador.

Capítulo I

1. La Industria de Software en el Ecuador

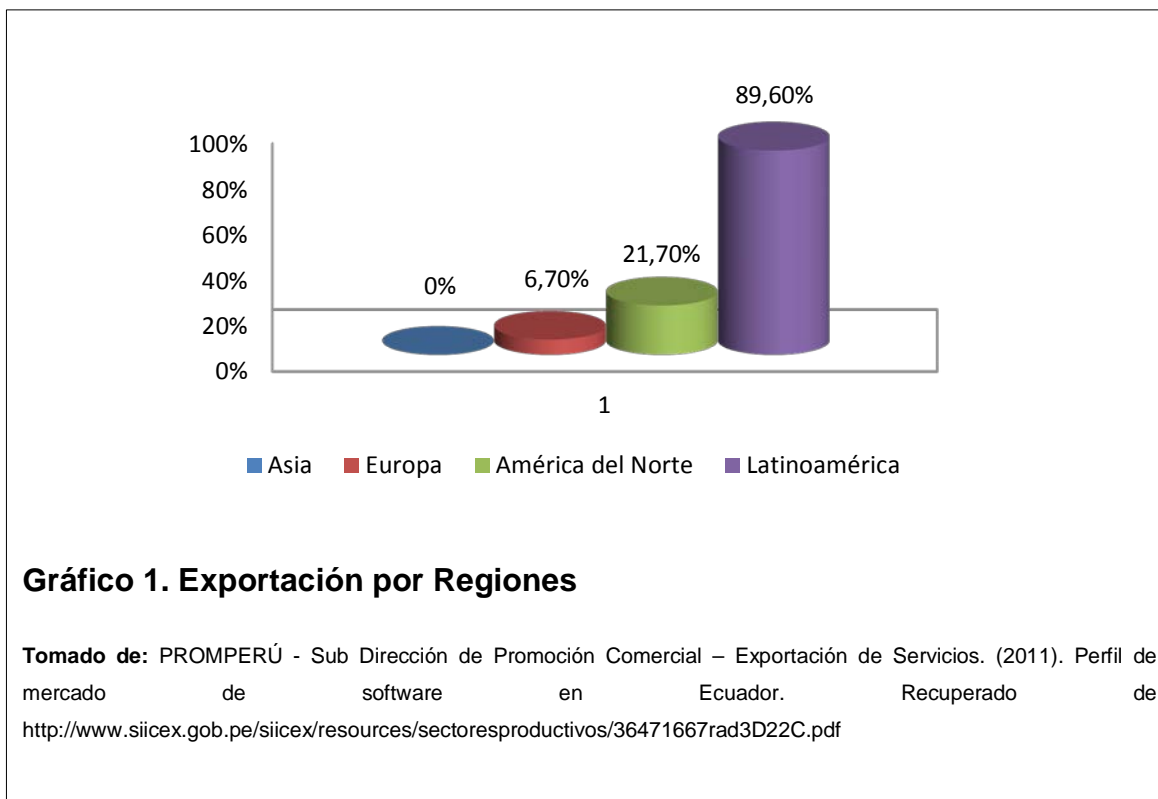
En el Ecuador, la Asociación Ecuatoriana de Software, AESOFT es una organización privada sin fines de lucro creada en mayo de 1995 en Quito, para monitorear y apoyar a las empresas pertenecientes a la industria de tecnologías de información y comunicación, a fomentar el crecimiento e innovación de productos tecnológicos de calidad.

Según Sánchez, Sandra, (2012), el desarrollo de software dentro de nuestro país ha aportado en la generación de empleos y gradualmente en la mejora de la productividad económica del país. Los estándares utilizados para el control de calidad deben estar adaptados a la cultura y a los sectores donde se apliquen.

“La industria de software representa una actividad económica de suma importancia para todos los países del mundo. La producción de software y la prestación de servicios relacionados son actividades económicas cada vez más importantes por ser inyección para el aumento de productividad del país, más su potencial para generar empleos calificados y divisas por exportaciones. En Ecuador, la industria de software va en aumento progresivamente y tiene un alto potencial de exportación (...) Para viabilizar que las micro, pequeñas y medianas empresas de la industria de software ecuatoriana utilicen estándares y procesos formales para desarrollo de software que mejoren la calidad de sus productos, es necesario adecuar dichos estándares y procesos a las especificidades de este sector.” (Sánchez, 2012)

1.1 Exportación por Regiones

Actualmente el software que se realiza en nuestro país cuenta con un alto potencial de exportación, como el desarrollado para la industria bancaria, farmacéutica, entre otras. Países de Latinoamérica, Estados Unidos y Europa son clientes potenciales que pueden apoyar el crecimiento de la industria, debiendo para ello generar productos altamente competitivos e innovadores.



1.2 PYMES en el Ecuador

En el Ecuador, las micro, pequeñas, medianas y grandes empresas, se clasifican utilizando tres variables principales: personal, valor bruto de ventas anuales, y total de activos.

A continuación se presenta una tabla donde se podrá apreciar de mejor manera la clasificación de las PYMES en nuestro país.

Tabla 1. PYMES en el Ecuador

Variabes	Micro Empresa	Pequeña Empresa	Mediana Empresa	Grandes Empresas
Personal Ocupado	1 - 9	10 - 49	50 - 199	> 200
Valor Bruto de las ventas anuales	< 100.000	100.001 - 1'000.000	1'000.001 - 5'000.000	> 5'000.000
Montos Activos	Hasta US \$100.000	De US \$ 100.001 hasta US \$ 750.000	De US \$ 750.001 hasta US \$ 3'999.999	> US \$ 4'000.000

Tomado de: Solines Chacón, P. (2010). Clasificación de las PYMES, de acuerdo a la Normativa implantada por la Comunidad Andina en su Resolución 1260 y la legislación interna vigente. Recuperado de Consultores de Negocios con una Perspectiva Internacional - Russell Bedford Ecuador S.A.: <http://www.russellbedford.com.ec/images/Boletines%202010/12.%20Resolucion%20SUPER%20CIAS%20PYMES%20-%20SC-INPA-UA-G-10-005.pdf>

De acuerdo a este estudio se estima que por lo menos el 90% de las empresas de desarrollo de software corresponden a MIPYMES (micro, pequeñas y medianas empresas).

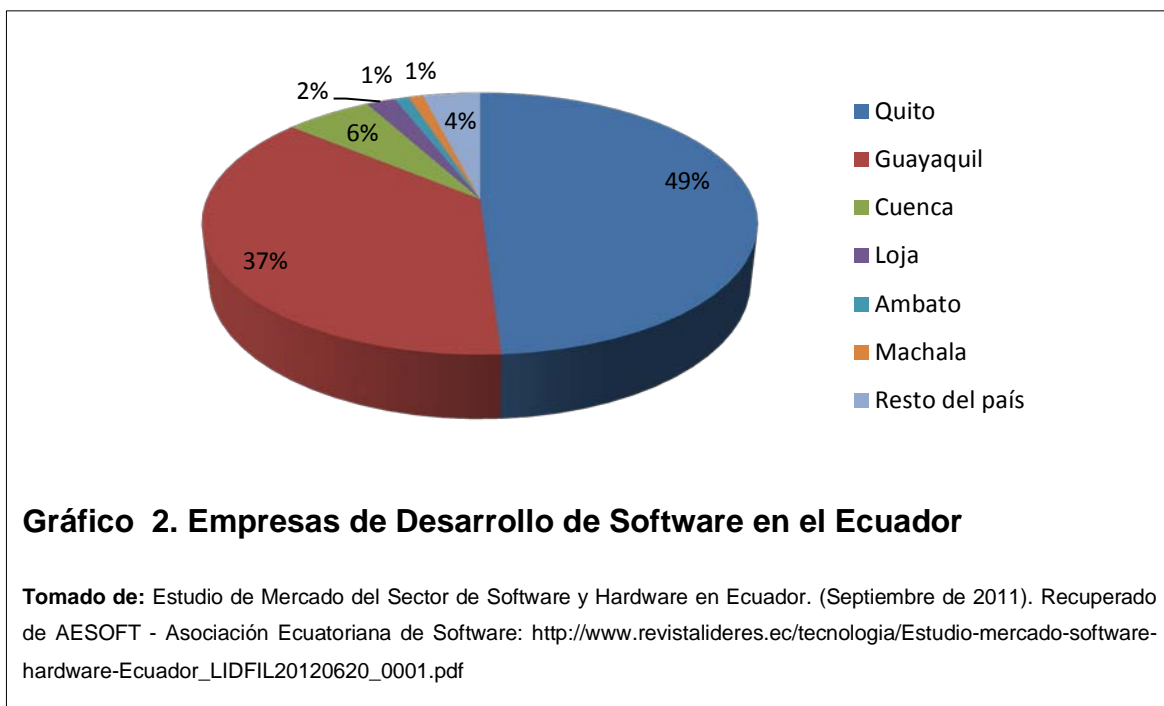
Según un estudio de la Superintendencia de Compañías, en el año 2011

- 633 empresas se dedicaban al desarrollo de software.
- 277 eran micro empresas, 117 pequeñas empresas, 16 medianas empresas y un pequeño número de 7 eran las consideradas grandes empresas.

Actualmente, las empresas dedicadas al desarrollo de software se distribuyen en los siguientes sectores: administrativo, bancario, educativo, salud y contable, aun cuando las cifras disponibles no son exactas, conociéndose que su desarrollo cubre más segmentos de mercado.

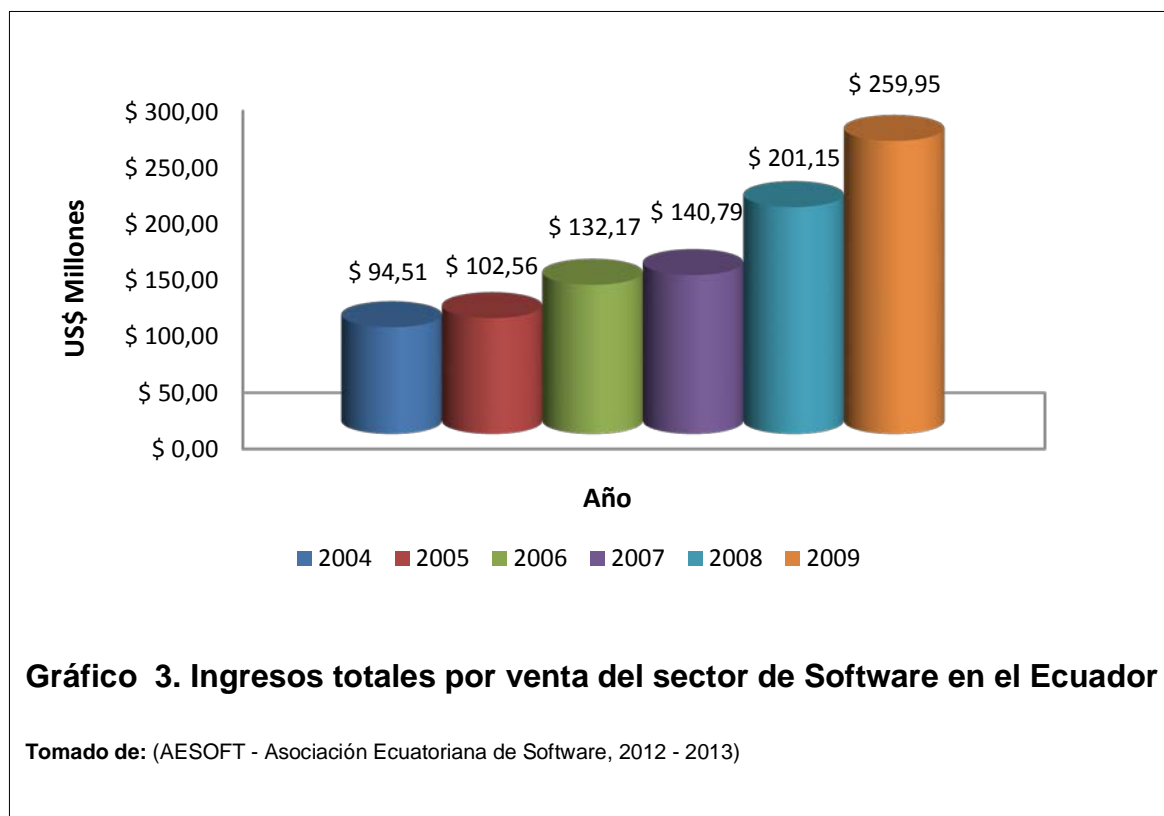
1.3 Empresas de Desarrollo de Software en el Ecuador

Quito se encuentra en un 49%, representado a 319 empresas; Guayaquil se encuentra en un 37%, representando a 240 empresas; Cuenca se encuentra en un 6%, representando a 42 empresas. El 8% restante representa a 50 empresas de desarrollo de software en el resto del país.



1.4 Ingresos totales del sector de Software en el Ecuador

Según un estudio realizado por el SRI, el sector de software ha alcanzado una importante evolución entre 2004 y 2009 de US\$ 95 millones a US\$ 260 millones con una tasa de crecimiento anual compuesta de 22,4%. Este comportamiento, permite calificar a la industria como emergente con un alto potencial de generar un importante impacto en el Producto Interno Bruto Nacional, si mantiene los niveles de crecimiento existentes.



El software señalado en el Gráfico 3 se entiende desarrollado en Ecuador, lo que indica que no es importado.

1.5 Estándares de Calidad en el Ecuador

Entre el año 2003 - 2004 un estudio realizado por Salazar, D., Villavicencio, M., Macías, M., & Snoeck, M. con una muestra de 77 empresas desarrolladoras de software de Quito, Guayaquil y Cuenca, se realizó un análisis de la familiaridad con los estándares de calidad. El 94.8% de las empresas encuestadas afirmó conocer sobre la ISO 9001, el 48% acerca de MSF y el 29.8% conocía sobre CMM, únicamente el 36.3% de las empresas utilizan estándares de calidad en el desarrollo de software, de la muestra sólo el 24.6% utilizaba estándares con reconocimiento internacional y el 37.6% estaba analizando la probabilidad de implantar algún estándar en la empresa.

Tabla 2. Empresas ecuatorianas en el área de Software por ciudad

Ciudad	Empresas Estudiadas	Empresas Pequeñas	Empresas Medianas	Empresas Grandes
Guayaquil	13	8	4	1
Quito	47	19	24	4
Cuenca	17	15	2	0
TOTAL	77	42	30	5

Tomado de: Salazar, D., Villavicencio, M., Macías, M., & Snoeck, M. (2003 - 2004). Estudio estadístico exploratorio de las empresas desarrolladoras de software asentadas en Guayaquil, Quito y Cuenca. Recuperado de <https://www.fiec.espol.edu.ec/resources/investigacion/articulo90.pdf>

Tabla 3. Familiaridad y Estado de Uso de estándares de calidad de las empresas ecuatorianas desarrolladoras de software

Estándar	Empresas	Empresas Pequeñas	Empresas Medianas	Empresas Grandes
ISO 9001	73 (94.8%)	39	30	4
CMM	23 (29.8%)	7	13	3
MSF	37 (48%)	18	16	3
Utiliza estándar	28 (36.3%)	12	13	3
Utilizó estándar	2 (2.5%)	0	2	0
Posibilidad de uso de estándar	29 (37.6%)	16	11	2
En el pasado hubo la posibilidad de uso de estándar	3 (3.8%)	2	1	0
Nunca han utilizado estándar	18 (23.3%)	12	6	0

Tomado de: (Salazar, Villavicencio, Macías, & Snoeck, 2003 - 2004)

Capítulo II

2. Sistemas de Gestión de Calidad en el Desarrollo de Software

Los Sistemas de Gestión de la Calidad son normas y estándares internacionales cuya aplicación permite mejorar la calidad de bienes y servicios producidos, apoyados en parámetros auditables, los cuales pueden ser sujetos a cualquier tipo de validación.

Con el objetivo de aumentar la productividad en la implementación de aplicaciones con eficiencia y calidad, las empresas de desarrollo de software buscan hacer sus productos bajo modelos y estándares internacionales que sean capaces de establecer márgenes de competencia basados en la mejora de procesos. Estos estándares de calidad se los determina en base a los objetivos que se pretenda alcanzar, sin embargo, se pueden aplicar metodologías genéricas como la ISO 9001 o CMMI, dependiendo de cada caso.

Es importante el compromiso de la alta gerencia en hacer del conocimiento y cambio de mentalidad al personal de la organización, de que es indispensable implantar un sistema de gestión de calidad, debido a la necesidad de mejorar los procesos internos del desarrollo de software que tiene actualmente la organización. Los altos mandos deben tomar en cuenta la cultura de la organización y la adaptabilidad al cambio, dado que su objetivo es generar altos índices de ganancia y obtener un certificado o reconocimiento internacional en la industria del software, esto mejorará las buenas prácticas en el desarrollo de software y el cliente obtendrá productos de calidad.

Para la implantación de un sistema de gestión de calidad se necesita que todas las áreas de la organización estén comprometidas y que exista una rígida supervisión del avance de la implantación en la organización, adicionalmente, se debe conocer detalladamente el estándar como los beneficios, experiencias, riesgos y limitaciones, ser capaz de evaluarlo, realizar un análisis de la evolución que se pretende dar al estándar o modelo, saber cómo obtener métricas y

técnicas que permitan evaluar la etapa o estado de la implantación del nuevo sistema de calidad.

Además de lo anterior, el seleccionar un estándar dependerá de los recursos y el tamaño de la organización, también es importante evitar que la implantación del sistema de calidad de como resultado papelería y burocracia excesiva que complicará la entrega del producto.

2.1 Gestión de la Calidad del Software

Existen diversos enfoques de gestión de la calidad, desde una simple inspección hasta la gestión de la calidad total.

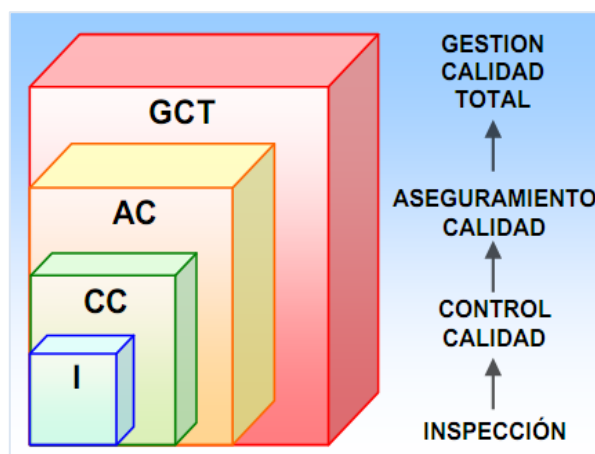


Gráfico 4. Componentes de la Gestión de la Calidad Total

Tomado de: López Bustamante, G., & Flores Vera, J. (26 de 05 de 2010). Sistema de Gestión de Calidad. Recuperado de [https://www.dspace.espol.edu.ec/bitstream/123456789/10181/1/Sistema de Gestión de la Calidad.ppt](https://www.dspace.espol.edu.ec/bitstream/123456789/10181/1/Sistema%20de%20Gesti3n%20de%20la%20Calidad.ppt)

La Gestión de la Calidad del Software, está integrada por cuatro componentes, los cuales interactúan en un ciclo continuo de retroalimentación, contribuyendo al mejoramiento continuo de la empresa, estos son:

- **Planificación de la Calidad del Software:** se establece los objetivos de calidad y la especificación de los procesos.

- **Control de Calidad del Software:** son pruebas que permiten detectar defectos en el software.
- **Aseguramiento de la Calidad del Software:** se evalúa las características del software en base a su cobertura, capacidad de servicio y satisfacción de las necesidades del cliente. Aspectos que van inmersos en los procesos utilizados para su desarrollo.
- **Mejora de la Calidad del Software:** permite auditar el cumplimiento de las características del software para determinar las posibles mejoras y/o correcciones.

2.2 Beneficios de los Sistemas de Calidad

Los beneficios que se obtienen de la definición, desarrollo e implantación de un Sistema de Gestión de Calidad son los siguientes:

2.2.1 Punto de Vista Interno

- Mejor calidad de productos y servicios debido a procesos más eficientes.
- Mejor diseño del producto.
- Mayor productividad debido a la eficiente mano de obra.
- Eliminación de cuellos de botella en la producción de software.
- Creación de la conciencia con respecto a la calidad, mejora de la cultura de calidad, dando como resultado una mayor satisfacción laboral de los empleados en el trabajo.
- Disminución de costos de no calidad.
- Crecimiento de ingresos por el incremento de clientes nuevos.

2.2.2 Punto de Vista Externo

- Disminución de correcciones, cambios y quejas por parte de los clientes
- Mejora de la confianza de los clientes hacia la empresa.

- Mejora de la imagen y credibilidad de la empresa frente a los clientes y los mercados internacionales, facilitando la penetración o ampliación de nuevos mercados en el exterior, generando mayores ingresos.

2.3 Calidad del Software

La Calidad del Software es un conjunto de cualidades medibles y específicas de un producto que varía de un sistema a otro, en el cual el usuario puede percibir que el software desarrollado satisface sus requerimientos y expectativas. El interés por que sus productos sean cada vez más fiables y que cumplan sus necesidades han ido creciendo a medida que los clientes se han vuelto más selectivos con el software que adquieren.

Se debe diseñar la calidad del software junto con el sistema, no al final cuando el producto ya está terminado, si no se cumple esto podría provocar costos por fallas al producir software o por costos al reparar defectos en software ya desarrollado. El costo de calidad debe ser considerado en todo el ciclo de vida del proyecto, es decir a mayor calidad van a existir mayores costos, pero también van a ser mayores los beneficios adquiridos.

Para evaluar la calidad de software en un producto se debe utilizar las Revisiones Técnicas Formales (RTF), las cuales permiten mejorar la Calidad del Software, ayudando así a descubrir errores en la función, lógica o implementación de cualquier producto de software. Es un proceso riguroso que ayuda a detectar los posibles defectos en los productos y verifica que estos se ajusten a los estándares establecidos, entre los cuales se encuentra el tiempo de respuesta, seguridad de datos, cobertura entre otros.

Los proyectos de desarrollo de software, tienen como objetivo producir software de la mejor calidad posible, pero no necesariamente debe alcanzar una calidad perfecta, sino únicamente la necesaria y suficiente para que se ajuste al contexto de uso y de ser posible supere las expectativas del usuario. Al producir software se puede restar cualquier funcionalidad pero no se puede restar la calidad, se debe hacer un análisis del costo/beneficio en cuanto a los límites de errores o

casos especiales que se pueden dejar pasar, ya que estos no afectan o tienen un alto impacto en el funcionamiento general del software.

2.3.1 Modelos y/o Estándares de Calidad

Los modelos de calidad son sistemas basados en estudios que ayudan a las empresas a utilizar las mejores prácticas para el aseguramiento de la calidad. Estos se definen de forma jerárquica, este es un concepto que se deriva de un conjunto de sub-conceptos, los cuales se evalúan a través de un grupo de indicadores o métricas.

El uso de modelos y/o estándares permite que la calidad de software se la pueda definir, medir y planificar, también son una guía para lograr la productividad y la calidad. Su estructura es por lo general de tres niveles.

Factores de Calidad: También llamados “Atributos de Calidad Externos”, representan la calidad desde el punto de vista del usuario/cliente y son las características que componen la calidad. Entre los cuales se encuentran facilidad de uso, solución de problemas, información, registros y reportes.

Criterios de Calidad: También llamados “Atributos de Calidad Internos”, se descompone en un conjunto de criterios de calidad cada

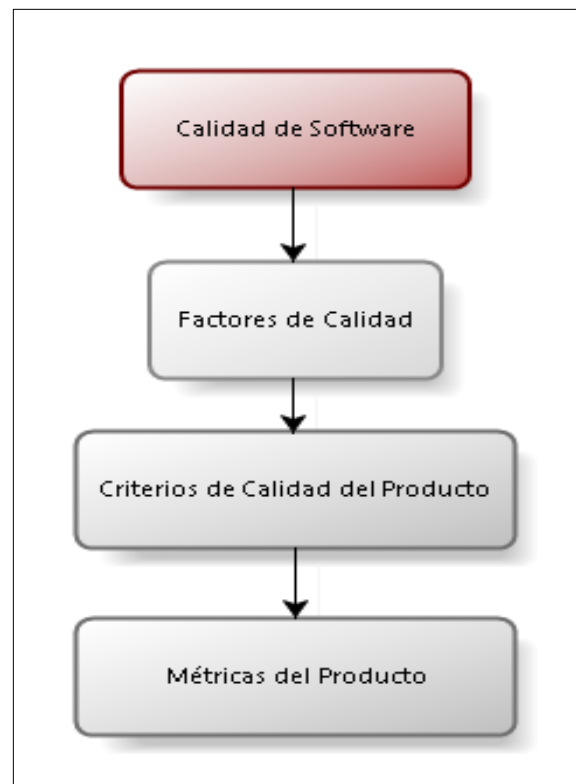


Gráfico 5. Estructura de un Modelo de Calidad del Software

Tomado de: Scalone, F. (2006). ESTUDIO COMPARATIVO DE LOS MODELOS Y ESTANDARES DE CALIDAD DEL SOFTWARE. Recuperado de <http://laboratorios.fi.uba.ar/lsi/scalone-tesis-maestria-ingenieria-en-calidad.pdf>

uno de los factores. Son atributos que, cuando están presentes, contribuyen al aspecto de la calidad que el factor asociado representa. Es una visión de calidad desde el punto de vista del producto de software. En estos se encuentra seguridad de registros, procesamiento de datos y cobertura de procesos internos.

Métricas: Se define un conjunto de métricas para cada uno de los criterios de calidad. Son medidas cuantitativas de ciertas características del producto, que dan una indicación del grado en que el producto posee un determinado atributo de calidad. En este caso se usan medidas relacionadas a la eficiencia y eficacia como la medición de tiempos y costos.

2.3.2 Calidad del Producto y del Proceso

El concepto de calidad de software abarca dos áreas importantes; la calidad del producto de software y la calidad del proceso de desarrollo. Ambos son independientes pero relacionados en la medida que la calidad del producto demanda de un proceso eficiente que permita su desarrollo.

Los objetivos que se establezcan para la calidad del producto dependen exclusivamente del proceso, debiendo en cada actividad aportar valor que se oriente a satisfacer necesidades estudiadas. Para ejemplificar lo antes dicho podemos basarnos en el caso de desarrollo de un software financiero, donde no solo basta con levantar correctamente los requerimientos, sino que estos deben ser resueltos y plasmados a través del diseño (arquitecto), implantación (desarrollador), validación y verificación (control de calidad).

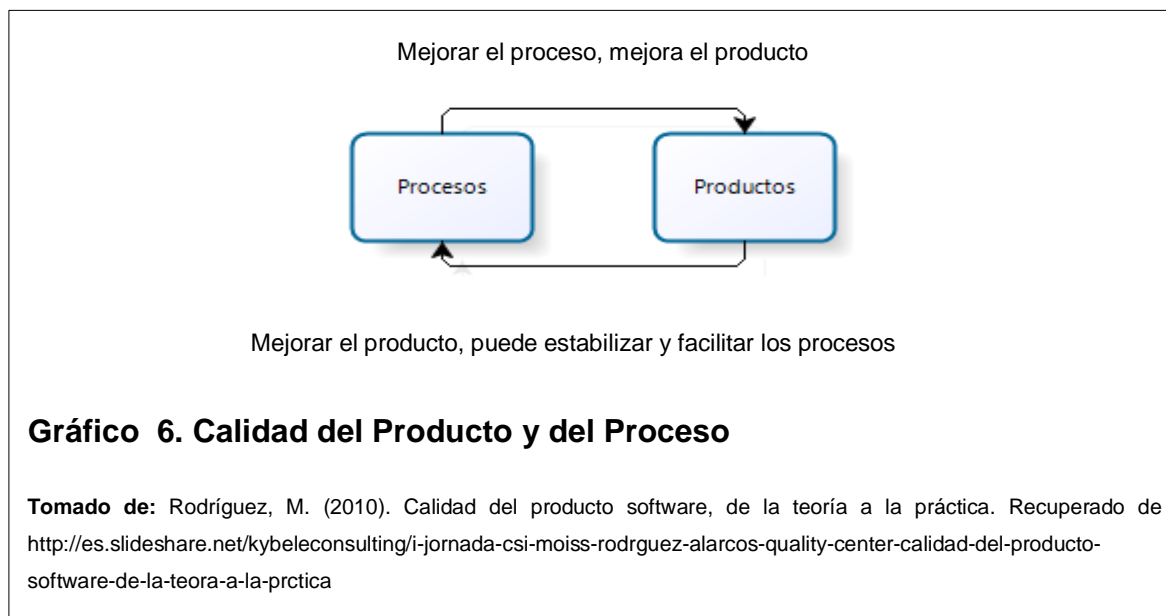
La calidad del producto, va a depender de cómo se lleve a cabo todo el proceso y cada subproceso del proyecto. Si se realiza un proceso de alta calidad producirá productos de alta calidad y todo esto a un menor costo, simplemente porque el número de retrasos será mínimo y no habrá que realizar doble trabajo. Podría darse el caso en que se pueda obtener alta calidad del producto de un proceso de baja calidad, pero esto solamente si se lo inspecciona lo suficiente y se

rechaza lo suficiente de lo que se ha producido, lo cual involucra tiempo y costos para la empresa.

La calidad del producto de software abarca los siguientes aspectos:

- **Calidad Interna (Verificación):** se lo mide a partir de las características intrínsecas, como el código fuente. Tipo de software utilizado, normativas para uso de las aplicaciones, criterios de programación.
- **Calidad Externa (Validación):** se lo mide en el comportamiento del producto, como en una prueba. Abarca la cobertura del sistema, funcionalidad y aporte a solucionar problemas existentes.
- **Calidad en Uso (Uso y Retroalimentación):** se lo realiza durante la utilización efectiva por parte del usuario.

En este contexto, se puede representar lo mencionado anteriormente en las dos dimensiones esenciales de la calidad.



Capítulo III

3. Análisis Crítico de los Principales Marcos de Referencia y Estándares de Calidad existentes para el Desarrollo de Software

3.1 Antecedentes

El desarrollo de software demanda del cumplimiento de normas y procedimientos que permitan mejorar la calidad de las aplicaciones, las cuales están orientadas a solucionar necesidades de los usuarios y mejorar la seguridad, transparencia y fidelidad en la información generada.

En este sentido, su desarrollo debe sustentarse en diferentes herramientas disponibles para que el producto final pueda disponer de todos los recursos que permitan su utilización, siendo en este caso, la estandarización y construcción de software mecanismos necesarios, indispensables y correlacionales.

Según Cortés, Roberto, (2009), el desarrollo de software implica la necesidad de alcanzar soluciones que permitan a los procesos ser más eficientes, eficaces y efectivos. Su diseño por lo tanto debe estar amparado en procedimientos que permitan su control, actualización y uso, siendo los marcos de referencia indispensables para ello. Al respecto, señala;

“El diseño viene a ser la estrategia que define la implementación técnica de los requisitos especificados en el análisis. En la visión genérica de la Ingeniería de Software, vendría a representar el cómo se debe llegar a la solución de los problemas existentes” (Cortés, 2009, p. 4)

Analizando lo expuesto, es evidente que el desarrollo de software requiere de procesos que orienten tanto su diseño como cobertura de las necesidades del usuario, para que los productos resultantes puedan aportar a mejorar la gestión de donde se utilicen. En este sentido, son justamente los marcos de referencia aquellos encargados de establecer los lineamientos y procedimientos para alcanzar niveles de calidad que permitan a los productos resultantes ser utilizados de manera adecuada y aportar el mejoramiento en el rendimiento de los procesos que los utilicen.

Al respecto, los marcos de referencia basados en la normativa ISO y Modelos de Madurez CMMI, objetos de estudio en la presente investigación, establecen una serie de parámetros complementarios, los cuales ameritan ser identificados y analizados para que puedan ser utilizados de manera efectiva. Su uso permite disponer de un modelamiento ordenado y sistemático para la construcción de software respetando normativas que eviten errores que afecten la calidad de los productos y generen falencias en los procesos que apoyan.

De esta manera, el presente capítulo, presenta un estudio completo de los diferentes estándares y procedimientos existentes que permitan orientar el desarrollo del software a fin de que se puedan aprovechar de manera efectiva, elevando la calidad de los productos desarrollados en el país.

Concretamente, su desarrollo relaciona diferentes marcos de referencia existentes, verificando la cobertura, características, semejanzas y diferencias, aspectos que como se verán más adelante, permitirán utilizarlos de manera complementaria en la medida que cada uno se focaliza en áreas necesarias a considerar en el desarrollo del software.

3.2 Los Marcos de Referencia y su Aplicabilidad Universal

El avance de la ciencia y tecnología ha generado la mayor accesibilidad a procesos, metodologías y normas para el diseño de software. Cada uno estructurado en base a conceptos que permitan la optimización de los recursos, el respeto al marco jurídico establecido y el fomento de los principios básicos de la gestión, para que apoyen a mejorar la planificación, organización, dirección y control de quienes lo utilizan.

Sin embargo, la disponibilidad de estándares e inclusive procesos no es suficiente para alcanzar niveles de calidad efectivos en el desarrollo del software, más aun cuando su existencia puede inclusive generar confusiones en el diseñador, no sabiendo cuales utilizar o cómo combinarlos para alcanzar el mejor resultado. Otra debilidad existente es la falta de relación de los estándares con la cultura y comportamiento de las áreas donde se aplican, debiendo adaptarse a procesos universales que en la práctica no siempre son viables y adecuados.

Lo expuesto, revela un problema claro en el desarrollo de software, dado principalmente por la amplia disponibilidad de modelos y normativas las cuales no siempre se relacionan y responden a las necesidades propias de los mercados en donde se desea desarrollar los productos. Este es principalmente una de las principales barreras existentes en países como Ecuador, que tienen que ajustarse a normas y procesos internacionales, mismos que no han sido diseñados para brindar respuestas específicas sino universales, las que inclusive pueden ser objeto de errores y falencias que atenten contra la calidad del producto y servicio.

Además, la amplia oferta de marcos de referencia, pueden dar lugar a conflictos en su aplicación, no pudiendo ser totalmente implementados, lo que tiende a generar vacíos en el desarrollo, dando lugar a falencias que afecten la calidad de los productos tecnológicos generados.

Surge de esta manera preguntas con respecto a la validez de las normativas internacionales y su aplicación en procesos nacionales, encontrando que si bien estas responden a estudios e investigaciones que buscan ser universales, no garantizan una aplicación específica, lo que en el diseño de software tiende a generar debilidades que pueden afectar a los diferentes usuarios de los productos desarrollados.

Bajo este entorno, es importante analizar los marcos de referencia a fin de identificar sus procesos y estructuras para que estas puedan ser aprovechadas de manera eficiente, pudiendo inclusive establecer reformas o ajustes que permitan fomentar el alcance de altos niveles de calidad y procesos de mejora continua que eleven el rendimiento de quienes lo utilicen y les permitan alcanzar mejores niveles de desempeño y rendimiento.

3.3 Fundamentos principales de los Marcos de Referencia en el Ciclo de Vida del Software

La ingeniería de software cuenta dentro de sus procesos de desarrollo varios elementos de soporte que permiten asegurar la calidad de sus productos y su efectiva utilización, brindando soluciones que dinamicen y mejoren su rendimiento. Son por lo tanto los marcos de referencia soportes necesarios que

permiten estandarizar los procesos internos de desarrollo y orientar su ejecución, evitando falencias que no permitan alcanzar los objetivos propuestos.

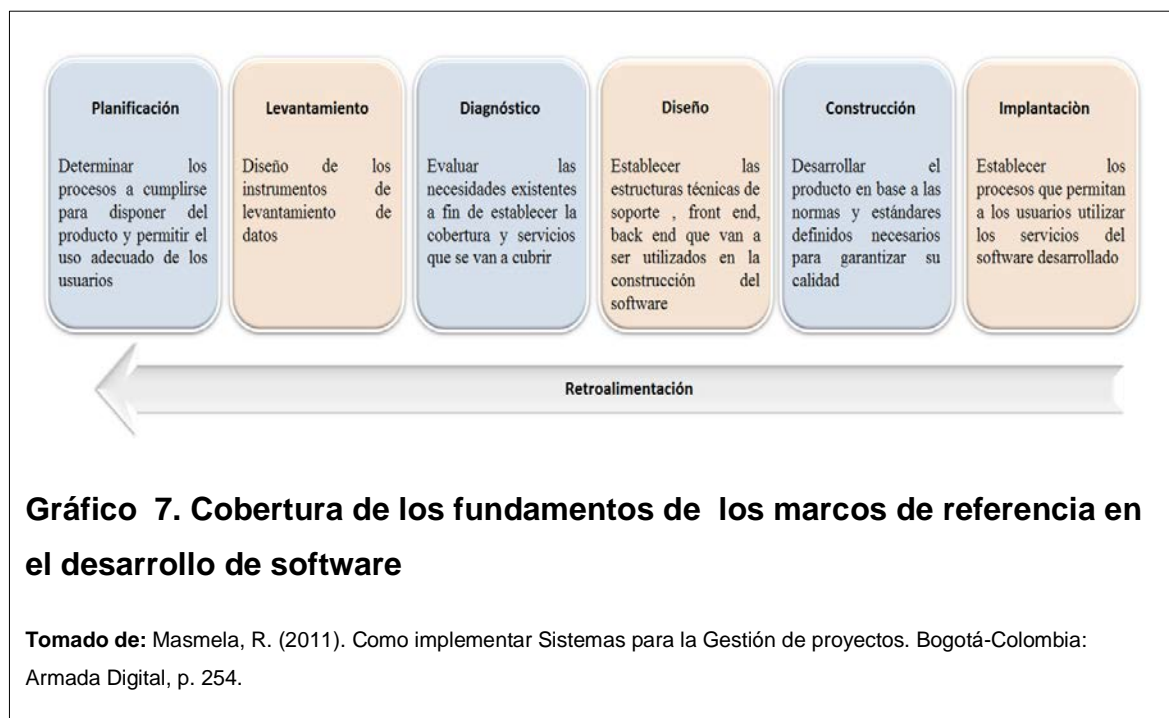
Su cobertura como se observa es amplia, cubriendo aspectos relacionados al desarrollo, mantenimiento y construcción de productos que respondan a necesidades existentes en los usuarios para satisfacerlas plenamente. En este sentido, los fundamentos básicos de los marcos de referencia pueden determinarse en función de los siguientes aspectos citados a continuación:

Tabla 4. Fundamentos de los marcos de referencia en el desarrollo de software

Fundamento	Descripción
Planificación	Disponer de mecanismos que permitan el diseño del software, contemplando los diferentes procesos inmersos en su construcción con una clara alineación a los objetivos que se desean alcanzar.
Análisis	Contar con procesos que permitan mejorar el aseguramiento de la calidad, verificando el cumplimiento de las normativas existentes para evitar discrepancias en el producto final que afecten el rendimiento de los productos.
Estandarización	Cumplimiento de normativas específicas para evitar resultados no esperados, fomentando la organización interna en los procesos necesarios para el diseño y construcción del software
Procedimientos	Establecer las fases de desarrollo necesarias para: <ul style="list-style-type: none"> • Identificar la necesidad • Establecer la cobertura de las aplicaciones a desarrollar • Definir los aspectos técnicos para la construcción • Determinar las normativas internas para la construcción del software • Establecer los procesos de implementación • Identificar los mecanismos de validación y evaluación
Organización	Establecer la estructura de soporte que permita el adecuado funcionamiento del software, señalando los perfiles de quienes lo administran, utilizan y ajustan
Gestión de cambios	Permitir la actualización y mejoras de las aplicaciones desarrolladas para que estas puedan responder a los cambios en el mercado existentes

Gobernar	Definir los procesos necesarios para administrar las aplicaciones y servicios del desarrollo del software así como para dirigir a los equipos responsables de su diseño y construcción.
Implantación	Comprende los mecanismos que permiten utilizar los diferentes servicios de los productos desarrollados.

Analizando los fundamentos citados es importante identificar su amplia cobertura, en la medida que su gestión debe abarcar todos los procesos necesarios para desarrollar software, pudiendo estos clasificarse en función al siguiente organizador gráfico:



Cada uno de los elementos citados es importante y su ejecución determinante en la obtención de calidad y capacidad de solventar problemas en los usuarios. Su desarrollo debe ejecutarse optimizando los recursos disponibles a fin de evitar desperdicios de los recursos que afecten su diseño y ejecución.

Su amplia cobertura además, permite establecer la capacidad de relación que existe en los diferentes marcos de referencia, en la medida que estos pueden ser especializados en diferentes elementos, pudiendo ser complementados de manera adecuada, no representando esta situación una inconformidad en el

diseño de software, sino principalmente un aprovechamiento de sus procesos de manera adecuada, maximizando su utilidad en cada uno de los fundamentos existentes que permitan alcanzar máximo rendimiento y calidad.

La capacidad de relación e integración de los diferentes marcos de referencia comprenden parte de la responsabilidad en la planificación requerida en el desarrollo, aspecto que solo será viable en función del amplio conocimiento de las características que estos tienen. En tal virtud, se procede a realizar un análisis comparativo del Modelo de Madurez de Capacidades CMMI y las Normas ISO (ISO 9001:2008, ISO 9000-3:2004)

3.4 Características de las Normas ISO y CMMI

3.4.1 Normas ISO para el desarrollo de software

Las normas ISO enfocadas en el desarrollo de software, comprenden un conjunto de normativas que buscan la estandarización de los procesos a fin de que estos puedan desarrollarse en cumplimiento de disposiciones que eviten errores y riesgos que afecten el servicio prestado.

Su estructura abarca diversos sistemas que permiten gestionar la calidad en el diseño, construcción, inspección, instalación y utilización del software permitiendo disponer de información especializada sobre cada uno de estos que permita tomar decisiones que impulsen el crecimiento organizacional.

En el ámbito de software, la estandarización de los procesos no solo debe comprender las características del software sino su estructura técnica necesaria para que este pueda operar adecuadamente, pudiendo actualizarse acorde a las necesidades de los procesos en los cuales se implementan. Señala Salavert, Isidro (2008) al respecto lo siguiente:

“A pesar del avance que suponen los marcos de referencia, estos solo constituyen el primer paso para asegurar la calidad de los modelos conceptuales, ya que es necesario establecer métricas para valorar las diferentes factores de calidad de manera objetiva y consistente. En Ingeniería de Software se han propuesto centenares de métricas para

productos, procesos y recursos software. Desgraciadamente, la mayor parte de ellas se han centrado solo en las características de los programas, sin tener en cuenta los modelos de datos” (Salavert, 2008, p. 111)

Lo expuesto permite entender que las normativas de estandarización, necesariamente deben cubrir elementos como los siguientes:

- Normativas para la evaluación y calificación de aplicaciones para el front end y back end
- Normativas para la relación identidad de los datos
- Normativas para la nomenclatura de los objetos y procesos de programación
- Normativas para la seguridad de los datos
- Normativas para el almacenamiento y procesamiento de datos
- Normativas en la presentación de las pantallas del software
- Normativas para los procesos de evaluación
- Normativas para los procesos de implantación
- Normativas para la evaluación de los procesos internos

Definidos estos procesos, las normas ISO evitan costos y gastos innecesarios y permiten disponer de mayor aceptación por parte de los clientes y usuarios de los productos de software diseñados.

En este caso, las normativas que conforman la familia ISO aplicada al diseño de software se describen en el siguiente cuadro:

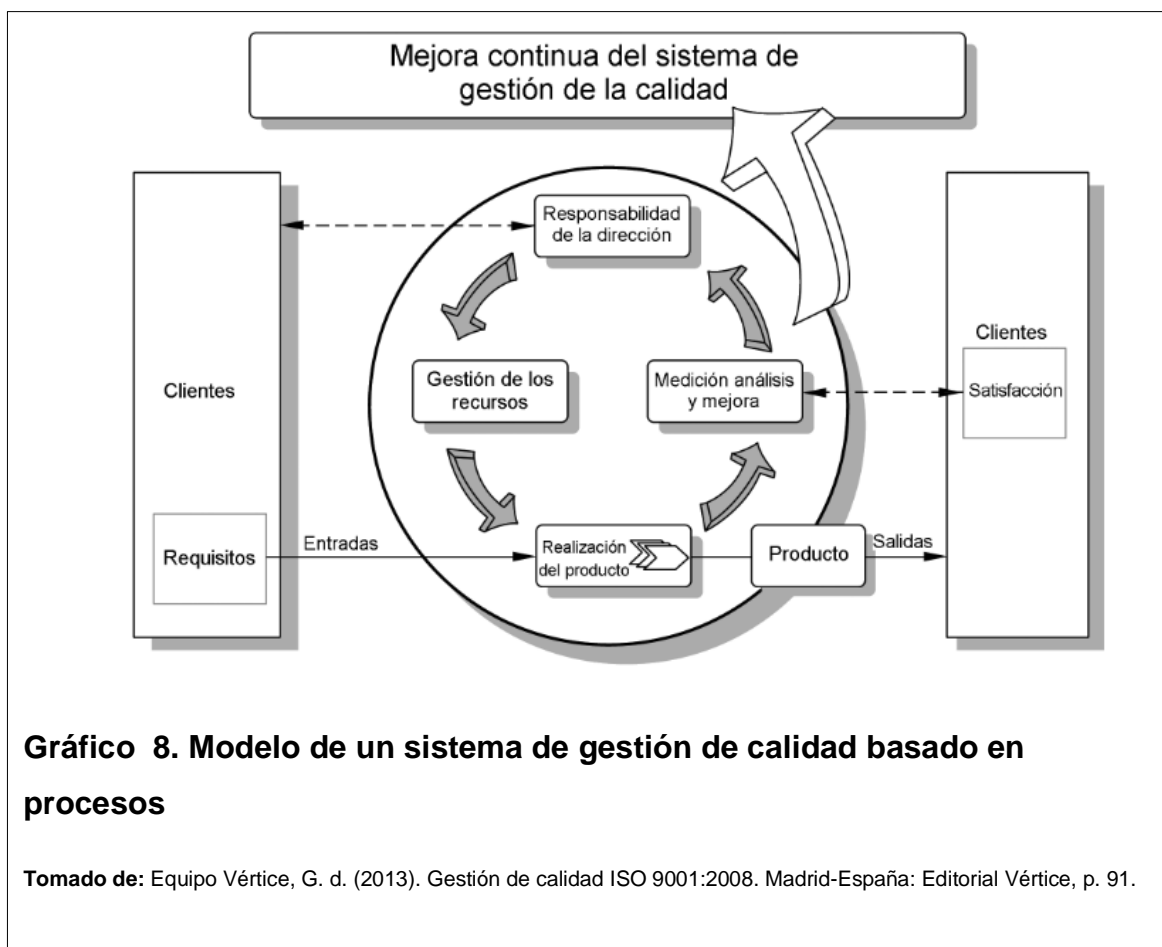
Tabla 5. Normas Internacionales ISO

Norma	Descripción
ISO 9001:2008	<p>Establece la necesidad de la adopción de procesos estructurados que permitan mejorar la eficiencia y eficacia de la gestión interna permitiendo atender adecuadamente los requerimientos de los clientes y usuarios.</p> <p>Su desarrollo se basa en el “Enfoque basado en procesos”; que establece una definición de los procesos y dentro de estos las actividades que los comprenden y su relación para alcanzar un resultado esperado.</p> <p>De esta manera, su desarrollo comprende los siguientes elementos:</p> <ul style="list-style-type: none"> • Conocimiento y cumplimiento de normas que permitan a los procesos cumplir con los parámetros definidos para alcanzar la calidad en el servicio y atención de los clientes. • Mejorar los procesos definidos para aportar valor y poder contribuir efectivamente a atender las necesidades de los clientes. • Establecer mecanismos de evaluación que permitan determinar la eficiencia, eficacia y efectividad de las aplicaciones y su funcionalidad. • Mejora continua de los procesos para que estos contribuyan al crecimiento y desarrollo organizacional.
ISO 9000-3:2004	<p>Su enfoque se basa en el mejoramiento de las operaciones y falencias en los procesos de diseño y construcción de software, brindando claras orientaciones para el desarrollo, suministro y mantenimiento del software. La normativa comprende el desarrollo de sistemas de información, los procesos de ciclo de vida y calidad del software.</p> <p>De esta manera, la estructura de ejecución es amplia debiendo atender los siguientes aspectos:</p> <ul style="list-style-type: none"> • Implementación de sistemas de calidad • Responsabilidad de la gestión de calidad en el desarrollo del software • Mantenimiento del software • Control de documentos • Procesos de medición, validación y evaluación • Determinación de procesos de aprovisionamiento • Herramientas y técnicas utilizadas para el diseño y construcción del software • Auditorías internas de calidad
ISO 15504	<p>Se enfoca en el desarrollo del software, concentrándose en tres áreas específicas:</p> <ul style="list-style-type: none"> • Estándares para la evaluación de los procesos de software. • Mecanismos de evaluación de la calidad de software. • Promover la transferencia de tecnología. <p>Su aplicación permite someter a los proveedores de software a esquemas estandarizados de evaluación. Además, permitirá establecer sistemas de mejora continua que permitan perfeccionar constantemente los productos desarrollados mejorando su enfoque hacia las necesidades del mercado demandante.</p>

Tomado de: Norma Internacional - ISO 9001. (2013). Recuperado de <http://farmacia.unmsm.edu.pe/noticias/2012/documentos/ISO-9001.pdf>

Las normativas expresadas, se fundamentan en estandarizar los procesos, alinearlos a objetivos establecidos y abarcar los diferentes recursos para disponer de productos de calidad en el ámbito de software. Su modelo de ejecución es cíclico y permanente, tomando en consideración que siempre cada producto es sujeto a un perfeccionamiento y mejora.

Su desarrollo responde a un modelo integral, relacionado orientado en procesos definidos, pudiendo estos ser representados en el siguiente organizador gráfico:



En base de lo expuesto, se observa la secuencia cíclica existente que permite desarrollar el producto enfocado a la satisfacción de los clientes, fomentando su mejora continua, aspecto que abarca diversos niveles de ejecución y relación.

3.4.2 CMMI

El diseño y desarrollo de software requiere de modelos que permitan orientar su ejecución, cumpliendo con todos los procesos necesarios que permitan disponer de un producto de alta calidad. En este caso, CMMI se orienta principalmente a disponer de una visión de procesos que permita su mejora continua y evaluación a fin de que se detecten oportunamente posibles falencias que atenten contra la calidad de los servicios prestados.

Principalmente su ejecución abarca tres áreas de gestión las cuales son independientes pero interrelacionadas e indispensables en el diseño y construcción de software estas son las siguientes:

- **Desarrollo de software:** Comprende diferentes actividades necesarias para desarrollar el producto. En este caso abarca los procesos de levantamiento, diagnóstico, diseño y construcción. Los principales entregables de su cumplimiento son:
 - Información actualizada sobre los requerimientos de los usuarios
 - Información técnica de las aplicaciones a utilizar en el Front y Back End
 - Prototipo desarrollado
 - Informe de las pruebas ejecutadas al prototipo
 - Producto terminado
 - Proceso de implementación disponible para su utilización.
- **Adquisición**
- **Servicios**

Las áreas descritas, permiten determinar la cobertura del CMMI, pudiendo esta describirse en los siguientes aspectos:

- Una forma de integrar los elementos funcionales de una organización
- Un conjunto de mejores prácticas basadas en casos de éxito probado de organizaciones experimentadas en la mejora de procesos
- Ayuda a identificar objetivos y prioridades para mejorar los procesos de la organización, dependiendo de las fortalezas y debilidades existentes

- Un apoyo para que las empresas complejas en actividades productivas puedan coordinar sus actividades en la mejora de procesos
- Un punto de referencia para evaluar los procesos actuales de la organización. (Nava, 2013)

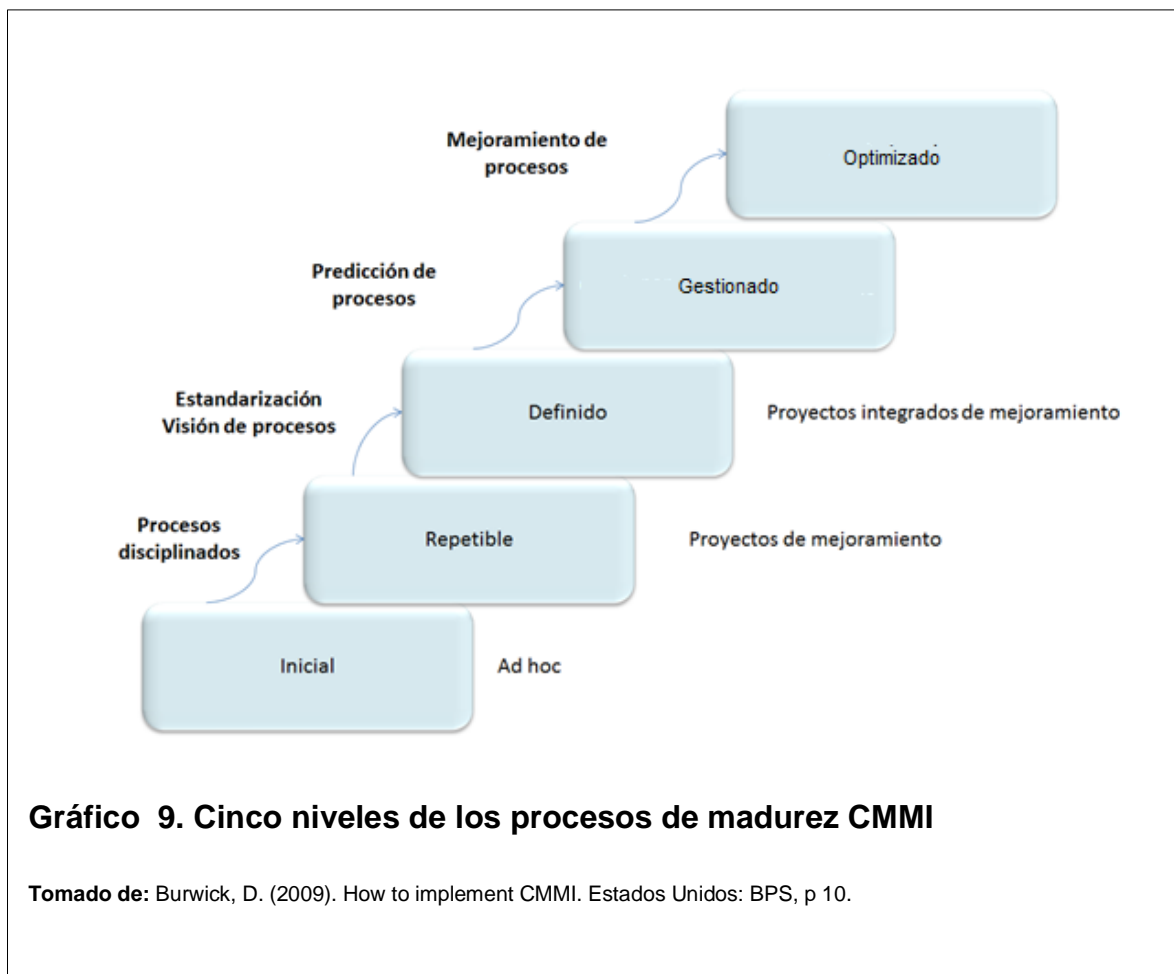
Según Burwik, Diane (2009), el CMMI es un modelo de creación y desarrollo, orientado a la mejora de procesos y evaluación comparativa que permite implementar medidas enfocadas a mejorar el rendimiento constante de los procesos en donde se aplican. Señala sobre lo expuesto, lo siguiente:

“La ingeniería de software se compone de cinco niveles que permiten alcanzar una madurez sostenida necesaria para que los productos elaborados puedan responder efectivamente a las demandas existentes en el mercado, contribuyendo a mejorar la gestión de las empresas, profesionales y usuarios” (Burwick, 2009, p. 10)

Como se observa, el CMMI determina un proceso de maduración necesario para alcanzar productos de calidad, elaborados e implementados en función del seguimiento de modelos que eviten errores que afecten al usuario.

En este sentido, su estructura de desarrollo comprende un conjunto de niveles interrelacionados y enfocados al mejoramiento del desarrollo del software, en donde cada empresa puede evaluar su comportamiento, disponiendo de parámetros de control que le ayuden a definir acciones y estrategias para mejorar su madurez en el desarrollo. De esta manera, el CMMI orienta a las empresas a mejorar sus procesos, en la medida que pueden identificar sus falencias y establecer mecanismos que mejoren la calidad de sus productos, proceso que va relacionado directamente con su crecimiento en el mercado.

Los cinco niveles de madurez, se describen en el siguiente organizador;



El proceso de madurez se conforma de la definición de procesos, alineados a los objetivos empresariales, apoyados en normativas que permiten mantener un control actual con proyección visionaria, capaz de tomar continuamente decisiones que impulsen un mejoramiento constante. Técnicamente se compone de 22 áreas que permiten a la organización avanzar en los diferentes niveles. (Campderrich, 2008, pp. 31-33)

Conforme se ha expuesto en el organizador gráfico, los niveles de madurez pueden ser interpretados de la siguiente manera:

- **Nivel I - Inicial:** La calidad interna del software, depende exclusivamente de los recursos que intervienen, no de los procesos los cuales no se encuentran definidos en su totalidad.

- **Nivel II - Repetible:** El proyecto es gestionado y controlado durante el desarrollo, se establecen prácticas institucionalizadas, el control interno y la fijación de requerimientos.
- **Nivel III - Definido:** El desarrollo y mantenimiento de software se encuentran documentados. Las personas responsables cuentan con información que orienta su diseño y desarrollo. Se realiza el enfoque en el proceso de manejo de proyectos (IPM).
- **Nivel IV - Gestionado:** Se usan objetivos medibles para alcanzar las necesidades de los clientes y de la organización. Se usan métricas cuantitativas para la gestión de proyectos.
- **Nivel V - Optimizado:** Implica la mejora continua como un proceso para mantener un perfeccionamiento constante de los mecanismos y resultados esperados.

El desarrollo de los niveles de madurez, tienden a gestionar proyectos integrados de mejoramiento, los cuales comprenden modelos enfocados a innovar, crear y fomentar un crecimiento sostenido apoyado en productos concebidos para que se optimicen y aprovechen los recursos disponibles, mejorando los sistemas de información, comunicación, control y atención a las diferentes áreas internas y externas de una organización.

Los niveles expuestos señalan que la madurez se alcanza en la medida que la empresa pueda disponer de procesos definidos y estructurados que aporten valor al cliente interno y externo. En este sentido, su desarrollo comprende los siguientes aspectos:

- Procesos definidos en todas las áreas de gestión
- Procesos identificados y levantados asignados a responsables que responden a perfiles determinados y necesarios
- Procesos de levantamiento, medición y evaluación determinados
- Mecanismos de mejora implementados y controlados

En la medida que el desarrollo del software vaya atravesando los diferentes niveles, se perfeccionará los productos, entendiendo que estos contribuirán al mercado demandante a mejorar su desempeño.

3.5 Comparativa de Normas ISO (9001:2008, 9000-3:2004) y CMMI en el control de calidad del software (Ventajas, desventajas y características)

3.5.1 Ámbito comparativo

Una vez analizados los procesos inmersos en la normativa ISO y CMMI, se observa que ambos se enfocan en la visión de procesos, la cual representa la base elemental para el desarrollo de software y su adecuada implantación y uso.

Su aplicación se enfoca en mejorar la gestión en el desarrollo, permitiendo disponer de una mejor calidad en los productos desarrollados, la cual como se evidenció dispone de diversos niveles que permiten alcanzar la madurez.

Bajo este aspecto, el ámbito comparativo establece diferentes elementos que se han agrupado de la siguiente manera:

- **Cobertura de desarrollo:** Abarca las áreas de gestión que cubren los marcos de referencia evaluados para identificar sus semejanzas y diferencias.
- **Procesos incorporados:** Determina los enfoques que estos tienen y los objetivos de su implementación.
- **Alcances:** Determina los elementos esperados en su implementación y como estos contribuyen a la empresa que los aplica.

En cada uno de los ámbitos se determinarán características que permitan comparar los marcos de referencia y obtener información útil para identificar fortalezas y debilidades existentes.

3.5.2 Estudio de las características

El estudio comparativo en base de los ámbitos establecidos, se han agrupado acorde a las siguientes características:

Tabla 6. Características de los Marcos de Referencia Estudiados

Ambito	Característica	Normativa ISO	CMMI
Cobertura de desarrollo	Estructura basada en	Procesos	Procesos
	Enfoque basado en	Mejoramiento de los procesos para alcanzar altos niveles de calidad	Mejoramiento de los procesos en base a niveles de madurez que permiten disponer de productos de mayor calidad
	Aplicación en	Procesos técnicos, administrativos y de gestión en el diseño y construcción del software	Desarrollo, adquisición y servicios del software
	Objetivos	Establecer normativas cuyo cumplimiento evite errores internos que afecten la satisfacción el cliente, determinando procedimientos internos que deben ser cumplidos	Disponer de un sistema de gestión que permita alcanzar un mejor nivel de madurez en los productos de software diseñados para alcanzar el mejoramiento continuo
	Mecanismo	Estructura de normas interrelacionadas	Niveles de gestión de cumplimiento
Procesos incorporados	Procesos inmersos	Normas clasificadas en familias que deben ser implementadas de manera estandarizada	Enfoque en el desarrollo de proyectos de ingeniería de software con proyección de escenarios
	Estructura de ejecución	Aplicación de normas para la estandarización de los procesos	Modelo de gestión que orienta la ejecución, diseño, construcción e implementación de software.
	Evaluación	Verificación del cumplimiento de los requerimientos exigidos	Cumplimiento de los pasos progresivos en base a los niveles que conforma el modelo
	Desarrollo	La norma es flexible por lo que en base a sus requerimientos deben formularse las guías de implementación, sin que estas	Mantiene una secuencia lógica y ordenada de implementación que
Procesos			

incorporados		sean dispuestas de manera obligatoria.	permita a las empresas ir alcanzando mayores niveles de maduración.
	Cobertura	La norma es amplia y comprende varios procesos que demandan de una selección y atención para evitar confusiones.	Se focaliza en el diseño y desarrollo de software, incorporando procesos de adquisición y servicios
	Carácter	Estandarizado, estático, normalizado, universal	Flexible, adaptado a la empresa, enfocado en sus requerimientos puntuales
Alcances	Ejecución	Se enfoca en alcanzar una certificación que describa la existencia de calidad en sus procesos	Su desarrollo se focaliza en perfeccionar los procesos de diseño y construcción de software no en certificaciones
	Control	Evalúa el estado de la empresa en base a estándares fijados los cuales no pueden ser omitidos o cambiados	Su desarrollo es flexible y se ajusta al comportamiento y cultura tanto de la empresa como el mercado.
	Sistema de control	Estandarizado bajo la intervención de entes externos a la empresa que determinan la renovación de la certificación	Realizado por la propia empresa en base a lineamientos que consideren adecuados y requeridos para la empresa.

Analizando las características desarrolladas en los ámbitos evaluados, es posible entender que los marcos de referencia evaluados son complementarios, en la medida que pueden ser utilizados de manera paralela. En este caso, cada uno dispone de ciertas ventajas y desventajas que pueden apoyarse alcanzando mejores resultados en el diseño y desarrollo de software.

3.5.3 Ventajas y desventajas

Cada marco de referencia posee características que pueden derivar en ventajas para la empresa las cuales se describen a continuación:

- **Ventajas de la Norma ISO**

Su cobertura es amplia y abarca varios procesos de evaluación, no necesariamente especializados en software lo que puede apoyar a mejorar los procesos sobre los cuales estos se desarrollan e implementan. En este caso, intervienen los recursos, metodologías, mecanismos de levantamiento y evaluación, permitiendo tener una visión amplia de las necesidades del cliente y los procesos que permiten satisfacerlos.

Al ser parte de una amplia familia, es posible abarcar varios procesos de estandarización, disponiendo de mecanismos que permitan apoyar en los diferentes procesos de construcción de software.

La obtención de certificación responde a procesos de evaluación externos, lo que genera seguridad y garantías de haber alcanzado una estandarización en los procesos. Esto ayuda a fomentar la imagen de los productos en el mercado, pudiendo ser inclusive requisitos de los clientes la disponibilidad de certificaciones que abalicen la calidad de sus productos.

- **Desventajas de las Normas ISO**

Su aplicación no es totalmente especializada en el campo de software lo que puede omitir procesos técnicos inmersos en su diseño y construcción, afectando el cumplimiento los requerimientos de los clientes.

Al ser procesos universales y estandarizados tienen poca flexibilidad en su aplicación, debiendo apegarse a lo dispuesto en la norma sin excepción. Esto puede generar falencias en cuanto a la atención de requerimientos puntuales de los clientes, tomando en cuenta diferencias referentes a cultura y comportamiento organizacional. En este caso por ejemplo pueden omitir aspectos propios de la cultura y costumbre de los usuarios que afecten la comprensión de las aplicaciones requeridas. Ejemplo: Sistemas de codificación en base a uso del idioma inglés.

La amplia familia de normas puede generar confusión en los equipos responsables de su implementación, en donde se distorsione el objetivo principal de la calidad del software.

Finalmente, al ser un proceso estandarizado su objetivo se fundamenta en la certificación, lo que puede generar problemas en cuanto a procesos que demande el cliente y que no puedan atenderse a través del cumplimiento de las normas.

- **Ventajas del CMMI**

Las ventajas del CMMI se basan en disponer de un modelo que orienta los procesos internos a seguir, pudiendo estos ser evaluados en base a parámetros de madurez. Su desarrollo tiene un amplio enfoque en el mejoramiento continuo de los productos de software permitiendo su orientación a satisfacer las necesidades de los clientes.

Su desarrollo es flexible y se adapta a la cultura organizacional de la empresa, facilitando la adopción de mecanismos que se consideren necesarios para mejorar la calidad del producto.

La principal ventaja es la especialización del modelo en el diseño de software permitiendo disponer de guías adecuadas para que las empresas del sector alcancen un direccionamiento adecuado.

Otro aspecto relevante es la facilidad de autoevaluación que promueve su implementación, lo que apoya a desarrollar una cultura focalizada en sus necesidades. Finalmente, su aplicación fomenta un crecimiento constante y permanente, relacionando sus procesos y recursos para que estos puedan ser aprovechados de mejor manera.

- **Desventajas del CMMI**

Al ser un proceso especializado puede omitir procesos complementarios los cuales de alguna manera incidan en la calidad del producto. Estos no siempre pueden detectarse oportunamente por lo que las medidas implementadas no siempre se enfocan a solventar las causas de los problemas existentes.

Su desarrollo demanda de un adecuado criterio por parte de los responsables, en la medida que al tener flexibilidad pueden caerse en aspectos que no tienen un verdadero aporte de valor, incrementándose los costos y afectando la rentabilidad y/o proyecto de la empresa.

El análisis realizado muestra aspectos semejantes en ambos marcos, concluyendo como se indicó que estos pueden ser complementarios, en la medida que ambos se focalizan en procesos y buscan la calidad como un fin.

En el siguiente cuadro, se han desarrollado aspectos semejantes y diferentes para que se tenga una visualización más adecuada de su cobertura:

Tabla 7. Cuadro de semejanzas y diferencias

Semejanzas
Ambos sistemas están enfocados en procesos
Buscan la calidad como un fin necesario a alcanzar
Disponen de estructuras enfocadas a fines determinados
Comprenden procesos de implantación que ameritan recursos internos y sistemas de adopción
Se focalizan en el cliente
Permiten el mejor rendimiento de la empresa

Diferencias
La normativa ISO es estandarizada poco flexible y enfocada en una amplia cobertura universal. El CMMI es abierta y puede ser desarrollada mediante mecanismos que la empresa considere adecuados para crecer en los niveles citados
La normativa ISO es amplia y abarca temas adicionales al diseño de software. El CMMI es especializado en software.
La normativa ISO permite alcanzar una certificación de calidad.
La normativa ISO es estática no acepta cambios. El CMMI es flexible.
La normativa ISO demanda de auditorías externas. El CMMI mantiene sistemas de autoevaluación que son ejecutados por su propio personal sin que exista intervención externa en las mismas. En este caso se enfoca al mejoramiento continuo interno.

Capítulo IV

4. Adaptación y Propuesta de un nuevo Marco de Referencia de Calidad para el Desarrollo de Software

El cambio de la matriz productiva propuesta por el Gobierno Nacional, tendiente a fortalecer la economía nacional, a incrementar el comercio exterior y reducir la dependencia en el sector petrolero demanda del desarrollo de segmentos, los cuales a nivel nacional e internacional se encuentran marcando profundas diferencias, impulsando un crecimiento sostenido y sustentable de los diferentes sectores económicos.

La tecnología es un medio capaz de acelerar la productividad, mejorar la gestión administrativa, operativa, financiera, entre otras. Su desarrollo promueve la eliminación de las barreras físicas entre países permitiendo una mejor comunicación y difusión de información.

Ecuador no es un país pionero en el desarrollo de tecnología siendo principalmente consumidor, lo que ha incrementado ampliamente sus costos, reduciendo sus niveles de competitividad. De esta manera, su desarrollo representa una oportunidad para fomentar el crecimiento a nivel interno con amplia proyección internacional.

Al ser la tecnología un ámbito amplio demanda de estructuras y marcos de referencia que le permitan disponer de efectivas orientaciones que mejoren los productos de calidad, es decir que estos se enfoquen en necesidades existentes, sean accesibles y brinden las facilidades y seguridades que son necesarias para que aporten a mejorar la gestión y fomentar un crecimiento tanto individual como colectivo de los diferentes usuarios.

En este sentido, el presente capítulo propone la creación de un marco de referencia especializado en el país, que aporte al fomento y desarrollo del software de calidad para que sea una guía que permita evitar errores y obtener productos adecuadamente elaborados fomentando un crecimiento integral de la sociedad.

4.1 Lineamientos del marco de referencia

El desarrollo de un marco de referencia para el diseño y elaboración de software debe cumplir con lineamientos que permitan obtener un mecanismo útil, confiable y adecuado. De esta manera, se proponen los siguientes lineamientos que se consideran necesarios para los usuarios interesados en el marco propuesto:

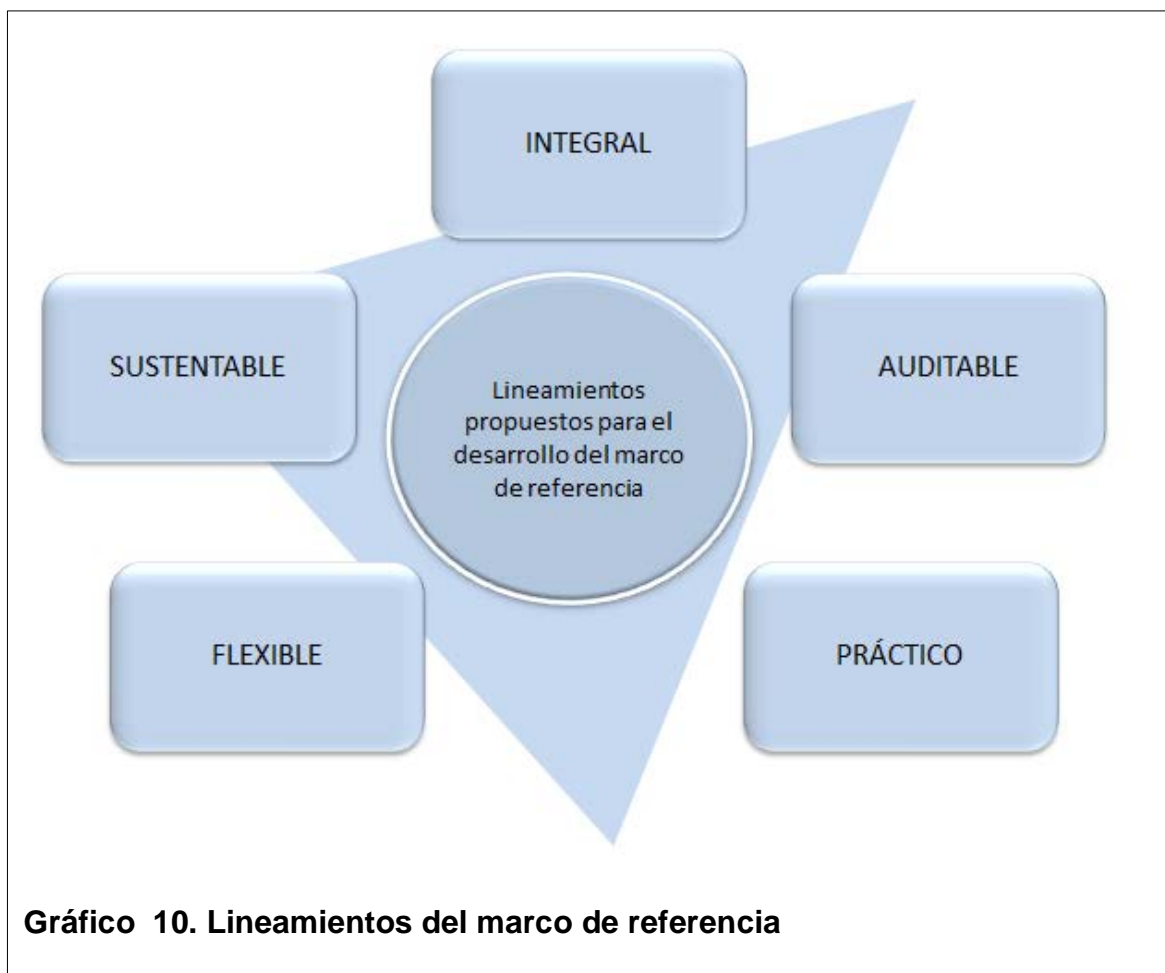


Gráfico 10. Lineamientos del marco de referencia

Los lineamientos que conformarán el desarrollo de la propuesta son:

- **Integral:** El marco de referencia debe abarcar todos los procesos inmersos en el diseño y desarrollo de software. Incluirá los procesos de identificación de la necesidad, diseño, desarrollo e implementación, permitiendo mejorar la calidad de los productos desarrollados.

- **Sustentable:** Su desarrollo contará con procedimientos, técnicas y metodologías que permitan respaldar las acciones a cumplir, brindando seguridad a los usuarios.
- **Flexible:** El marco de referencia no puede ser estático, debe tener la capacidad de adaptarse a los cambios que existen en la sociedad, siendo útil su aplicación para desarrollar productos de calidad orientados a las necesidades que el mercado demanda.
- **Práctico:** El marco de referencia debe ser viable para quien lo aplique, pudiendo utilizarlo conforme sus intereses. Es importante citar que el marco puede ser utilizado de manera parcial o total en base a las necesidades que tenga el usuario.
- **Auditable:** El marco de referencia debe ser sujeto a toda prueba y evaluación, mejorando su eficiencia y permitiendo que los usuarios tengan la seguridad de contar con un modelo útil que aporte el direccionamiento para alcanzar calidad en los productos y aplicaciones desarrolladas.

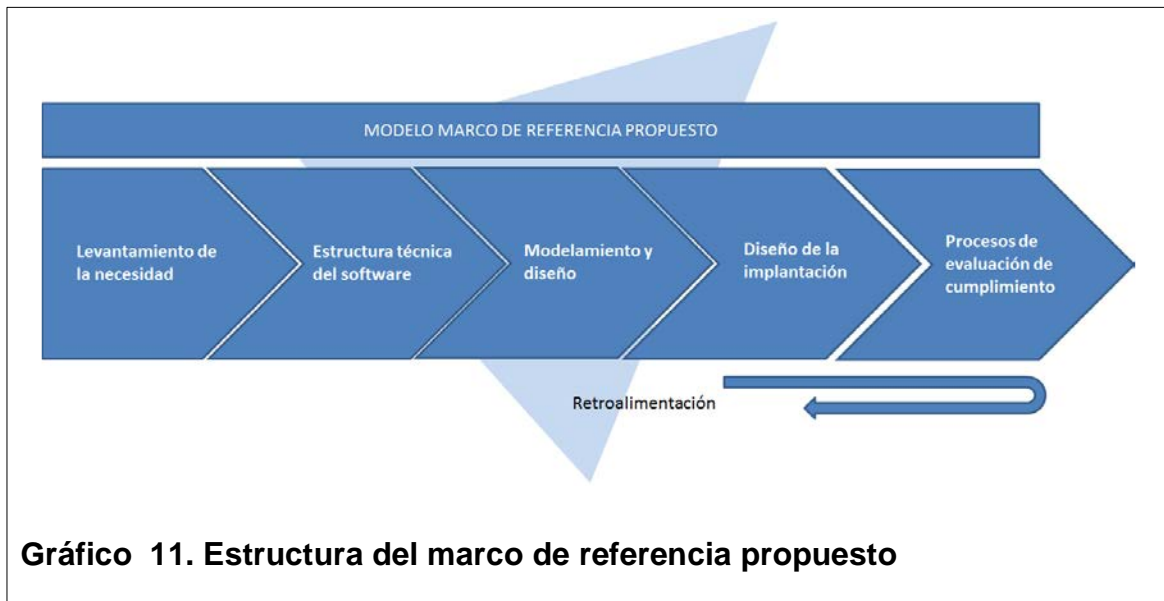
4.2 Estructura del marco de referencia propuesto

En base a los lineamientos expuestos, la estructura del marco de referencia propuesto se conforma por procesos sistemáticos debidamente relacionados que orienten al usuario a cumplir con lineamientos necesarios para el diseño y desarrollo de software.

En este caso, cada usuario tendrá la alternativa de utilizar todo el marco o las áreas de interés acorde a sus necesidades, aspecto que será fácilmente identificable en función de estar conformado por fases claramente delimitadas.

Este marco de referencia no establece por principio herramientas, ya que estas dependerán de quien lo aplique. Lo que se propone son guías de acción para mejorar la calidad del software.

Su desarrollo se representa mediante el siguiente organizador gráfico:



La estructura propuesta se enfoca en todos los aspectos requeridos para el desarrollo del software, brindando una orientación integral que apoye en cada uno de los procesos necesarios desde la propia investigación hasta la obtención del producto. Es importante citar que su desarrollo propone una alternativa innovadora, en la medida que abarca procesos no referenciados pero que son determinantes en el desarrollo de un software.

El marco propuesto tendrá como finalidad la entrega de un software de calidad, basado en un estudio de la necesidad, aplicación técnica y procesos de desarrollo.

4.2.1 Obtención de calidad en el producto a desarrollar

El marco de referencia propuesto comprende varios procedimientos que orientan al usuario para disponer de productos de calidad. Este concepto se establece en base varios factores que se integran para desarrollar un software adecuado y orientado a brindar soluciones a situaciones puntuales. La calidad por lo tanto se establece en la medida que el producto creado cumpla con las siguientes características:

- **Solución a un requerimiento:** El software desarrollado en base al marco de referencia cumplido debe solucionar de mejor manera un requerimiento

que el usuario tiene. En este sentido, el software brinda mecanismos que permiten al usuario mejorar su rendimiento, control y gestión.

- **Respeto al marco legal:** El desarrollo del software y su funcionalidad debe cumplir con todas las normativas vigentes. Es decir su diseño debe respetar el marco jurídico vigente, las políticas y normativas existentes en el mercado, permitiendo su cumplimiento. Su utilización bajo ningún sentido omitirá disposiciones existentes o generará situaciones de evasión u omisión e información.
- **Capacidad de adaptación:** Las aplicaciones deben ser flexibles a determinados ajustes y cambios que se consideren necesarios para que siempre se encuentren alineados a satisfacer las necesidades de los usuarios.

La calidad del software evaluado por los elementos citados es un medio sobre el cual se evaluará el producto, aportando al usuario a mejorar su gestión y conocimientos. Para ello, es importante que se cumpla a cabalidad las actividades de cada fase a fin de que el producto resultante sea eficiente.

4.2.2 Descripción de las fases propuestas

A continuación se describen los procesos que conforman el marco de referencia propuesto siendo esta la guía que cada usuario deberá cumplir para el desarrollo del software. Para disponer de una guía efectiva, cada fase ha sido descrita en base de elementos que permiten su clara comprensión y seguimiento.

4.2.2.1 Levantamiento de la necesidad

La primera fase del marco de referencia propuesto es el levantamiento de la necesidad que evalúa las condiciones de un entorno determinado para identificar el posible desarrollo de un software.

A continuación se describen los procesos de esta fase:

Tabla 8. Descripción de la fase de levantamiento de la necesidad

Factor	Detalle
Objetivo	Identificar necesidades existentes en el entorno que puedan cubrirse con software.
Entregable	Información especializada que permita comprender la necesidad existente.
Técnica	Estudio de campo en el área donde se desarrollará el software
Procedimiento	<p>Para identificar la necesidad en el entorno, se deberá cumplir con las siguientes actividades:</p> <ol style="list-style-type: none"> <li data-bbox="600 779 1358 913">1. Delimitación del entorno: Comprende la identificación del área en donde se desarrollara el software. <li data-bbox="600 981 1358 1070">2. Desarrollo del macro proceso: Delimita los procesos gobernantes, habilitantes y de apoyo. <li data-bbox="600 1137 1358 1272">3. Definición del proceso: Identifica la relación de actividades que se desarrollan señalando las que puedan ser automatizadas <li data-bbox="600 1283 1358 1473">4. Medición de factores: Establece el comportamiento actual de las actividades a fin de que sirvan como parámetros para evaluar la eficiencia del software propuesto.

Guía de cumplimiento de las actividades de la fase:

- **Delimitación del entorno:**

Al ser la cobertura del software amplia, es importante que este quede delimitado a un entorno específico, aspecto que evitará que su cobertura sea demasiado amplia e inviable de aportar soluciones puntuales. Para ello, se ha establecido un marco de referencia que establece de manera clara el ámbito de desarrollo. Este deberá ser utilizado como un checklist que el diseñador seleccionará acorde a la necesidad.

Tabla 9. Delimitación del entorno

Entorno	Descripción
A.1 Empresarial	El software a desarrollar se enfocará a brindar soluciones en áreas relacionadas a la gestión de empresas.
A.1.1 Mercadeo	El software será enfocado en procesos de publicidad, promoción, asistencia, comercialización.
A.1.2 Operaciones	El software será enfocado en procesos de logística, inventarios, distribución.
A.1.3 Producción	El software será enfocado en procesos de transformación de materias primas, seguridad industrial.
A.1.4 Administración	El software será enfocado a procesos de talento humano.
A.1.5 Financiero	El software será enfocado a procesos de contabilidad, tributación, registro de transacciones, cálculo de escenarios, evaluación de rendimiento.
A.2 Salud	El software a desarrollar se enfocará a brindar soluciones a servicios de cuidado de salud.
A.2.1 Salud preventivo	El software se enfoca en brindar servicios para la prevención de enfermedades.
A.2.2 Salud Correctiva	El software se enfoca en brindar servicios para la atención de enfermedades y recuperación del paciente.
A.3 Educación	El software será enfocado a aportar en los procesos académicos de los estudiantes.
A.3.1 Nivel inicial	El software se especializa en procesos de educación inicial.
A.3.2 Nivel Medio	El software se especializa en procesos de educación media.
A.3.3 Nivel Superior	El software se especializa en procesos de educación superior.
A.3.4 Cursos informales	El software se especializa en programas y cursos informales de capacitación.
A.4 Alimentos	
A.4.1 Actividad comercial	El software se especializa en procesos especializados en actividades económicas. Ejemplo restaurantes, farmacias, entre otros.

La delimitación del entorno propuesta puede incrementarse con más categorías, siendo flexible ajustes requeridos siendo este un lineamiento anteriormente citado.

Es importante citar que la delimitación del entorno puede combinar varias categorías acorde a la necesidad existente, debiendo esto marcarse a fin de tener una visión clara de la cobertura que el software ofertará.

- **Desarrollo del macro proceso**

Una vez delimitado el entorno donde se desarrollará el software es importante iniciar con el levantamiento de los procesos en los cuales funcionará. Para ello, se propone establecer un marco de referencia especializado en procesos, en donde se identifiquen tres niveles:

- **Procesos Gobernantes:**

Determinan quienes serán responsables del proceso en donde se desarrolla el software.

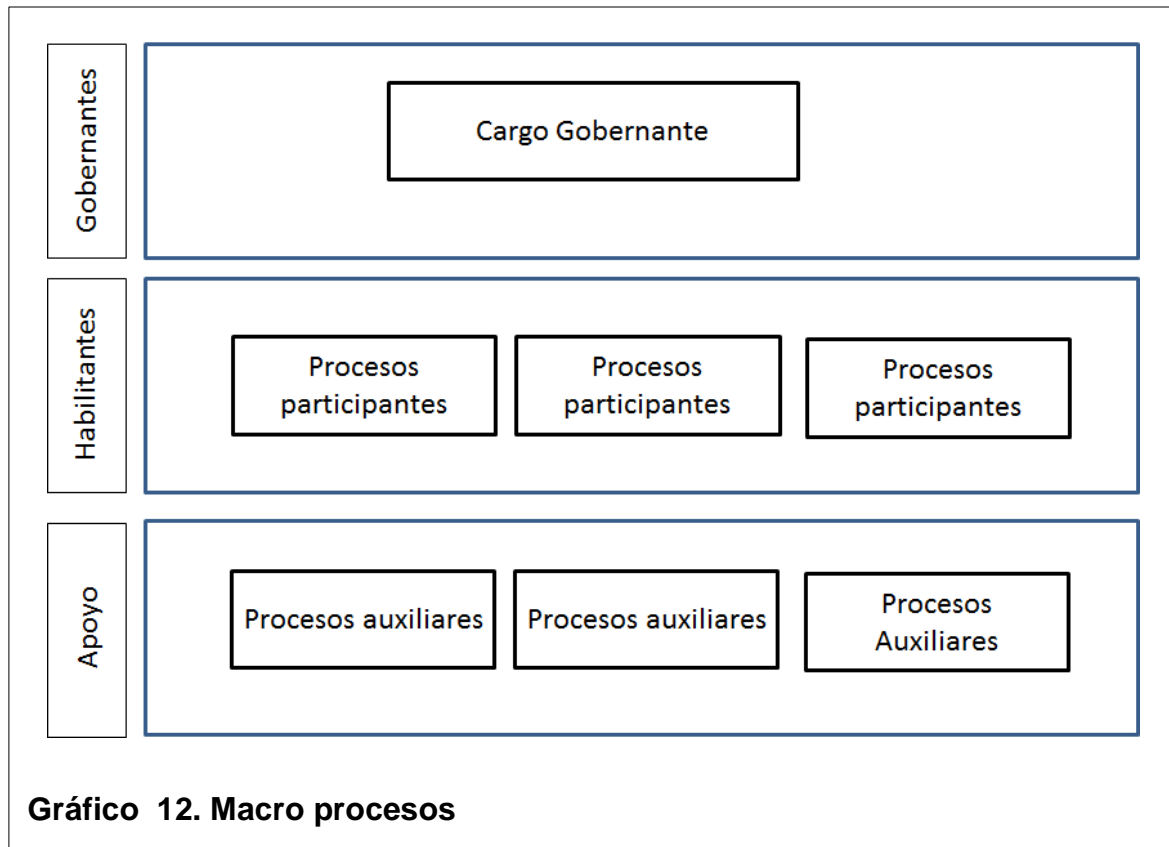
- **Procesos Habilitantes:**

Determinan los procesos participantes en los que se desarrollará el software

- **Procesos de Apoyo:**

Determina los procesos que apoyan con insumos requeridos para el cumplimiento de los procesos del software.

La determinación de los procesos permitirá tener una visión clara de los permisos de uso de las diferentes aplicaciones creadas, determinando aspectos relacionados a la seguridad en base a las aplicaciones e informes que el software generara. Para su desarrollo, es importante realizar un mapeo, para lo cual se propone el siguiente formato basado en macro procesos:



- **Definición del proceso:**

Definidos y clasificados los procesos es importante realizar un levantamiento profundo de las actividades que conforman el proceso identificando aquellos que van a ser apoyados con el diseño del software.

Se propone utilizar la técnica de flujos para que los procesos desarrollados sean visibles. Para ello, se define los siguientes conceptos gráficos:



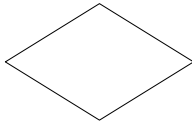
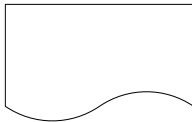

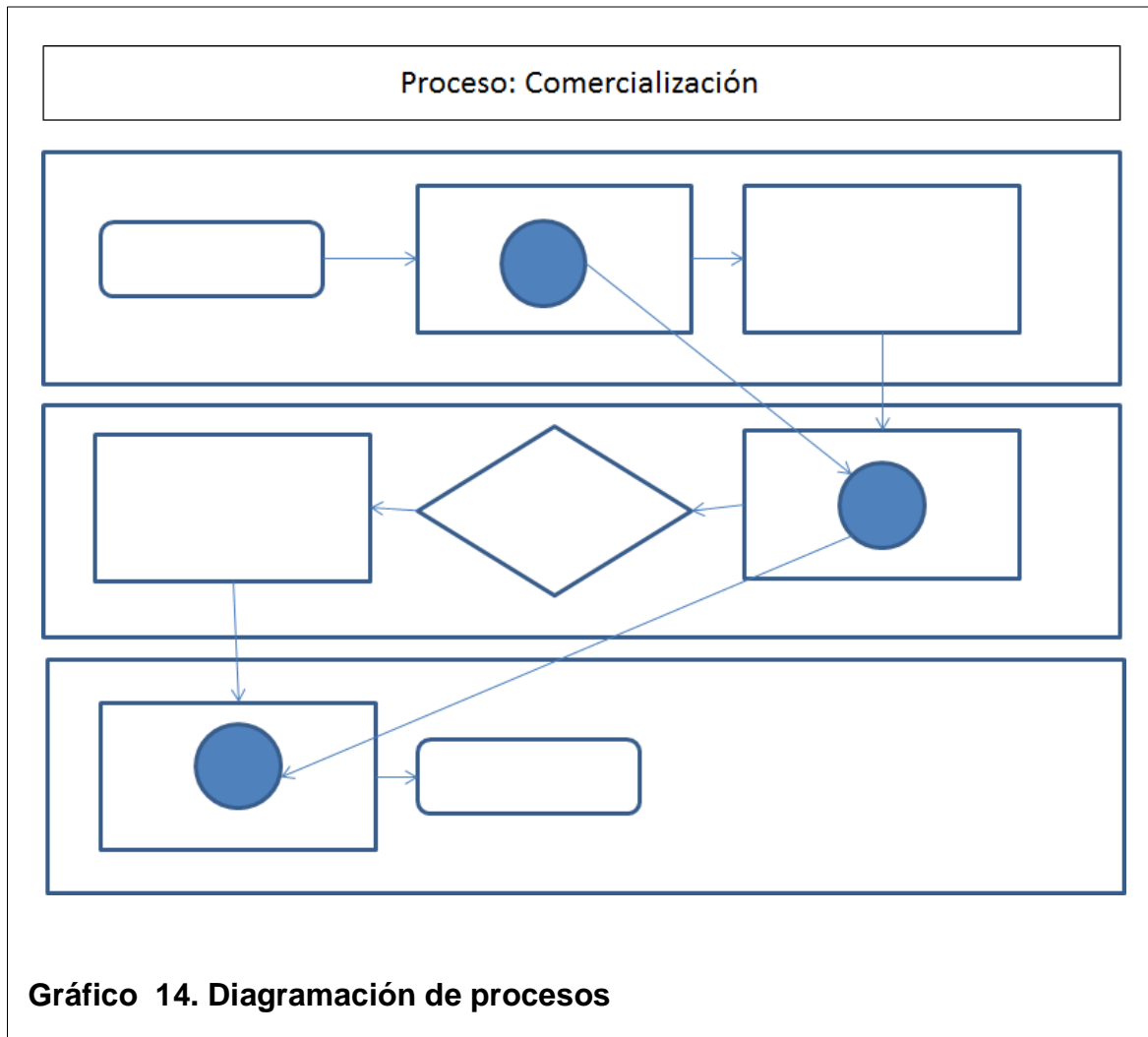
SÍMBOLO	DESCRIPCIÓN
	<p>Inicio/terminación: Representa el origen de una entrada o el destino de un resultado o salida, es utilizado para representar el inicio o la finalización de un conjunto de actividades</p>
	<p>Proceso: Representa una actividad o un conjunto de actividades</p>
	<p>Decisión: Representa una decisión</p>
	<p>Documento: Representa documentación</p>
	<p>Representa la secuencia en que se ejecutan las actividades y el flujo de productos, información, etc.</p>

Gráfico 13. Conceptos de diagramas

Diagramados los flujos se debe establecer la ruta de software que defina la cobertura de las actividades cumplidas. A continuación se presenta un ejemplo:



Los círculos azules muestran los procesos en los cuales existen aplicaciones de software que aportan con su desarrollo. En este sentido, se observa como permite relacionar actividades facilitando su desarrollo.

- **Medición de factores:**

Las actividades que serán apoyadas con el software deben ser evaluadas a fin de tener datos que posteriormente sirvan como parámetros de referencia de los beneficios provocados con el producto.

En base a las actividades seleccionadas es importante identificar las variables que participan en la ejecución del proceso, para luego proceder a levantar datos. No existe un número determinado mínimo por variable, no obstante para disponer

de información confiable esta tiene que realizarse por lo menos 50 veces en días diferentes por lo menos en una semana. Las variables posibles a ser utilizadas son:

- Tiempo: Duración del proceso
- Costo: Valor de insumos directos involucrados en el proceso
- Gasto: Valor de recursos indirectos involucrados en el proceso
- Fallas: Errores comunes presentados en el proceso

Para levantar los datos de las variables seleccionadas se presenta el siguiente cuadro de datos que se propone utilizar:

Tabla 10. Matriz para expresión de datos

Variable _____

Período	Valor

Los datos levantados deben necesariamente ser procesados a fin de que brinden información sobre el rendimiento de los procesos antes de la implantación del sistema. Para ello se propone dos procesos:

- **Evaluación de rangos de comportamiento**

Debido a que los datos mantienen una variabilidad por medición es importante identificar los rangos más comunes de comportamiento. Para ello se debe aplicar procesos estadísticos basados en la desviación estándar.

“La desviación estándar es una medida de dispersión para variables de razón y de intervalo que permite entender el comportamiento de un conjunto de datos en un tiempo determinado.” (Tomás, 2012, p. 31)

Su desarrollo se efectuará en base a la siguiente ecuación:

Ecuación No. 1 Desviación Estándar

$$s^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$$

Tomado de: (Moore, 2009, p. 56)

Donde,

S^2 = Desviación estándar

X_i = Valor de la variable

X = Promedio

N = Número de datos

- **Relación del comportamiento de las variables**

Debido a que el comportamiento de una variable puede incidir en otras, es importante marcar una relación entre las mismas. Para ello se propone utilizar procesos estadísticos relacionados a la correlación.

Ecuación No. 2 Correlación de Pearson

$$\rho_{X,Y} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$$

Tomado de: (Rodríguez C. , 2009, p. 212)

Donde,

σ_{XY} es la covarianza de (X, Y)

σ_X es la desviación típica de la variable X

σ_Y es la desviación típica de la variable Y

El cálculo de la correlación de Pearson entregará como resultado un valor comprendido entre -1 y 1, debiendo interpretarse sus resultados de la siguiente manera:

Tabla 11. Tabla de interpretación de los resultados

RESULTADO DEL COEFICIENTE DE CORRELACIÓN	TIPO DE RELACIÓN
$r=0$	No existe ninguna correlación
$r=1$	Existe una correlación positiva perfecta
$0 < r < 1$	Existe una correlación positiva
$r = -1$	Existe una correlación negativa perfecta
$-1 < r < 0$	Existe una correlación negativa

Tomado de: Enríquez, M. (2009). Modelos matemáticos en los negocios. México-México: Centrum.

Para describir mejor los procesos referenciados, se presenta el siguiente ejemplo guía:

Tabla 12. Cálculo de rangos por variable

Muestra	Tiempo (Seg)	Costo (USD)
1	10	\$ 98,00
2	11	\$ 98,89
3	10,98	\$ 98,00
4	11,9	\$ 98,99
5	11,89	\$ 101,12
6	12	\$ 112,90
7	12,1	\$ 113,00
8	12,11	\$ 113,88
9	12,12	\$ 113,91
10	12,13	\$ 114,00

Muestra	Tiempo (Seg) X	X-XPROM	(X-XPROM) ²
1	10	-1,623	2,63
2	11	-0,623	0,39
3	10,98	-0,643	0,41
4	11,9	0,277	0,08
5	11,89	0,267	0,07
6	12	0,377	0,14
7	12,1	0,477	0,23
8	12,11	0,487	0,24
9	12,12	0,497	0,25
10	12,13	0,507	0,26
XPROM	11,623	SUMA	4,69

Desviación	2,17
Rango Mayor	13,79
Rango Menor	9,46

Muestra	Costo (Usd)	X-XPROM	(X-XPROM) ²
1	\$ 98,00	-\$ 8,27	68,38
2	\$ 98,89	-\$ 7,38	54,45
3	\$ 98,00	-\$ 8,27	68,38
4	\$ 98,99	-\$ 7,28	52,98
5	\$ 101,12	-\$ 5,15	26,51
6	\$ 112,90	\$ 6,63	43,97
7	\$ 113,00	\$ 6,73	45,31
8	\$ 113,88	\$ 7,61	57,93
9	\$ 113,91	\$ 7,64	58,38
10	\$ 114,00	\$ 7,73	59,77
XPROM	\$ 106,27	SUMA	536,06

Desviación	23,15
Rango Mayor	129,42
Rango Menor	83,12

Conforme se observa en los cálculos los rangos de comportamiento del tiempo del proceso evaluado se encuentra entre 9,46 a 13,79 segundos. El costo entre 83,12 a 129,42 USD. El proceso ha permitido disponer de información que ayuda a su comprensión y posteriormente permitirá comprobar los beneficios alcanzados con el software.

A continuación se procede a relacionar las variables:

$$S_{XY} = \frac{1}{n} \sum_{i=1}^n x_i y_i - \overline{XY}$$

$$S_X^2 = \frac{1}{n} \sum_{i=1}^n x_i^2 - \overline{X}^2$$

$$S_Y^2 = \frac{1}{n} \sum_{i=1}^n y_i^2 - \overline{Y}^2$$

$$r = \frac{S_{XY}}{S_X S_Y}$$

Donde,

S_{XY} = Covarianza

S^2_x = Varianza Marginal al cuadrado de la variable x

S^2_y = Varianza Marginal al cuadrado de la variable y

r = Correlación de Pearson

Tabla 13. Cálculo de relación entre variables

Muestra	Tiempo (Seg) X	Costo (USD) Y	x_i^2	y_i^2	$x_i y_i$
1	10	98	100	9604	980
2	11	98,89	121	9779,2321	1087,79
3	10,98	98	120,5604	9604	1076,04
4	11,9	98,99	141,61	9799,0201	1177,981
5	11,89	101,12	141,3721	10225,2544	1202,3168
6	12	112,9	144	12746,41	1354,8
7	12,1	113	146,41	12769	1367,3
8	12,11	113,88	146,6521	12968,6544	1379,0868
9	12,12	113,91	146,8944	12975,4881	1380,5892
10	12,13	114	147,1369	12996	1382,82
Suma	116,23	1062,69	1355,6359	113467,059	12388,724
Promedio	11,623	106,269	135,56359	11346,7059	1238,8724

SXY	3,707793
S^2X	0,469461
S^2Y	53,605549
SX	0,68517224
SY	7,32158105
r	0,73911301

El valor obtenido de 0,7 señala que existe correlación positiva entre ambas variables. En este caso se entiende que si aumenta el tiempo, aumentará el costo.

En base a lo obtenido se detalla lo siguiente:

- Los outliers se reflejan en el comportamiento de cada variable por lo que están dentro del estudio.
- Las líneas de control interno se realizan al proceso que da como resultado el comportamiento de cada variable.
- El control del proceso determina la variabilidad en el comportamiento de cada variable.

4.2.2.2 Estructura técnica del software

Todo software antes de iniciar su desarrollo debe determinar anticipadamente las herramientas y recursos que van a ser utilizados. Esto determinará los costos necesarios a incurrirse permitiendo que los responsables tomen la decisión de ejecución del mismo.

El presente marco de referencia propone la definición técnica en base a dos elementos básicos; el front end y el back end. En cada uno de estos es importante establecer los siguientes recursos:

Tabla 14. Estructura técnica del software

Área: Front End

Recurso Humano	Cantidad	Costo Mensual	Tiempo (Mes)	Total
Recurso Tecnológico	Cantidad	Costo Mensual	Tiempo (Mes)	Total
Recurso Técnico	Cantidad	Costo Mensual	Tiempo (Mes)	Total
Total				

Área: Back End

Recurso Humano	Cantidad	Costo Mensual	Tiempo (Mes)	Total
Recurso Tecnológico	Cantidad	Costo Mensual	Tiempo (Mes)	Total
Recurso Técnico	Cantidad	Costo Mensual	Tiempo (Mes)	Total
Total				

Total Front End				
Total Back End				
Total				

El proceso de definición de la estructura técnica debe realizarse en base de las siguientes normativas:

- La descripción de la estructura técnica debe realizarse de manera independiente tanto para el front end como para el back end.
- Debe establecerse una temporalidad evaluada en una unidad uniforme. Se propone que sea mensual.
- Se debe indicar la cantidad de recursos requeridos identificados en base a la siguiente clasificación:
 - Recursos humanos: Personal encargado del desarrollo del software. El equipo debe conformarse por los elementos citados en la tabla No.15. (En estos se encuentran las funciones de liderazgo y control).
 - Recursos tecnológicos: Software de desarrollo a utilizarse. En el caso de que sea un software privado deberá incluirse los valores del licenciamiento de uso.
 - Recursos técnicos: Equipos a utilizarse para el desarrollo (Computador, Impresora, entre otros).
- Los costos serán presentados en dólares americanos.
- Se calculará el total clasificado por el rubro front end y back end.
- Debe establecerse la estructura orgánica responsable del diseño de software.

Como se indicó en los procesos citados, el diseño de software requiere de una estructura orgánica responsable del cumplimiento del presente marco de referencia. Este procedimiento es flexible pudiendo el diseñador conformar su estructura acorde a su capacidad y tipo de producto a desarrollar, sin embargo debe incorporar funciones específicas las cuales se describen en el siguiente cuadro:

Tabla 15. Organización para el desarrollo del software

Cargo	Descripción del Cargo	Competencias
Jefe de Proyectos	Encargado de todos los procesos de diseño y desarrollo del software. Es el líder del proyecto.	Liderazgo, comunicación constante y efectiva, planificación de proyectos, firmeza en la toma de decisiones, agilidad, planteamiento de objetivos.
Tesorero	Encargado de la administración de los recursos monetarios asignados para el desarrollo el software.	Optimización del uso de recursos, elaboración del presupuesto.
Timekeeper	Encargado del control de la planificación desarrollada para el desarrollo del software.	Seguimiento constante de los tiempos para la entrega de proyectos, implantación de acciones correctivas en el caso de incumplimiento de tiempos.
Arquitecto	Encargado de definir todo el diseño de la aplicación desde el front end hasta el back end. Definición del manejo de datos, tecnología, lenguajes de programación y tipos de comunicación que se soportará.	Completo conocimiento técnico de back end, middleware y front end.
Control de Calidad	Encargado de la validación y verificación de la calidad en el producto desarrollado.	Conocimiento de las normas de calidad, estándares de experiencia de usuario, funcionamiento de la aplicación.
Programador Front End	Grupo de programadores responsables del diseño y desarrollo del front end.	Conocimiento técnico de programación para desarrollo de interfaces para usuarios, herramientas de desarrollo, estándares de visualización.

Programador Back End	Grupo de programadores responsables del diseño, desarrollo y mantenimiento del back end.	Conocimiento técnico y de optimización de base de datos y servidores.
Programador Middleware	Grupo de programadores responsables de la inserción, actualización y eliminación de datos mediante el uso de la información enviada desde front end.	Conocimiento técnico de programación a nivel de manejo de datos (distintos lenguajes de programación, persistencia y consistencia en la información).
Diseñador gráfico	Encargado del diseño visual y gráfico del software.	Conocimiento técnico de estándares de experiencia de usuario, diseño visual y gráfico.
Funcionales	Encargado del desarrollo de los recursos complementarios como manuales de usuario, manual técnico, videos, etc.	Conocimiento funcional de la aplicación.
Instaladores	Grupo profesional encargado del proceso de instalación y prueba del software.	Conocimiento técnico de hardware y el software desarrollado.
Capacitador	Grupo profesional encargado de la capacitación a los usuarios del software.	Conocimiento técnico de hardware, software y funcional de la aplicación.

4.2.2.3 Modelamiento y diseño

El modelamiento y diseño comprende el cuerpo del software debiendo este acorde a los lineamientos presentados ser desarrollado mediante procesos modulares. Su gestión permitirá que su cobertura pueda expandirse o contraerse según las necesidades existentes, disponiendo de la suficiente flexibilidad y adaptabilidad al usuario. Desde el punto de vista técnico, el programador podrá continuar su trabajo progresivamente, poniendo en ejecución los módulos terminados.

En este caso, el marco de referencia establece la construcción de un software definido por los siguientes tipos de módulos que se describen a continuación:

Tabla 16. Estructura modular

Módulo	Descripción	Estado
Seguridad	Autoriza el ingreso a los usuarios para poder operar los servicios del software	Opcional
Introductorio	Presenta la imagen central del software por algunos segundos como elemento introductorio de ingreso	Opcional
MDI-Módulo Central	Establece la plataforma principal que permite la selección del módulo requerido y autorizado según el tipo de usuario	Requerido
Módulos	Presenta los servicios del software en módulos especializados	Requerido
Módulo de Créditos	Presenta información del equipo de trabajo, empresa, licencia de uso y responsables	Requerido

Nota: Una misma persona puede realizar varios de los cargos detallados anteriormente dependiendo del tamaño y la adaptabilidad de la empresa.

Existen ciertas normativas que deben ser cumplidas para que el software desarrollado sea accesible para el usuario, mismas que se describen a continuación:

Aspectos técnicos:

Front end:

- Debe mantenerse una codificación para nombrar a cada objeto utilizado en el diseño del software. El código será conformado de la siguiente manera: Siglas del nombre del objeto, campo de cumplimiento, número de formulario. A continuación se expone el código de los objetos comunes a utilizarse:

Tabla 17. Codificación de objetos en el software

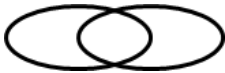

Objeto	Código	Ejemplo
Formulario	frmnombreúmero	Frminicio1
Caja de texto o etiqueta	txtnombreúmero	Txtnombre1
Botón	btnnombreúmero	Btngresar1
Check de verificación	chknombreúmero	Chkopcion1
List box	lstnombreúmero	Lstopcion1
Marco o frame	frmnombreúmero	Frmdatos1
Option button	optnombreúmero	Optopcion1
Barras de desplazamiento	hsrnombreúmero vsrnombreúmero	Hsrbarra1 Vsrbarra1
Cuadro de imagen	imgnombreúmero	Imgfoto1
Combo box	cmbnombreúmero	Cmbopciones1
Data object	datnombreúmero	Datsis1

- Los nombres en el software de programación seleccionado serán en minúsculas sin tilde y sin la letra “ñ”.
- No puede nombrarse dos objetos con el mismo nombre.
- Debe emitirse un listado con los nombres técnicos por cada formulario.

Back end:

- Debe manejarse una entidad relación en la estructura de datos. Esta puede ser de los siguientes tipos:
 - Uno a Uno: Un registro único en dos tablas. Generalmente este capo será la clave principal de las tablas relacionadas
 - Uno a Varios: Un registro único en una tabla relacionada con el mismo registro que permite varios ingresos
 - Varios a varios: Varios registros en una tabla relacionada con varios registros en otras.
 - La nomenclatura de la relación se describe en la siguiente tabla:

Tabla 18. Nomenclatura de relación

Nomenclatura	Descripción
	Varios registros
1	Un registro
	Secuencia de relación

- La relación en la base de datos puede presentarse de manera visual o mediante codificación.

Aspectos de diseño:

- El diseño debe contar con herramientas de navegación en cada módulo o pantalla permitiendo al usuario los siguientes desplazamientos:
 - Navegación siguiente pantalla: Permite al usuario avanzar a la siguiente pantalla.
 - Navegación inicio: Permite al usuario ir al módulo principal MDI
 - Navegación atrás: Permite al usuario regresar a pantallas anteriores
- No puede utilizarse colores fuertes de fondo de cada pantalla para evitar cansancio o fatiga en el usuario.
- No utilizar marquesinas en las pantallas
- Manejar una iconografía estandarizada para los botones que se utilizan en el software. Su definición es previa al inicio de la programación y depende del tipo de aplicación que se quiere realizar. Se puede utilizar iconografía estándar disponible en las aplicaciones propias de software en la librería de imágenes.
- El uso de variables será mediante vocales y números de la siguiente manera a1. Se deberá desarrollar en el manual técnico el detalle de la capacidad de almacenamiento de las variables.
- En el software se incluirán comentarios sobre la descripción de cada variable. Ejemplo: a1= tiempo.

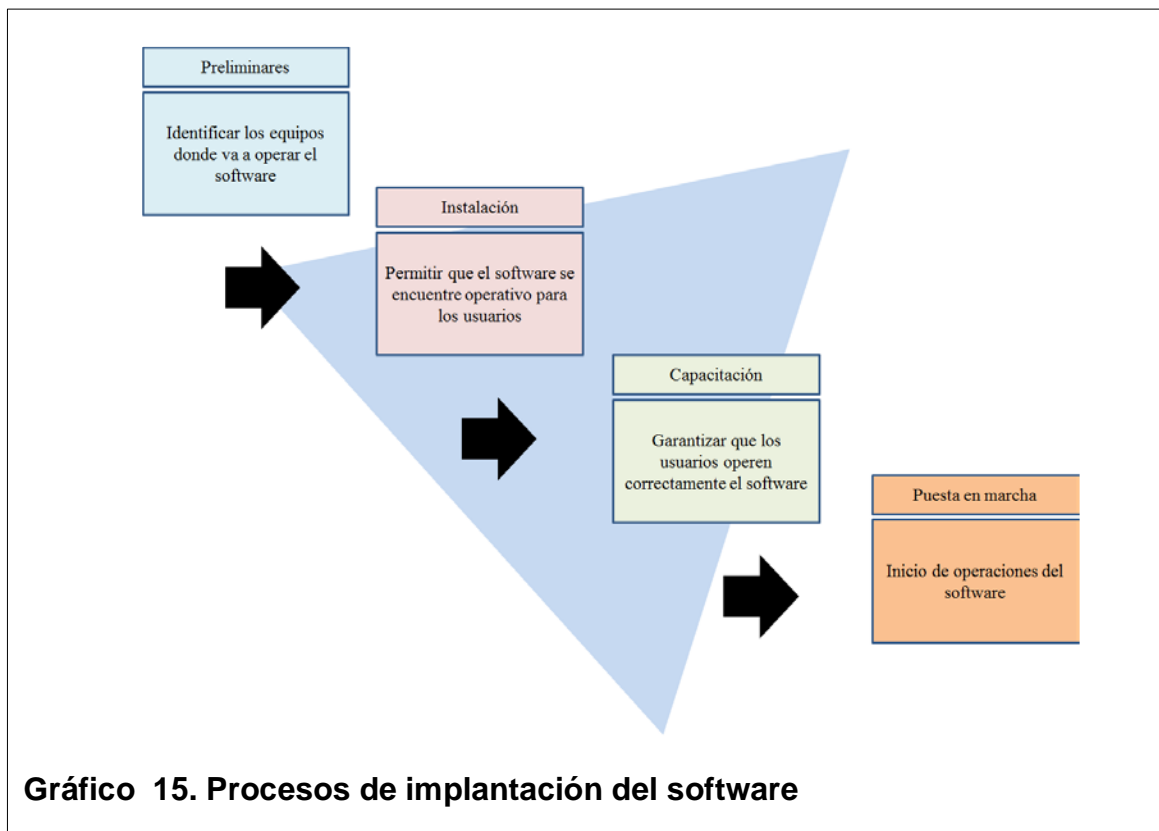
Una vez concluida la fase de programación se deberá cumplir con los siguientes procesos:

- Prueba de funcionamiento. Se nombrará un grupo focal con usuarios con perfiles similares a los destinatarios para someter el uso del software.
- Los procesos de registro de la funcionalidad se evaluarán en base al uso del siguiente formulario. (Se describe en la fase de evaluación)
- Los registros serán analizados por la organización de desarrollo y se establecerá el plan de mejoras del software.

4.2.2.4 Diseño de la implantación

Concluido el desarrollo del software debe iniciarse el proceso de la implantación que permite su operatividad en el lugar donde este se requiere. Este proceso varía en base a la complejidad del software pudiendo ser ejecutado por el propio usuario o por personal técnico.

Los procesos requeridos para la implantación se describen en el siguiente organizador gráfico:



- **Preliminares**

En primera instancia es importante realizar un reconocimiento de la infraestructura técnica disponible donde va operar el software para determinar si esta es adecuada para su funcionamiento. Esta actividad se desarrollará en la medida que el software requiera de una instalación intranet exclusivamente como se detallará más adelante.

Una vez levantada la infraestructura técnica se describe los elementos faltantes requeridos para que se determine la viabilidad de compra de la empresa. En la medida de capacidad de compra se determina la capacidad de cobertura de las aplicaciones a desarrollar.

El levantamiento del inventario deberá identificar los siguientes aspectos:

Tabla 19. Levantamiento de recursos disponibles

Código	Equipo	Funcionalidad (1 Adecuado, 0 Inadecuado)

El inventario deberá evaluar los recursos en base a los siguientes elementos:

- Equipos de computación: Equipos, servidor, impresoras, wireless.
- Red: Conectividad, tipo de cableado, puntos de red, seguridades.

Se deberá emitir un informe que describa los recursos identificados señalando si estos proceden para la instalación del software respaldando su seguridad y adecuado funcionamiento.

- **Instalación:**

El avance de la tecnología brinda diferentes mecanismos de implantación que responden al tipo de software, cobertura y demás. Este puede principalmente ser ejecutado de las siguientes maneras:

- Online: Cuando el software corre online mediante el servicio de Internet.
- Intranet: Cuando se crea una red interna en donde se instala la aplicación
- Escritorio: Cuando es instalado en un equipo determinado.

Como se indicó anteriormente, dentro de las normas del marco de referencia propuesto se incluye el desarrollo de manuales técnicos y de usuario en donde se incluirá los procesos e instalación y uso. El software online no requiere de un proceso de instalación, debido a que este corre en línea, el cual se lo debe instalar previamente una sola vez. En el caso del software de escritorio este puede ser ejecutado por el usuario. En el caso del software intranet demanda de una instalación y revisión de funcionamiento a cargo del personal técnico.

- **Capacitación:**

La capacitación deberá ser definida en base al uso de varios recursos que deben ser descritos. En este sentido podrá ser:

- **Capacitación física:** Cuando se establece una reunión programada con uno o varios tutores asignados que capacitan sobre el uso del software.
- **Capacitación online:** Cuando esta se desarrollará mediante el uso del internet o mediante videos definidos.
- **Capacitación por manuales:** Cuando la capacitación es responsabilidad del usuario mediante el uso del material creado.

- **Puesta en marcha**

Se debe determinar el arranque del sistema, siendo este el día en que entra en operación.

4.2.2.5 Procesos de evaluación del cumplimiento

Los procesos descritos requieren de una constante evaluación que permita mejorar permanentemente el software desarrollado. El marco de referencia propuesto establece tres niveles de evaluación que deben ser ejecutados de manera obligatoria. Estos son los siguientes:

- **Definición de la medición de la satisfacción del cliente**

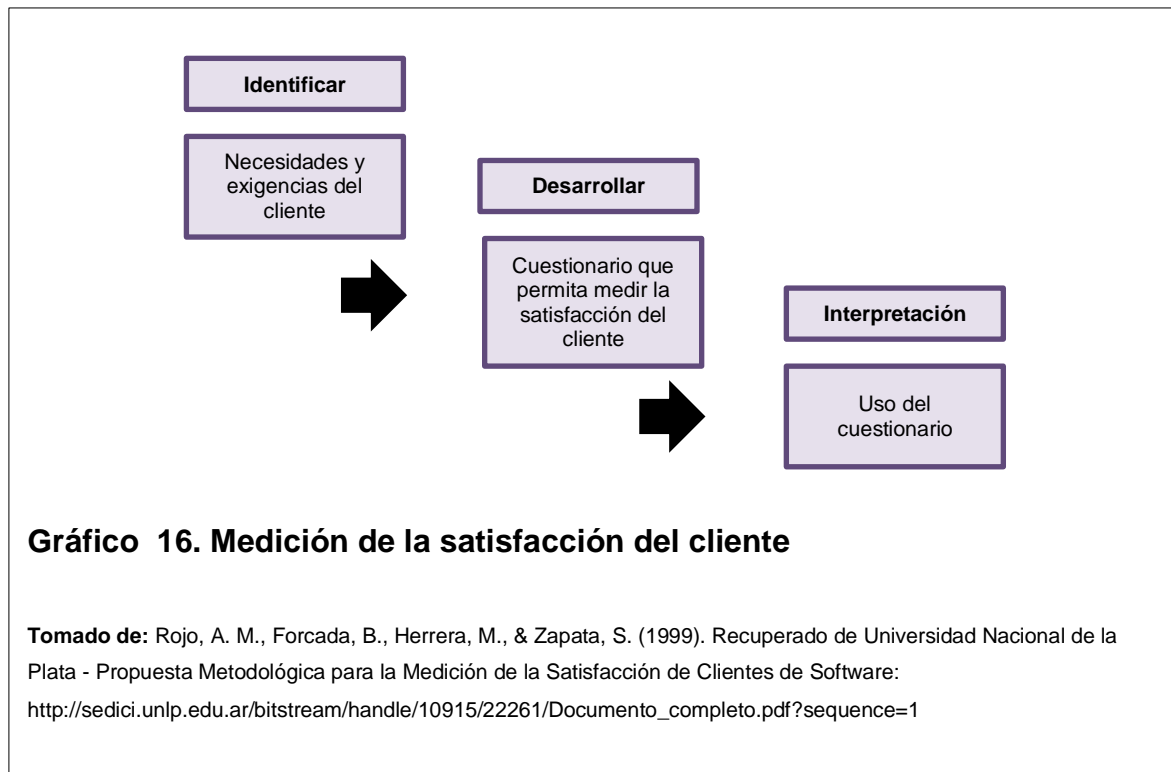
La medición de la satisfacción del cliente va a tener dos partes, la definición de los resultados y las encuestas de satisfacción que van a ayudar a determinar si se cumplió con las expectativas del cliente, éstas se usarán de manera informativa.

1. Definición de Resultados

Una vez identificadas las variables que afectan al proceso mediante el cálculo del Coeficiente de Pearson. Se deberá incluir puntos de control con la finalidad de disminuir el impacto de los mismos y mejorar la eficiencia en el proceso. Los resultados obtenidos en el proceso se deberán comparar con proyecto de complejidad similar, donde fácilmente se pueda identificar la satisfacción del cliente mediante la reducción de errores cometidos o tiempo de respuesta que se acordó con el mismo.

2. Encuestas de Satisfacción

Para realizar la medición de satisfacción del cliente, es necesario determinar cuáles son sus necesidades y exigencias, esto, mediante un cuestionario, tomando en cuenta tres puntos importantes:



o **Identificar necesidades y exigencias del cliente**

Se identifican, definen y se establecen los límites de calidad del servicio o producto de acuerdo a las necesidades y exigencias de los clientes. Es importante que también se establezcan los límites de calidad de los proveedores, ya que de estos dependerá parte de la calidad que se va a entregar al cliente.

Para identificar de una mejor manera, es aconsejable que personal de la empresa ayude en este análisis, compartiendo sus vivencias y casos reales que han surgido, de esta manera se darán pautas claves para delimitar la calidad que se va a entregar al cliente.

o **Desarrollar cuestionario que permite medir la satisfacción del cliente**

Los cuestionarios deben crearse siguiendo los siguientes pasos:

- Determinar la dimensión del contenido del cuestionario.
- Introducción del cuestionario: se define el objetivo del cuestionario y se indica las instrucciones del mismo.
- Establecer las preguntas: deben ser precisas, concisas y que no admita confusiones o equivocaciones.

- Tipo de preguntas: se define el formato que van a tener las preguntas. Se puede usar los siguientes tipos de respuestas:
 - Abiertas: este tipo de preguntas permite al encuestado expresarse ampliamente y dar una mejor retroalimentación en el caso que no se haya considerado algún tema específico.
 - Cerradas: este tipo de pregunta permite limitar el número de opciones de respuesta y centrarse en un contexto específico.
 - ❖ Selección múltiple: se puede escoger varias respuestas.
 - ❖ Selección única: se puede escoger una única respuesta. A continuación se mencionan algunas que podrían utilizarse:
 - Likert: se especifica el nivel de acuerdo o desacuerdo y se presentan normalmente de 5 a 10 niveles de respuesta.
 - Dicotómicas: presentan dos opciones de respuesta.
 - Semi-abiertas o semi-cerradas: este tipo de pregunta permite al entrevistado escoger entre una opción cerrada o proporcionar una respuesta diferente a las determinadas.

Es importante mencionar que dependerá de cada empresa la forma en que se realice la encuesta, esta puede ser en línea o en papel. Lo anteriormente mencionado también dependerán de la manera en que se vaya a analizar y elaborar los resultados.

- **Uso del cuestionario**

En este último paso se debe analizar e interpretar los resultados obtenidos, se determinará cual es la necesidad más importante para el cliente y si su nivel de satisfacción cumple los límites establecidos. Es aquí donde se debe entender cuál es el punto más importante que está relacionado directamente con la satisfacción general del cliente, para poder orientar hacia estos las estrategias y objetivos de la empresa.

- **Evaluación de prueba de software**

Esta evaluación se efectuará antes de la implantación y se desarrolla mediante la conformación de grupos focales con perfiles similares de los usuarios potenciales que van a utilizar la aplicación y prueban la aplicación para determinar si esta

resuelve los requerimientos exigidos que motivaron su realización. Su desarrollo se efectuará en base al siguiente formulario:

Tabla 20. Formulario de registro de prueba del software

Proceso de registro de prueba del software		
Fecha:		
Evaluación		
Variable	Registro	Calificación (3 Muy Bueno, 1 Bueno, 0 no procede)
Acceso	Comentar sobre el acceso al software y la identificación de los módulos de servicio.	
Reconocimiento	Identificación de los servicios del software.	
Solución	El software utilizado contribuyó a la solución de los requerimientos del usuario.	
Accesorios	El software cuenta con los recursos de orientación para el uso.	
Diseño	El software es fácil de entender, mantiene una adecuada distribución de los servicios para su uso.	
Información	El software provee de la información requerida para su desarrollo.	
Seguridad	El software brinda la seguridad necesaria para mejorar los datos ingresados.	
Cobertura	El software comprende los servicios que se requieren para el proceso en donde funciona.	
Cálculo	Los resultados emitidos por el software son correctos.	
Recursos	Se cuenta con recursos de consulta que permiten el adecuado uso del software.	

Los parámetros de evaluación establecen los siguientes criterios:

Calificación: 3, la aplicación satisface los requerimientos del usuario, comprendiendo su uso y resultados entregables.

Calificación: 2, la aplicación satisface parcialmente los requerimientos del usuario, existiendo procesos que no se conoce como se utilizan o no se entienden los entregables.

Calificación: 1, la aplicación no satisface los requerimientos de los usuarios.

La información levantada en este proceso deberá generar planes de perfeccionamiento del software ajustando los aspectos que se consideren necesarios a fin de que pueda ser operativo y útil.

- **Evaluación de funcionamiento**

En el proceso de planeación se debió haber definido los objetivos SMART (específicos, medibles, alcanzables, orientado a resultados y fecha límite de ejecución) mismos que serán respaldados con indicadores (Tabla 23), y que tendrán un enfoque hacia la satisfacción del cliente, eficiencia y eficacia. Es importante que se establezca un proceso de evaluación periódico una vez que el software se encuentre en operación.

Su desarrollo deberá basarse en los siguientes lineamientos:

Tabla 21. Proceso de evaluación de funcionamiento

Aspecto	Descripción
¿Qué Evaluar?	El nivel de satisfacción del usuario del software.
¿Por qué Evaluar?	Para determinar si el software aporta a mejorar la calidad de los servicios
¿Para qué Evaluar?	Para identificar debilidades que demanden de mayor atención y perfeccionamiento del software
¿Con que Criterio Evaluar?	Con el criterio de satisfacción del usuario desde el punto de vista del usuario como los módulos presentados aportan a su satisfacción
¿Quién evalúa?	El Dpto. de Asistencia Permanente
¿Cómo Evaluar?	Mediante el uso de indicadores definidos según el tipo de software
¿Con que Evaluar	Con las herramientas tecnológicas desarrolladas.

A continuación se presentan indicadores generales de ejemplo para orientar el proceso de evaluación. Estos indicadores dependerán de cómo la empresa quiere evaluarlos y de sus límites de control definidos.

Tabla 22. Indicadores

Indicador	Objetivo	Fórmula de Cálculo	Frecuencia
Control de implementación	Cumplir con los tiempos requeridos para la implementación del software	$\text{Tiempo Cumplido} = \frac{\text{Tiempo Real}}{\text{Tiempo Presupuestado}}$	Semanal
Satisfacción del usuario	Determinar si el usuario se encuentra satisfecho con el servicio y apoyo recibido.	$\text{Satisfacción} = \frac{\text{Percepción Recibida}}{\text{Percepción Esperada}}$ $\text{Satisfacción} = \frac{\text{Errores cometidos}}{\text{Tiempo gastado}}$	Mensual
Tiempo de atención al usuario	Determinar si la respuesta al usuario es menor al proceso anterior sin automatización	$\text{Tiempo} = \frac{\text{Tiempo de Respuesta}}{\text{Tiempo de Requerimiento}}$ $\text{Tiempo} = \frac{\text{Tiempo de respuesta actual}}{\text{Tiempo de respuesta anterior}}$	Mensual
Costos Incurridos en el proceso	Determinar si los costos incurridos están dentro de los presupuestados. Esto se efectúa cuando se evaluaron los recursos requeridos por la aplicación para su operatividad.	$\text{Costos} = \frac{\text{Costos Incurridos}}{\text{Costos presupuestados}}$	Mensual
Ahorro en costos y gastos	Determinar si el software contribuyo a reducir costos y gastos	$\text{Costos} = \frac{\text{Costos actuales}}{\text{costos anteriores}}$	Mensual

Es importante que los indicadores sean ajustados en base del tipo de software creado, mismos que será definidos con anticipación. La información levantada de su aplicación permitirá mejorar progresivamente el producto.

4.2.2.6 Mejora Continua (Retroalimentación)

El marco de referencia propuesto tendrá una mejora continua que se la realizará en base a la medición de la satisfacción del cliente, el cual será el principal insumo de retroalimentación para identificar cuáles son los procesos que se deben ir mejorando en el tiempo para que la creación del software cumpla las expectativas del cliente. También se considerará los indicadores presentados en la tabla anterior para medir la satisfacción del cliente.

Este proceso de retroalimentación se lo efectuará al finalizar el proceso de desarrollo y entregar el software al cliente.

4.3 Planificación integral de desarrollo

El marco de referencia propuesto, demanda de una planificación por cada software que sea elaborado. En este caso, cada una de las fases descritas debe ser definida en base a una escala temporal.

Para ello, se debe establecer una planificación guía que ayude su control de la ejecución de las fases, debiendo estas realizarse en base a una programación Gantt. A continuación se expone el modelo de planificación que deberá ser desarrollado:

Es importante citar que en el caso de que la evaluación determine situaciones no satisfactorias, el proceso debe regresar al nivel de diseño y repetir el procedimiento.

Capítulo V

5. Conclusiones y Recomendaciones

5.1 Conclusiones

Una vez concluidos los estudios que permitieron proponer un marco de referencia integral, innovador y especializado en la realidad del entorno nacional para que se convierta en una guía de apoyo para mejorar la calidad en el desarrollo de software, se formulan las siguientes conclusiones.

- La industria del software en Ecuador se encuentra en una etapa de desarrollo inicial. El país principalmente es un consumidor de tecnología lo que ha afectado a las empresas en relación a los altos costos existentes en el mercado. Esta situación demanda la necesidad del desarrollo de esta industria para poder disponer de una oferta nacional que sea especializada en función de los diferentes requerimientos de los clientes. Se puede concluir en base a lo expuesto que existe una oportunidad de mercado, tomando en cuenta que la propia búsqueda de una nueva matriz productiva en el país, exige la disponibilidad y acceso a aplicaciones y productos tecnológicos de calidad que permitan apoyar la gestión de los diferentes ámbitos de la sociedad.
- La calidad en el software comprende varios aspectos que van desde el estudio de la necesidad del cliente, los procesos técnicos de desarrollo, la implementación y el uso. Su cumplimiento demanda de estándares establecidos que permitan disponer de una efectiva orientación para que su uso provea de beneficios que pueden ser evaluados desde diversas perspectivas según el tipo de usuario. De esta manera, aspectos como la reducción de tiempo, costos, fallas, entre otros son elementos que pueden ser alcanzados cuando el software ha sido elaborado orientado a satisfacer una necesidad, cumpliendo objetivos propuestos que impulsaron su creación.

- Los principales marcos de referencias basados en las normas ISO y CMMI se han concentrado en procesos estableciendo normativas enfocadas principalmente en el aspecto técnico de desarrollo de software. Es decir, se enfocan en niveles de gestión que permiten establecer mecanismos estandarizados que orientan su desarrollo. Ambos se focalizan en el mejoramiento de los procesos, diferenciándose principalmente en la forma de gestión. La norma ISO promueve la aplicación de normas para estandarizar el desarrollo de software mientras que el CMMI establece un modelo clasificado en niveles de madurez cuyo alcance permite el perfeccionamiento de los productos desarrollados.
- Los estudios realizados revelaron una falencia principal con respecto a la disponibilidad de contar con un marco de referencia para el desarrollo de software. Esta es la carencia de un marco integral que comprenda todos los procesos necesarios para elaborar un software. Estos elementos fueron asumidos en la propuesta, para poder cubrir estas falencias y presentar una propuesta que sea útil, viable de ser utilizada y principalmente orientadora para disponer de productos de calidad.
- Se ha incluido en el marco referencial procesos técnicos apoyados con técnicas matemático-estadísticas acompañados por indicadores de gestión para evaluar el impacto que el software desarrollado va a tener en el usuario. En este caso, dependiendo del software se establecerán indicadores como costo, tiempo, errores, entre otros debiendo estos ser comparados entre los datos iniciales del proceso sin software con los datos una vez puesta en marcha el producto creado.

5.2 Recomendaciones

En base a las conclusiones formuladas, se recomienda lo siguiente:

- Se recomienda la creación de una asociación para el desarrollo del software especializada en la propuesta de productos especializados en los diferentes sectores de la sociedad, la cual sea conformada por profesionales, empresas privadas, universidades y escuelas politécnicas. Su desarrollo tendrá dentro de sus competencias la certificación de los productos elaborados nacionalmente para mejorar la calidad en su desarrollo y el respaldo integral para el usuario.
- Se recomienda que las universidades a través de sus programas de vinculación con la colectividad desarrollen procesos de capacitación dirigidos a empresas y profesionales referentes a la evaluación de la calidad de software orientando para el adecuado aprovechamiento de la tecnología como medio para fomentar el crecimiento sostenible y sustentable.
- Se recomienda que se formen profesionales especializados en auditoria de software para que estos contribuyan a perfeccionar los marcos de referencia existentes a fin de obtener productos de alta calidad y apoyo para los diferentes ámbitos de la sociedad. En este caso, es importante que las instituciones de educación desarrollen adecuadas mallas curriculares que permitan a los futuros profesionales disponer de competencias referentes a esta importante área de gestión.
- Se recomienda que el marco de referencia propuesto sea analizado y discutido por profesionales en sistemas y procesos a fin de que se contribuya a su perfeccionamiento y posterior difusión. La disponibilidad de conceptos permitirá que su desarrollo cuente con un mejoramiento continuo aportando al desarrollo de software en el país.

- Es importante que los procesos de evaluación de software cuenten con herramientas que faciliten su medición. En este caso, es viable la construcción de un software especializado en control y auditoría que puede aportar a identificar el rendimiento provocado por la automatización de los procesos.

GLOSARIO Y LISTA DE ABREVIATURAS

Auditoría de Sistemas de Gestión

La auditoría de los Sistemas de gestión es una auditoría no financiera cuyo propósito es el de evaluar tanto el desempeño como la eficiencia/eficacia del mismo de una organización o de parte de ella. Esta evaluación es considerada como un análisis crítico constructivo. (Vilar, 2010, p. 11)

Back end

Estructura de datos definida para una aplicación informática conformada por un concepto relacional que permita el adecuado almacenamiento de datos. (Advanced Programming, 2012)

CMMI

Integración de modelos de madurez de capacidades o *Capability maturity model integration* (CMMI) es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software. (CMMI, 2014)

Correlación

Proceso de verificación del nivel de relación existente en el comportamiento de dos variables aleatorias. (Paz, 2012, p. 102)

Desviación

Evaluación de la variabilidad de una variable en un tiempo determinado. Su utilidad permite identificar su comportamiento en base de condiciones establecidas en un entorno. (Paz, 2012, p. 81)

Evaluación

Es la revisión independiente de alguna o algunas actividades, funciones específicas, resultados u operaciones de una entidad administrativa, realizada por un profesional de la auditoría, con el propósito de evaluar su correcta realización y, con base en ese análisis, poder emitir una opinión autorizada sobre la razonabilidad de sus resultados y el cumplimiento de sus operaciones. (Muñoz C. , 2002, p. 11)

Front end

Aplicación informática enfocada en la funcionabilidad y uso del usuario de un sistema informático. (Advanced Programming, 2012)

ISO

International Organization for Standardization. Normas Internacionales proporcionan herramientas prácticas para abordar desafíos orientados en la estandarización de procesos para mejorar su funcionabilidad. (ISO, 2014)

MDI

Formulario primario de una aplicación informática para poder acceder a los servicios contenidos en la aplicación desarrollada. (Chin, 2011, p. 78)

Módulo

Aplicación que cumple un objetivo determinado dentro de la estructura de un software. Área de ejecución de una aplicación. (Chin, 2011, p. 109)

Procesos

Proceso que transforma entradas en salidas. El diseño de un proceso parte del conocimiento del fin que se desea conseguir y de la elección y ordenación de las actividades necesarias para ello. (Muñoz A. , 2009, p. 227)

Retroalimentación

Proceso de alimentación continúa de información del desempeño de un proceso para buscar planes de mejora. (Rodríguez J. , 2009, p. 44)

REFERENCIAS

- AESOFT - Asociación Ecuatoriana de Software. (2012 - 2013). Soluciones de Software 2012 - 2013. Recuperado el 5 de Mayo de 2014, de <http://www.matizediciones.com/aesoft2012/>
- Alfonzo, P., & Mariño, S. (15 de Abril de 2013). Los estándares internacionales y su importancia para la industria del software. Recuperado de <http://www.cyta.com.ar/ta1202/v12n2a3.htm>
- Armas Andrade, R., Chamorro Gómez, A., Montes Beobide, M., & Gutierrez de Mesa, J. A. (Enero de 2007). DESDE ISO 9001 HACIA CMMI, PASOS PARA LA MEJORA DE LOS PROCESOS Y MÉTRICAS. Recuperado de http://www.researchgate.net/profile/Jose-Antonio_Gutierrez_de_Mesa/publication/229049854_Desde_ISO_9001_hacia_CMMI_pasos_para_la_mejora_de_los_procesos_y_mtricas/file/d912f50b2907817ae3.pdf
- Avanced Progammig, S. (2012). Estructura de datos. Estados Unidos: APS.
- Burwick, D. (2009). How to implement CMMI. Estados Unidos: BPS.
- Campderrich, B. (2008). Ingeniería de software. Cataluña-España: Editorial UOC.
- Chin, S. (2011). Programación de objetos. Estados Unidos: Prentice Hall.
- CMMI, P. (2014). Recuperado de http://es.wikipedia.org/wiki/Capability_Maturity_Model_Integration
- Cortés, R. (2009). Introducción al análisis de sistemas y la ingeniería de software. Madrid-España: EUNED.
- Enríquez, M. (2009). Modelos matemáticos en los negocios. México-México: Centrum.
- Equipo Vértice, G. d. (2013). Gestión de calidad ISO 9001:2008. Madrid-España: Editorial Vértice.
- Estudio de Mercado del Sector de Software y Hardware en Ecuador. (Septiembre de 2011). Recuperado de AESOFT - Asociación Ecuatoriana de Software:

http://www.revistalideres.ec/tecnologia/Estudio-mercado-software-hardware-Ecuador_LIDFIL20120620_0001.pdf

Heredero, C. (2010). Organización y transformación de los sistemas de información de la empresa. Madrid-España: ESIC.

ISO, N. (2014). Normas ISO. Recuperado el 2014, de <http://www.iso.org/iso/home.html>

López Bustamante, G., & Flores Vera, J. (26 de 05 de 2010). *Sistema de Gestión de Calidad*. Recuperado de Repositorio de ESPOL: [https://www.dspace.espol.edu.ec/bitstream/123456789/10181/1/Sistema de Gestión de la Calidad.ppt](https://www.dspace.espol.edu.ec/bitstream/123456789/10181/1/Sistema%20de%20Gesti3n%20de%20la%20Calidad.ppt)

Masmela, R. (2011). Como implementar Sistemas para la Gestión de proyectos. Bogotá-Colombia: Armada Digital.

Moore, D. (2009). Estadística aplicada básica. Madrid-España: Antoni Bosch.

Muñoz Mata, M. (2006-2007). Análisis de estándares y modelos de referencia de mejores prácticas. Recuperado de http://www.dlsiis.fi.upm.es/docto_Isiis/Trabajos20062007/Munoz.pdf

Muñoz, A. (2009). La gestión de la calidad total en la administración pública. Madrid-España: Díaz de Saltos.

Muñoz, C. (2002). Auditoría de sistemas computacionales. México-México: Editorial Omega.

Nava, H. (2013). Estándares y modelos de calidad aplicados al software. Recuperado el 4 de Octubre de 2014, de <http://navabautista.wikispaces.com/file/view/Wiki.pdf>

Norma Internacional - ISO 9001. (2013). Recuperado de Sistemas de gestión de calidad aplicada al Software: <http://farmacia.unmsm.edu.pe/noticias/2012/documentos/ISO-9001.pdf>

Paz, C. (2012). Estadística. México-México: Editorial Cálculo.

- PROMPERÚ - Sub Dirección de Promoción Comercial – Exportación de Servicios. (2011). Perfil de mercado de software en Ecuador. Recuperado de <http://www.siicex.gob.pe/siicex/resources/sectoresproductivos/36471667rad3D22C.pdf>
- Rodríguez, C. (2009). Estadística. Bogotá-Colombia: Enseñanza numérica.
- Rodríguez, J. (2009). Procesos administrativos. Madrid-España: ESIC.
- Rodríguez, M. (2 de Junio de 2010). Calidad del producto software, de la teoría a la práctica. Recuperado de Alarcos Quality Center: <http://es.slideshare.net/kybeleconsulting/i-jornada-csi-moiss-rodriguez-alarcos-quality-center-calidad-del-producto-software-de-la-teora-a-la-prctica>
- Rojo, A. M., Forcada, B., Herrera, M., & Zapata, S. (1999). Recuperado de Universidad Nacional de la Plata - Propuesta Metodológica para la Medición de la Satisfacción de Clientes de Software: http://sedici.unlp.edu.ar/bitstream/handle/10915/22261/Documento_completo.pdf?sequence=1
- Salavert, I. (2008). Ingeniería de Software y bases de datos. España: Universidad de Castilla.
- Salazar, D., Villavicencio, M., Macías, M., & Snoeck, M. (2003 - 2004). Estudio estadístico exploratorio de las empresas desarrolladoras de software asentadas en Guayaquil, Quito y Cuenca. Recuperado de <https://www.fiec.espol.edu.ec/resources/investigacion/articulo90.pdf>
- Sánchez, S. (2012). Criterios para la Adaptabilidad de Estándares y Modelos de Procesos de software en PYMES Ecuatorianas. Recuperado de <http://www.slideshare.net/cisoft/criterios-para-la-adaptabilidad-de-estndares-y-modelos-de-procesos-de-software-en-pymes-ecuatorianas>
- Sánchez, S. (Febrero de 2014). Caracterización de la Industria de Software en Ecuador. Recuperado de <http://prezi.com/pxkkbjg8ulm/11bcaracterizacion-de-la-industria-del-software-en-ecuador/>

Santambrosio, R. O. (s.f.). Proyecto Asociativo para Mejora de Procesos ISO 9001 y CMMI. Recuperado de <https://www.inti.gob.ar/software/pdf/ProyectoISO9001.pps>

Scalone, F. (Junio de 2006). ESTUDIO COMPARATIVO DE LOS MODELOS Y ESTANDARES DE CALIDAD DEL SOFTWARE. Recuperado de 2014, de <http://www.iidia.com.ar/rgm/tesistas/scalone-tesis-maestria-ingenieria-en-calidad.pdf>

Solines Chacón, P. (Diciembre de 2010). *Clasificación de las PYMES, de acuerdo a la Normativa implantada por la Comunidad Andina en su Resolución 1260 y la legislación interna vigente*. Recuperado de Consultores de Negocios con una Perspectiva Internacional - Russell Bedford Ecuador S.A.: <http://www.russellbedford.com.ec/images/Boletines%202010/12.%20Resolucion%20SUPER%20CIAS%20PYMES%20-%20SC-INPA-UA-G-10-005.pdf>

Tomás, J. (2012). Estadística. Madrid-España: MAD.

Torres Samaniego, M. Z. (Abril de 2007). ESTUDIO COMPARATIVO ENTRE LOS ESTANDARES ISO/IEC TR 15504 Y CMMI. Recuperado de <http://bibdigital.epn.edu.ec/bitstream/15000/240/1/CD-0637.pdf>

Vilar, F. (2010). La auditoría de los sistemas de gestión de calidad. Estados Unidos: ADS Quality.

Xitumul Ruiz, D. M. (Febrero de 2007). NORMAS ISO 9000 VS. CMMI-SW COMO ESTÁNDAR DE CALIDAD EN EL DESARROLLO DEL SOFTWARE Y EL PROCESO DE OBTENCIÓN DE LA CERTIFICACIÓN EN CADA ESTÁNDAR. Recuperado de http://biblioteca.usac.edu.gt/tesis/08/08_0357_CS.pdf