



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

TRABAJO DE GRADO DE MAESTRÍA

**MODELO DE INTEGRACION DE LOS PROCESOS INVOLUCRADOS EN EL
DESARROLLO DE SOFTWARE EN LAS PEQUEÑAS Y MEDIANAS
EMPRESAS EN VENEZUELA**

Presentado por:

Castellanos Espinoza, Nubia Ligtsolet

Para optar al título de
Magíster en Sistemas de Información

Tutor
Remedios, María Esther

Caracas, Junio de 2018

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

TRABAJO DE GRADO DE MAESTRÍA

**PROPUESTA DE UN MODELO DE INTEGRACION DE LOS PROCESOS
INVOLUCRADOS EN EL DESARROLLO DE SOFTWARE EN LAS PEQUEÑAS
Y MEDIANAS EMPRESAS EN VENEZUELA**

Presentado por:

Castellanos Espinoza, Nubia Ligtsolet

Para optar al título de
Magíster en Sistemas de Información

Tutor
Remedios, María Esther

Caracas, Junio de 2018

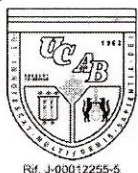
CARTA DE APROBACIÓN DEL TUTOR

Por la presente hago constar que he asesorado y leído el Trabajo de Grado de Maestría, presentado por la ciudadana **Nubia Ligtsolet Castellanos Espinoza**, titular de la cédula de identidad N° 11.944.503 para optar al Título de Magíster en Sistemas de Información, cuyo título es: PROPUESTA DE UN MODELO DE INTEGRACION DE LOS PROCESOS INVOLUCRADOS EN EL DESARROLLO DE SOFTWARE EN LAS PEQUEÑAS Y MEDIANAS EMPRESAS EN VENEZUELA; y que apruebo sea evaluado por el jurado que se designe.

En la ciudad de Caracas, a los 18 días del mes de julio de 2017

María Esther Remedios

CI. 5.530.488



UNIVERSIDAD CATÓLICA ANDRÉS BELLO

Urb. Montalbán - La Vega - Apartado 20332

Telf.: (0212) 407-44-44 Fax: 407-43-49

Estudios de Postgrado

Ref. J-00012255-5

ACTA DE EVALUACIÓN DE PRESENTACIÓN Y DEFENSA DE TRABAJO DE GRADO DE MAESTRÍA MAESTRÍA EN SISTEMAS DE INFORMACIÓN

Nosotros, Profesores MARÍA REMEDIOS DE LEGONIA (tutor), MAYRA NARVAEZ GONZALEZ DE PEREIRA y SAMUEL VILLEGAS ARVELO, designados por el Consejo de Postgrado de la Facultad de Ingeniería a los seis días del mes de junio del año dos mil dieciocho, para conocer y evaluar en nuestra condición de jurado del Trabajo de Grado de Maestría "PROPUESTA DE UN MODELO DE INTEGRACIÓN DE LOS PROCESOS INVOLUCRADOS EN EL DESARROLLO DE SOFTWARE EN LAS PEQUEÑAS Y MEDIANAS EMPRESAS EN VENEZUELA", presentado por la ciudadana Castellanos Espinoza, Nubia Ligtsolet, C.I. N°. 11944503, para optar al grado de Magister en Sistemas de Información.

Declaramos que:

Después de haber estudiado dicho trabajo, presenciamos la exposición del mismo, a los once días del mes de julio del año dos mil dieciocho, en la sede de los Estudios de Postgrado de la Universidad Católica Andrés Bello.

Hechas por nuestra parte, las preguntas y aclaratorias correspondientes y, una vez terminada la exposición y el ciclo de preguntas, hemos considerado formalizar el siguiente veredicto:

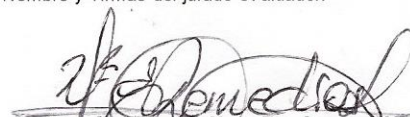
APROBADO

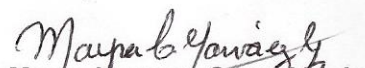
Hemos acordado calificar la presentación y defensa del Trabajo de Grado de Maestría con Dieciocho (18) puntos.

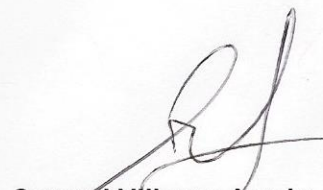
(Observaciones o declaratoria de recomendación)

En fe de lo cual, nosotros los miembros del jurado designado, firmamos la presente acta en Caracas, a los once días del mes de julio del año dos mil dieciocho.

Nombre y firmas del jurado evaluador:


María Remedios De Legonia
C.I.: 5530488


Mayra Narvaez González de
C.I.: 4688671


Samuel Villegas Arvelo
C.I.: 4.580.757



LISTA DE ACRÓNIMOS Y SIGLAS

BPM: Bussines Process Managment, Gestión de Procesos de Negocio.

BPMN: Business Process Model and Notation. Notación BPM.

CMMI: Capability Maturity Model Integration. Integración de modelos de madurez de capacidades.

CONSECOMERCIO: Consejo Nacional del Comercio y los Servicios

ECS: Equipo de Calidad de Software.

EDA: Equipo de Desarrollo de Aplicaciones.

EDT: Estructura de Descomposición del Trabajo

ERF: Equipo de Requerimientos Funcionales.

ESP: Equipo de Seguimiento de Proyectos.

IEEE: Instituto de Ingeniería Eléctrica y Electrónica.

ISO: International Organization for Standardization.

MoProSoft: Es un modelo de procesos para la industria de software en México. Fomenta la estandarización a través de buenas practica en al gestión y desarrollo de software.

PMBOK: Project Management Body of Knowledge. Es un libro en el que se presentan estándares, pautas y normas para la gestión de proyectos.

PMI : El Project Management Institute. Instituto de Gerencia de Proyectos.

PRINCE: Projects in Controlled Environments. Es un método de gestión de proyectos que cubre la gestión, el control y la organización de un proyecto.

PROJECT: software de administración de proyectos y programas de proyectos, diseñado, desarrollado y comercializado por Microsoft

PYMES: Pequeñas y Medianas Empresas.

SQA: Software Quality Assurance (Calidad de Software).

SEI: Software Engineering Institute.

UCS: Unidad de Calidad de Software.

UDA: Unidad de Desarrollo de Aplicaciones.

URF: Unidad de Requerimientos Funcionales.

USP: Unidad de Seguimiento de Proyectos.

WBS: Work Breakdown Structure.



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

PROPUESTA DE UN MODELO DE INTEGRACION DE LOS PROCESOS
INVOLUCRADOS EN EL DESARROLLO DE SOFTWARE EN LAS PEQUEÑAS Y
MEDIANAS EMPRESAS EN VENEZUELA

Autor: CASTELLANOS ESPINOZA, NUBIA LIGTSOLET
Tutora: REMEDIOS, MARÍA ESTHER
Año: 2017

RESUMEN

Este trabajo de investigación se enfoca en la elaboración de un Modelo que permite integrar cuatro de los procesos que intervienen en los Proyectos de Construcción de Software: Proceso de Levantamiento de Requerimientos, Proceso de Construcción de Software, Proceso de Pruebas de Calidad y Proceso de Gestión de Proyectos; aplicable a empresas de que desarrollan internamente sus artefactos de software o fabricas software. Para ello se plantea una investigación de campo, no experimental, para la recolección de datos se utilizaron cuestionarios, entrevistas y observación directa para proceder a construir y describir los mapas de procesos necesarios a fin de obtener una visión sistémica de la situación actual. El uso del concepto de Gestión de Procesos de Negocio y el uso de BPM apoyó la construcción y el modelado de procesos. Con la aplicación del Modelo se obtuvieron resultados satisfactorios, al final del proceso de construcción de los Sistemas o Artefactos de Software, optimizando: la comunicación entre los equipos de trabajo, los tiempos de entrega de los proyectos, la calidad en el producto y en el servicio; creando el precedente de una base de conocimiento para futuros proyectos. Para lograr este objetivo se usó el estándar de calidad ISO 9126.

Palabras Clave: Desarrollo de Software, Mejores Prácticas, Documentación de Procesos De Software.

Línea de Investigación: Ingeniería de Software.

INDICE GENERAL

Carta de Aprobación del Tutor	i
Lista de Acrónimos y Siglas	ii
Resumen	iii
Índice de Figuras	xi
Índice de Tablas	xiii
Introducción	1
Capítulo I	4
El Problema	4
1.1 Planteamiento del Problema	4
1.1.1 Formulación del Problema	11
1.1.2 Sistemización del Problema	11
1.2 Objetivos	11
1.2.1 Objetivo General	11
1.2.2 Objetivos Específicos	11
1.3 Justificación de la Investigación	12
1.4 Alcances y limitaciones de la Investigación	13
Capítulo II	14
Marco Teórico	14
2.1 Antecedentes de la Investigación	15
2.2 Fundamentos Teóricos	21
2.2.1 Definición de Modelo	21
2.2.2 Los Procesos	23
2.2.2.1 Definición de Procesos	23
2.2.2.2 La Gestión Integrada por Procesos	25
2.2.2.3 Mapas de Procesos	26
2.2.2.4 Rediseño de Procesos de Negocio	28
2.2.3 Proceso de Desarrollo de Software	29
2.2.3.1 Definición de Software	30
2.2.3.2 Desarrollo de Software	31
2.2.3.3 Ingeniería de Software	31

2.2.3.4 Sistemas de Información	32
2.2.3.5 Modelos de Desarrollo de Software	33
2.2.4 Proceso de Levantamiento de Requerimientos	36
2.2.4.1 Ingeniería de Requerimientos	37
2.2.4.2 Documento de Especificación de Requerimientos	40
2.2.5 Proceso de Calidad de Software	41
2.2.5.1 Calidad de Software	41
2.2.5.2 Estándar ISO 9126	42
2.2.5.3 ISO/IEC 25000	44
2.2.6 Proceso de Gestión de Proyectos	45
2.2.6.1 Definición	45
2.2.6.2 Proyecto de Software	46
2.2.7 Qué es Una Métrica	47
2.2.7.1 Métricas de Requerimientos de Software	47
2.2.7.2 Métricas de Desarrollo de Software	47
2.2.7.3 Métricas de Calidad de Software	48
2.2.7.4 Métricas de Gestión de Proyectos	49
2.2.8 BPM – Gerencia de Procesos de Negocio	50
2.2.8.1 Definición BPM	50
2.2.8.2 Beneficios de Implementar BPM	51
2.2.8.3 Recomendaciones al Implementar BPM	52
2.2.8.4 BPMS (Business Process Management Suite)	53
2.2.8.5 BPMN -Business Process Management Notation	54
2.3 Marco Referencial	56
2.3.1 Historia de las Pequeñas y Medianas Empresas de Venezuela	56
2.3.2 El Software de Pequeñas y Medianas Empresas de Venezuela	57
2.4. Bases Legales	57

Capítulo III	59
Marco Metodológico	59
3.1 Enfoque de la Investigación	59
3.2 Tipo de Investigación	60
3.3 Diseño de la Investigación	61
3.4 Nivel de la Investigación	62
3.5 Población y Muestra	63
3.6 Técnicas de Recolección de Datos	64
3.7 Técnicas de Análisis de Datos	65
3.8 Fases de la Investigación	67
3.9 Variables, Definición Conceptual y Operacional e Indicadores	68
3.10 Estructura Desagregada de Trabajo	71
3.11 Aspectos Éticos	72
3.12 Cronograma	74
3.13 Recursos	75
Capítulo IV	
Análisis de Datos	76
4.1 Análisis de encuestas y entrevistas	76
4.2 Resultados de la Entrevistas	84
4.2.1 Unidad de Desarrollo de Software	84
4.2.2 Unidad de Calidad de Software	87
4.2.3 Unidad de Requerimientos Funcionales	89
4.2.4 Unidad de Seguimiento de Proyectos	90
4.3 Matriz FODA de Procesos	93
4.4 Acciones a Tomar	95
4.5 Informe Descriptivo de los Procesos	95
4.5.1 UDA: Unidad de Desarrollo de Software	95
4.5.2 UCS: Unidad de Calidad de Software	97
4.5.3 URF: Unidad de Requerimientos Funcionales	98
4.5.4 USP: Unidad de Seguimiento de Proyectos	99

Capítulo V

RESULTADOS	100
5.1 Mapa Conceptual del Modelo Propuesto	100
5.1.1 Mapa de Proceso General del Modelo Propuesto	100
5.1.2 Mapa de Proceso Unidad de Seguimiento de Proyectos	101
5.1.3 Mapa de Proceso Unidad de Requerimientos Funcionales	102
5.1.4 Mapa de Proceso Unidad de Desarrollo de Aplicaciones	103
5.1.5 Mapa de Proceso Unidad de Calidad de Software	104
5.2 Diseño del Modelo Propuesto con BPMN 2.0	105
5.3 Formatos del Modelo Propuestos	113
5.3.1 Solicitud de Registro de Proyectos de TI	113
5.3.2 Registro de Proyectos de TI	114
5.3.3 Plantilla Gantt de Proyectos de TI	115
5.3.4 Minuta de Proyectos de IT	116
5.3.5 Requerimientos de Proyectos de TI	118
5.3.6 Aclaratorias de Dudas de Requerimientos de Proyectos de TI	120
5.3.7 Aprobación de Requerimientos de Proyectos de TI	122
5.3.8 Diseño de Pruebas de Requerimientos de Proyectos de TI	124
5.3.9 Cierre de Caso de Prueba Funcional de Requerimientos	126
5.3.10 Cierre de Caso de Prueba Funcional de Requerimientos	128
5.3.11 Cierre de Proyectos de TI	129
5.4 Diseño de Indicadores de Gestión	130
5.4.1 Indicadores de la Unidad de Seguimiento de Proyectos	130
5.4.2 Indicadores de la Unidad de Requerimientos Funcionales	131
5.4.2 Indicadores de la Unidad de Calidad de Software	132
5.4.3 Indicadores de la Unidad de Desarrollo de Aplicaciones	132
5.5 Validación del Modelo según el estándar ISO 9126	129
5.5.1 Portada del Formato de Evaluación	130
5.5.2 Parámetros de Evaluación	131
5.5.2 Tabulación del Formato de Evaluación	132
5.5.2 Resultados de la Evaluación	136

Capítulo VI	142
CONCLUSIONES Y RECOMENDACIONES	142
CONCLUSIONES	142
RECOMENDACIONES	143
 Referencias Bibliográficas	145
Anexos	148
N° 1 Evolución del PIB de los principales sectores desde 1997-ENE. /SEPT. 2015	149
N° 2 Diagramas de Flujo Relacionado con la Revisión de los Procesos	150
N° 3 Ejemplo de Mapa de Proceso	151
N° 4 Diez Prácticas recomendadas de BPM	152

INDICE DE FIGURAS

Figura	Pág.
N° 1: Proceso para la planificación y control de proyectos de software	9
N° 2. Mapa Mental Fundamentos Teóricos	15
N° 3. Representación General de un Proceso	23
N° 4. Enfoque en los Procesos	24
N° 5 Diagrama de un Mapa de Proceso	27
N° 6. Simbología para Construir Mapas de Procesos	28
N° 7. Riesgos del Cambio Organizacional	29
N° 8. Capas de la ingeniería de software	32
N° 9. Inversión en capital de Tecnología de la Información	33
N° 10. Proceso General de Levantamiento de Requerimientos	40
N° 11. Diagramas de Calidad	43
N° 12. Grupos de Procesos de la Dirección de Proyectos	46
N° 13. Enfoque por Departamentos vs. BPM	51
N° 14. 7 pasos para el éxito de BPMS	53
N° 15. Ciclo Metodológico de la Investigación Proyectiva	68
N° 16. EDT Modelo Integración	72
N° 17. Diagrama Gantt	74
N° 18. Mapa de Proceso General del Modelo Propuesto	100
N° 19. Mapa de Proceso Unidad de Seguimiento de Proyectos	101
N° 20. Mapa de Proceso Unidad de Requerimientos Funcionales	102
N° 21 Mapa de Proceso Unidad de Desarrollo de Aplicaciones	103
N° 22 Mapa de Proceso Unidad de Calidad de Software	104
N° 23 Solicitar Registro de Proyectos de TI	105
N° 24 Registrar Proyectos de TI	106
N° 25 Notificación de Inicio de Proyecto de TI	107
N° 26 Aprobación de Requerimientos	108
N° 27 Aprobación Funcional del Requerimiento	108
N° 28 Actualizar Plantilla Gantt	109

N° 29 Construcción de Componentes de Software	110
N° 30 Culminación de Pruebas Funcionales	111
N° 31 Firma de Acta de Certificación	111
N° 32 Aceptación de Pase a Producción	112
N° 33 Cierre de Proyecto	112

INDICE DE TABLAS

Tabla	Pág.
N° 1. Modelos y Estándares Internacionales	10
N° 2. Tipos de Modelo	22
N° 3. Principales diferencias entre Gestión por Procesos y Gestión Integrada por Procesos	26
N° 4. Elementos del Proceso de Desarrollo de Software	30
N° 5. Modelos Evolutivos para Desarrollo de Software	34
N° 6. Principios para Levantamiento de Requerimientos	38
N° 7. Resumen Estándar ISO 9126	44
N° 8. Métricas de Desarrollo de Software	48
N° 9. Métricas de Calidad de Software	49
N° 10. Métricas de Gestión de Proyecto de Software	49
N° 11. Categorías básicas de elementos BPMN	55
N° 12. Procesos Resultantes del Requerimiento a Aprobado	110

INTRODUCCIÓN

Actualmente en Venezuela, existen un importante número de Pequeñas y Medianas Empresas (PYMES), que poseen su propia estructura para desarrollar proyectos Informáticos, especialmente los relacionados con el desarrollo de software y aplicaciones; esto con la finalidad de adaptarse rápidamente a las exigencias del mercado, donde cada vez los clientes tanto internos como externos demandan más productos y servicios. Los clientes internos (directores, gerentes, jefes de las diversas unidades de negocio) por un lado, demandan desarrollo de nuevos módulos, o mantenimientos de módulos ya existentes de los sistemas hechos en casa o de aquellos de los que poseen licencia para hacer modificaciones. Los clientes externos (consumidores de los productos o servicios de las PYMES) se les ofrece una línea de funcionalidades “ON LINE” que le permiten agilizar la cotidianidad y garantizar en términos generales un mejor servicio; tales como: cotizar pólizas, reportar facturas de pagos, consultar estados de cuentas, listar precios de productos o servicios, efectuar reclamos de siniestros, entre otros. Desde cualquier perspectiva del cliente en las PYMES siempre existe la necesidad de Desarrollar Software.

La problemática existente en la mayoría de los casos, radica en que el desarrollo de los productos de software desarrollados suelen no cumplir con los tiempos preestablecidos, ya que durante el proceso de desarrollo suelen presentarse innumerables situaciones que retrasan la entrega, o en el mejor escenario, si se entregan en el tiempo estimado, el esfuerzo que debe hacerse para cumplir, es mucho mayor al planificado inicialmente. No se sabe específicamente cuales son las causas, puede ser a la falta de comunicación efectiva entre los equipos de trabajo, la falta de uso de metodologías y estándares, falta de documentación de procesos, o a la escasa planificación.

Para llevar a cabo el estudio, en la presente investigación, inicialmente se diagnostica el problema, se diseña el Modelo Integrado de Procesos, el cual generó un aporte significativo a la Ingeniería de Software, específicamente en las Pequeñas

y Medianas Empresas que poseen una área de Desarrollo de Software dentro de su estructura organizativa, con la finalidad de mejorar los tiempos de respuestas, la calidad en el proceso y la calidad en el producto. La integración es necesaria, por lo que se hace imperativo buscar un mecanismo para que se dé, en función a esta premisa se desarrolló el objetivo principal de este trabajo de grado. El beneficio de integrar los procesos se obtiene al permitir que todos los entes participantes en un Proyecto de Desarrollo de Software compartan la misma información acerca del proyecto, unificando así la información para todos los equipos de trabajo, otro de los beneficios son: generación de estadísticas reales de la cantidad de requerimientos, cantidad de defectos, horas hombres invertidas, entre otras estadísticas que permitan evaluar y gestionar más eficientemente proyectos futuros, entre muchos otros beneficios. Adicionalmente, se pretende fomentar una cultura de trabajo colaborativo, documentación de procesos y uso de estándares que permitan la prestación de servicios de calidad con el uso del Modelo Integrado de Procesos.

Para resolver la problemática expuesta anteriormente, y a fin de lograr los objetivos, esta investigación se estructuró de la siguiente manera: El Problema, Marco Teórico y Marco Metodológico, que a continuación se describen:

En el Capítulo I - “El Problema” se hace una descripción detallada del planteamiento del problema, los Objetivo que se pretenden alcanzar en el desarrollo de la investigación, se explica por qué es importante llevarla a acabo y las limitaciones presentes en la investigación.

El Capítulo II – “Marco Teórico”, inicia con los antecedentes de la investigación, mostrándose algunos estudios que sirvieron de referencia para ubicar aspectos relacionados al tema, se encontraron muchos antecedentes de investigación en el área de calidad de procesos y productos de software. Luego se define el marco teórico el cual soporta las bases de la investigación, evolucionando desde la definición de conceptos básicos, hasta la definición aspectos más técnicos. Así como también se presenta el marco referencial y las bases legales que sustentan la investigación.

En el Capítulo III - Marco Metodológico, se describe la manera en la que será abordada la investigación. Se eligió un enfoque cuantitativo, un tipo de investigación de campo con un diseño No Experimental porque se observan los hechos tal y como se presentan en su contexto real, se determina que nivel de la investigación es Proyectivo porque conducen a una creación basada en conocimientos anteriores. Así como también se determinaron las variables y los indicadores bajo las cuales serán medidas. Las fases diseñadas para esta de la investigación son las siguientes: fase exploratoria, fase diagnóstica, fase analítica, fase descriptiva, fase de diseño, fase de construcción, fase evaluativa, fase documental.

Por consideraciones éticas, se mantiene el nombre de las empresas objeto del estudio en anonimato, esperando que el proceso de recolección de información sea más fluido y sea positivo para la investigación.

Finalmente se diseñó un cronograma de actividades con una carta Gantt, luego presentan las referencias bibliográficas, que sirvieron de soporte documental para construir la investigación y los anexos que la complementan.

Se aplicó el modelo en dos empresas, una del sector Retail y otra del sector Droguerías, El resultado de la aplicación de modelo, fue evaluado por un panel experto constituido por tres personas desarrolladoras de software con más de diez años de experiencia en el área de desarrollo de proyectos informáticos, quienes certificaron el resultado satisfactoriamente en los formatos de evaluación del modelo.

CAPITULO I: EL PROBLEMA

En este capítulo se presenta el planteamiento y la formulación del problema que se aprecian en las PYMES al momento de desarrollar artefactos de software para los diversos Sistemas de Información, en su propia fábrica de software. También se define el objetivo general, los objetivos específicos, la justificación de la investigación, limitaciones y alcance de la presente investigación.

1.1 Planteamiento del Problema

Actualmente en Venezuela, muchas de las PYMES poseen algún tipo de Sistema de Información Administrativo-Gerencial que permite llevar a cabo las operaciones el CORE del negocio; por lo general dicho sistema es mantenido o administrado por la Unidad de Sistemas o Unidad de Informática de la Organización, en algunos casos son propietarios del código fuente de la aplicación, del modelo de datos y los componentes que la conforman. El presente trabajo de grado está enfocado al estudio de este grupo de PYMES, donde La Unidad de Sistemas cumple funciones de fábrica de software (Software Factory) interno de cada PYME.

Es frecuente encontrar en una PYME que el sistema CORE del negocio lleva implantado muchos años en la organización, y son muchos los analistas que han hecho modificaciones a la lógica del negocio y en el transcurso del tiempo muy pocos se han encargado de documentar los cambios en el código fuente del sistema, así como tampoco han documentado los procesos desarrollados.

A pesar de que cada PYME tiene una estructura de trabajo diferente, lo que ocurre cuando se solicita el desarrollo de una nueva funcionalidad de Software en el Sistema CORE, el cual puede consistir en adicionar nuevas funcionalidades a un desarrollo ya existente, o en desarrollar una nueva interfaz de usuario, un proceso, procedimiento de base de datos o reportes; el proceso que se sigue para cumplir tal fin es el siguiente: se activa el **Equipo de Requerimientos Funcionales** (ERF), quienes se encargan de documentar todas las funcionalidades que el usuario desea

cubra el nuevo proyecto. Para redactar el documento de funcionalidades se coordinan una serie de reuniones, inicialmente con la unidad dueña del proceso quien describe verbalmente todo lo que considera necesario para cumplir con lo solicitado; poco a poco se incorpora en estas reuniones otras áreas como el **Equipo de Desarrollo de Aplicaciones** (EDA), quien hace sus aportes para determinar la factibilidad técnica del mismo. El producto final del ciclo de reuniones, es un documento el cual suele titularse “Diseño Propuesto”, que contiene la descripción de lo solicitado por el usuario; existe la figura del **Equipo de Seguimiento de Proyectos** (ESP) quienes se encargan de construir el PROJECT, y en conjunto con el EDA se asigna una fecha de inicio del proyecto en función de la disponibilidad de recursos.

Cuando la persona que cumple el rol de Desarrollador de Software inicia sus actividades, es altamente probable que le surjan muchas dudas, principalmente porque no conoce cómo funciona el negocio, o porque no participa en las reuniones de levantamiento de información. En ese momento debe interrumpir su jornada de desarrollo para indagar el cómo debe hacer para cumplir con la premisa planteada en el documento, se presenta entonces el primer problema que se traduce en pérdida de tiempo del EDA y retraso del plan inicial por falta de especificación funcional/técnica del requerimiento. En ocasiones, es necesario hacer una reunión con los involucrados para aclarar el proceso, no siempre se hace una minuta, o si se hace, es luego de varios días de efectuada la reunión; puede ocurrir también que resultado de esa reunión inicial, se convoque a una nueva reunión con otros integrantes, como también puede ocurrir que se plantee un cambio de alcance, y a su vez la elaboración de un documento de “Diseño Propuesto” complementario.

Una vez que el Desarrollador de Software aclara todas sus dudas, puede continuar con el desarrollo y efectuar las pruebas unitarias; una vez da por culminado todo el desarrollo del proyecto, se entrega al **Equipo de Calidad de Software** (ECS) el ambiente de trabajo listo para que inicien las pruebas de certificación, es importante destacar que no todas las PYMES cuentan con este equipo de trabajo. Cuando en

las pruebas de certificación se detecta que el desarrollo no cumple con algunas de las “premisas” del “Diseño Propuesto”, conlleva a un retrabajo por parte del EDA; aquí se detecta el segundo problema, no se aplica calidad al proceso de desarrollo, sino que se espera hasta el final para determinar si existe alguna falla. Otro de los grandes problemas que se detecta es que no existe documentación de los procesos, y es por ello que todos los equipos de trabajo (EDA, ECS, ERF y ESP), no tienen una visión macro de cómo debe ser el flujo del proceso del nuevo proyecto.

Dentro de los principales Problemas detectados se listan los siguientes:

- ✓ No existe documentación de los procesos, lo cual hace que el tiempo invertido por parte de los Desarrolladores de Software en análisis sea mayor; también puede ocurrir que cada quien le dé un sentido diferente a la lectura de las premisas escritas en el documento de funcionalidades, por no existir un proceso claramente documentado.
- ✓ Las pruebas de Calidad de Software se aplican al finalizar el desarrollo de todos los artefactos involucrados en el proyecto, esto es particularmente desventajoso ya que cualquier error que presente un artefacto de software puede incidir directamente en todos los demás, lo cual podría comprometer significativamente los tiempos de entrega.
- ✓ El concepto de Levantamiento de Requerimientos es pobre, a pesar de que se elabora un documento con las premisas y la narrativa de lo que el cliente desea obtener al final del desarrollo, no existe un documento de especificaciones técnicas, se propone desarrollar el contenido del documento actual en tres documentos: diseño conceptual, diseño técnico, diseño de pruebas unitarias.
- ✓ Se elaboran las minutas de Reunión a destiempo: cuando se efectúa una reunión de seguimiento, bien sea para aclarar algún punto que no quedó claro en el documento de especificaciones, o porque hubo algún cambio de alcance, o por cualquier otra situación presentada en el proyecto; no se hace una minuta o borrador de la minuta con los acuerdos, no son leídos los acuerdos al concluir la reunión. Según FRANCO (2004), en su artículo

CONDUCCIÓN DE REUNIONES PARA LA OPERACIÓN EFICAZ DE LOS EQUIPOS DE TRABAJO sugiere “Diseñe un método de registro de acciones que se concentre en el contenido de la reunión, en lugar de los procedimientos de la misma. Es importante que las actas reflejen lo que la reunión produjo en vez de lo que se trató en ella.”

- ✓ No existe un repositorio con la bitácora del proyecto, donde todos los involucrados puedan acceder y tener una visión general o detallada de todo lo que ocurre en el proyecto, en este sentido se sugiere centralizar la información para que todos hablen un mismo lenguaje, y que una vez culminado el proyecto sirva de marco de referencia para proyectos futuros.
- ✓ Cuando se modifican artefactos de software existentes, suelen salir errores que no tienen relación directa con el desarrollo actual, lo cual requiere investigar la causa, se traduce en inversión de tiempo adicional al planificado.
- ✓ Si hay un nuevo alcance, se elabora un documento Anexo, y se pueden crear tantos documentos como cambios hayan, lo cual constituye una problemática para el versionamiento de los documentos, puede ocurrir que a algún integrante no le llegue la última actualización del documento.
- ✓ Las pruebas unitarias son deficientes, probablemente las pruebas que hace el desarrollador, no contempla todas las pruebas que aplica el equipo de calidad al producto final.

En resumen, el problema que se detecta es la una mejor orquestación entre los equipos mencionados anteriormente, para que el producto solicitado por los clientes pueda:

- ✓ Culminarse en el tiempo establecido.
- ✓ Cumplir con los requisitos de calidad en el proceso de desarrollo del proyecto.
- ✓ Cumplir con los requisitos de calidad en la entrega del producto final.

Según Marín-Sánchez (2014). Proceso para la planificación y control de proyectos de software utilizando Xedro-GESPRO. Menciona que los principales problemas del

desarrollo de software, vienen dados por insuficiencias durante los procesos de planificación y control de los proyectos, entre los que resaltan:

- ✓ Escasa definición de los objetivos del proyecto, su prioridad, estudio de factibilidad, estructura de desglose de trabajo y contratos, lo cual dificulta su gestión y alcance.
- ✓ Las planificaciones se realizan basadas en la experiencia, utilizando hojas “Excel” o “Word”, o herramientas de gestión de proyectos con tecnologías propietarias como *Microsoft Project* (Microsoft®) o *Primavera* (Oracle®), incurriéndose en gastos adicionales por concepto de pagos de licencias a terceros.
- ✓ Los proyectos se controlan en base a preguntas que los jefes realizan a sus subordinados y no en función de indicadores objetivos obtenidos a través de fuentes de información confiables.
- ✓ No se cuenta con registros históricos que permitan crear métodos propios para planificar y/o tomar decisiones.
- ✓ Gran volumen de información sobre modelos y estándares para la mejora de procesos (ISO, CMMI, PMBOK), pero escaso conocimiento en las organizaciones sobre cómo aplicarlos de forma práctica.

Según, Marín-Sánchez (2016) en su artículo Control de proyectos de software: actualidad y retos para la industria cubana, concluye que para motivar la mejora, las lecciones aprendidas se deben documentar a lo largo del ciclo de vida del proyecto, o como mínimo, durante el cierre del mismo. Estas deben generar información para mejorar la gestión y toma de decisiones de futuros proyectos, quedando archivadas en una base de conocimiento de la organización. Una base de conocimiento almacena información histórica y lecciones aprendidas sobre los resultados de las decisiones y el desempeño de los proyectos. Con el fin de agilizar los procesos de intercambio de información y control, las bases de conocimiento deben contener funcionalidades necesarias para la recolección, organización y recuperación de la experiencia acumulada mediante el uso de tecnologías. El proceso planteado por el

autor cuenta con actividades, entradas, salidas, roles y sus responsabilidades para cada actividad (ver figura 1).

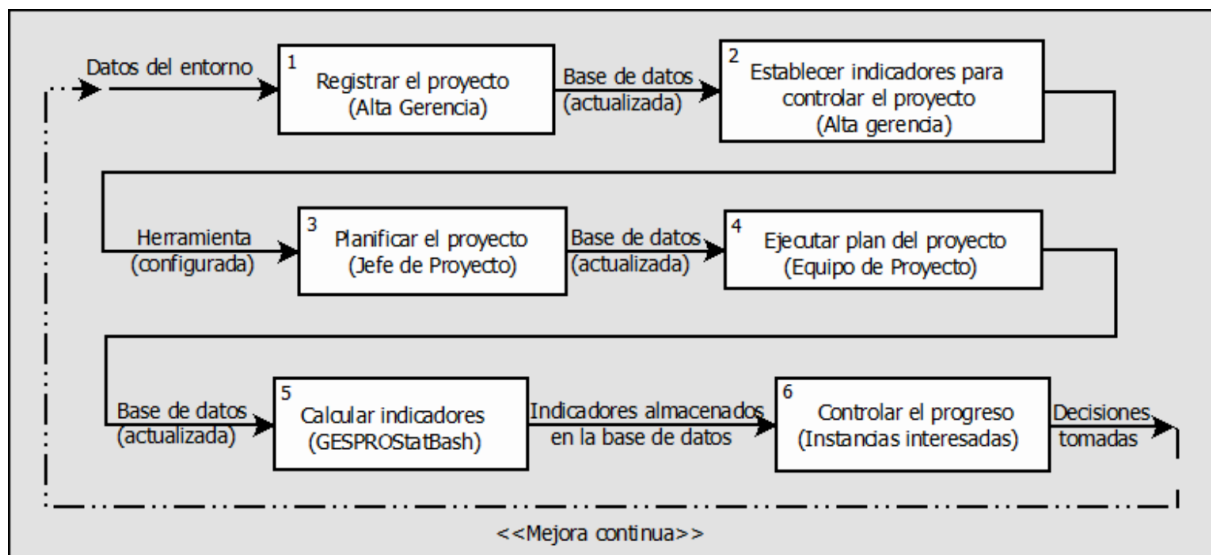


Figura N° 1 Proceso para la planificación y control de proyectos de software.
Fuente: Marín-Sánchez (2014)

Según, Marín-Sánchez, (2014), en su artículo Proceso para la planificación y control de proyectos de software utilizando Xedro-GESPRO, menciona que, si bien PMBOK y PRINCE proporcionan buenas prácticas para la gestión de proyectos de manera general, no especifican acciones concretas a tener en cuenta en el desarrollo de proyectos de software. Por su parte, pese a la gran ventaja que CMMI y las ISO proporcionan en condición de valor agregado a las entidades, su aplicación resulta costosa en términos económicos y de esfuerzo, requiriendo gran inversión de capital, tiempo y recursos y el retorno de la inversión se produce a largo plazo. Se puede apreciar además, que solo PRINCE y *MoProSoft* proponen roles concretos, PMBOK es el único que describe indicadores para controlar el proyecto y ningún modelo o estándar de los analizados proponen y/o especifican herramientas informáticas como apoyo al proceso de planificación y control de proyectos.

En la tabla 1 se puede observar que existen varios estándares internacionales que bien pudiesen adoptar a las pequeñas y medianas empresas para mejorar los procesos de Desarrollo de Software, sin embargo, no existe en muchas empresas la

madurez para implantar estos estándares, es por ello que surge esta investigación cuyo propósito es diseñar un modelo que sustentado en estos estándares que contribuya con una mejor Gestión de los proyectos de Desarrollo de Software.

Tabla N° 1. Comparación entre Modelos y Estándares Internacionales							
Criterio	ISO 15504	ISO 12207	CMMI-DEV	PMBOK	PRINCE2	MoProSoft	MPS
Dirigido al software	Sí	Sí	Sí	General	General	Sí	Sí
Referencia a otros estándares	---	---	---	---	PRINCE	ISO 9001:2000, CMM v1.1, ISO/IEC 15504-2:1998.	ISO/IEC 12207:2008, ISO/IEC 15504 y CMMI-DEV.
Estructura	Delega en ISO 12207, por mayor aplicabilidad.	Estructurado en 7 grupos de procesos. Se apoya en otras normas ISO.	Estructurado en 22 áreas de procesos y 5 niveles de madurez.	Estructurado en 5 grupos de procesos y 10 áreas de conocimiento.	Estructurado en componentes, procesos y técnicas.	Estructurado en 3 categorías y 6 procesos. Estructura propia.	Estructurado por 7 niveles de madurez y 17 procesos.
Establece roles	Propone roles para la evaluación.	Propone roles genéricos.	Propone ejemplos.	Enfocado en el jefe de proyecto.	Sí	Sí	No
Describe indicadores para el control	No	No	No	Sí	No	No	No
Propone técnicas	No	No	Sí	Sí	Sí	No	Sí
Propone herramientas informáticas	No	No	No	Propone pero no específica.	No	No	No
Costo del estándar	Sí	Sí	No	Sí	Sí	Sí	Sí

Fuente: Marín-Sánchez (2014)

1.1.1 Formulación del Problema

¿Cómo debe estar conformado un modelo que integre de los Procesos involucrados en el Desarrollo de Software en las Pequeñas y Medianas Empresas de Venezuela, de forma tal que cumplan criterios de calidad durante el desarrollo del producto y calidad en el producto final?

1.1.2 Sistemización del Problema

¿Cómo se llevan a cabo los procesos de Desarrollo de Software en las Pequeñas y Medianas Empresas de Venezuela?

¿Cuáles serían los procesos involucrados en el Desarrollo de Software en las Pequeñas y Medianas Empresas de Venezuela?

¿Bajo qué criterios debe llevarse a cabo el proceso de integración de los equipos de trabajo que participan en los proyectos?

¿Cómo son validados los procesos que intervienen en el Desarrollo de Software en las Pequeñas y Medianas Empresas de Venezuela?

1.2. Objetivos

Diseñar un Modelo de Integración de los Procesos involucrados en el Desarrollo de Software de las Pequeñas y Medianas Empresas de Venezuela.

1.2.1 Objetivo General

Diseñar un Modelo Conceptual de Integración de Procesos de Desarrollo de Sistemas de Información en las Pequeñas y Medianas Empresas de Venezuela.

1.2.2 Objetivos Específicos

- ✓ Diagnosticar la situación actual, acerca de cómo se llevan a cabo los procesos de Desarrollo de Software en las Pequeñas y Medianas Empresas de Venezuela.

- ✓ Describir los procesos y los equipos de trabajo involucrados en el Desarrollo de Software en las Pequeñas y Medianas Empresas de Venezuela.
- ✓ Elaborar un Modelo que integre los Procesos que participan en el Desarrollo de Software en las Pequeñas y Medianas Empresas de Venezuela.
- ✓ Validar el modelo propuesto a través de juicio experto.

1.3. Justificación de la Investigación

Hoy día, con el fenómeno de la globalización, es casi imposible que las personas trabajen de manera aislada, la tendencia es que los equipos de trabajos compartan más información, y que toda esta información se encuentre centralizada en algún sistema, de manera que los gerentes y personal directivo de las organizaciones pueda hacer uso útil de ella, por ejemplo, para planificación estratégica o para toma de decisiones. Es por ello que resulta relevante pensar en integrar los procesos de los equipos que participan en los Proyectos de Desarrollo de Software de las Pequeñas y Medianas Empresas de Venezuela, con la finalidad de centralizar la información y compartirla entre los diversos equipos de trabajo. Se justifica entonces mejorar la comunicación entre los equipos de trabajo, mejorar la calidad del proceso de desarrollo de software, y, mejorar el producto antes de ser entregado al equipo de calidad de software. Si se logran estos preceptos, como consecuencia directa se beneficiarán todos los involucrados, los diferentes equipos que participan en el proyecto, inclusive el usuario solicitante, el dueño del proceso. Estos beneficios finalmente se traducirán en la contraprestación de un servicio de calidad a los usuarios finales de los sistemas.

Según Marín-Sánchez (2016), en su artículo “Control de proyectos de software: actualidad y retos para la industria cubana” concluye que para motivar la mejora, las lecciones aprendidas se deben documentar a lo largo del ciclo de vida del proyecto, o como mínimo, durante el cierre del mismo. Estas deben generar información para mejorar la gestión y toma de decisiones de futuros proyectos, quedando archivadas en una base de conocimiento de la organización. Una base de conocimiento almacena información histórica y lecciones aprendidas sobre los resultados de las decisiones y el desempeño de los proyectos. Con el fin de agilizar los procesos de

intercambio de información y control, las bases de conocimiento deben contener funcionalidades necesarias para la recolección, organización y recuperación de la experiencia acumulada mediante el uso de tecnologías.

Esta premisa resume el beneficio que se obtiene de integrar la información generada por los proyectos de desarrollo de software, que se pretende proponer en el desarrollo de esta investigación.

1.4. Alcances y limitaciones de la Investigación

Esta investigación está orientada a estudiar el mecanismo de cómo se pueden integrar los proceso de desarrollo de software en las Pequeñas y Medianas Empresas de Venezuela que posean un área de Desarrollo de Software propio. Se profundizó en el análisis de los procesos particulares de cada uno de los equipos de trabajos que participan en los Proyectos de Desarrollo de Software: Equipo de Desarrollo de Aplicaciones, Equipo de Calidad de Software, Equipo de Requerimientos Funcionales, Equipo de Seguimiento de Proyectos; para determinar la mejor manera de integrarlos en las áreas que los involucren en el Proyecto de Desarrollo de Software. Esta investigación arrojó los siguientes resultados:

- ✓ Diagnóstico del proceso Actual.
- ✓ Modelado de procesos de cada una de las unidades involucradas en el proceso de Desarrollo de Software.
- ✓ Diseño Conceptual del Modelo propuesto
- ✓ Modelado del diseño.
- ✓ Encuesta para validación del modelo propuesto.

Las limitaciones de esta investigación vienen principalmente por:

- ✓ Al no poseer una muestra significativa de las Pymes desarrolladora de Software en Venezuela, sólo se analizaron los procesos de dos empresas una del sector Droguerías y otra del sector Retail.
- ✓ No poder revelar la información de las empresas donde se lleve a cabo la validación del modelo.

CAPÍTULO II: MARCO TEÓRICO

En el presente capítulo se abordan dos aspectos de la investigación, los antecedentes de la investigación y los fundamentos teóricos.

Los antecedentes de la investigación, la cual está enfocada en diseñar un modelo de integración de varios procesos que intervienen en el ciclo de vida de desarrollo de un sistema, desde su concepción hasta su finalización. Se estudian los antecedentes relacionados con mejoras de procesos en general, mejoras de procesos particulares del área de desarrollo de Sistemas de Información, los cuales pueden ser denominados también Desarrollo de Software. Se eligió una Tesis Doctoral, cinco trabajos de Grado de Maestría y un artículo arbitrado, enfocadas a la investigación de mejora de procesos, aplicación de estándares de calidad para desarrollo de software, mejores prácticas como Normas ISO 9126, ISO 25000, BPM; con la intención de conocer como otros autores han desarrollado el tema relacionado con los procesos que involucran de Desarrollo de Software.

Los fundamentos teóricos están orientados a definir y describir cada uno de los aspectos mencionados en los antecedentes, utilizando un método inductivo, el cual presenta una visión de lo particular a lo general, para ello se diseñó un mapa conceptual a fin de facilitar la investigación teórica. En esta sección se definen los concepto básicos de: modelo, proceso, desarrollo de software, ingeniería de software, ingeniería de requerimientos, estándares de calidad de software, gestión de proyectos de software, métricas y gerencia de manejo de procesos; así como también se amplían los conocimientos en cada uno de estos aspectos.

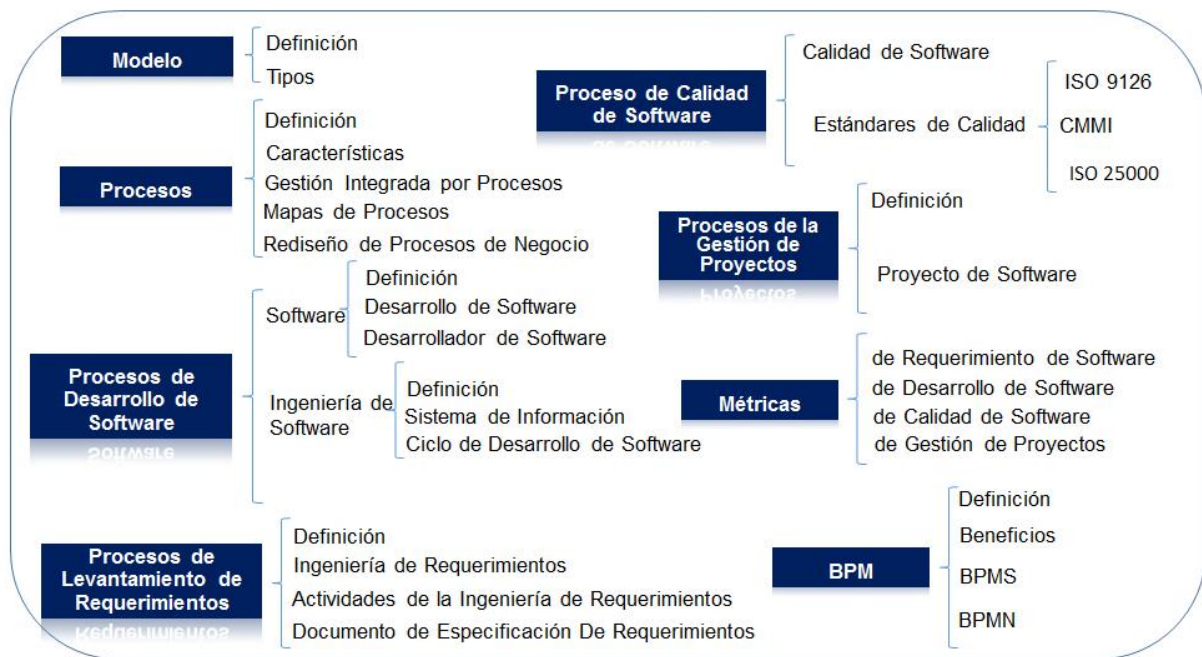


Figura N° 2. Mapa Mental Fundamentos Teóricos

2.1 Antecedentes de la Investigación

Tesis Doctoral

Sanz (2012). Marco Metodológico para la mejora de las actividades de Verificación y Validación de Productos Software. El propósito de esta tesis doctoral es definir un marco metodológico para la mejora del proceso de pruebas software, que integre el conjunto de procesos requeridos para mejorar la eficiencia del desarrollo de las actividades de verificación y validación de productos software; y que a su vez permita a las organizaciones afrontar iniciativas de mejoras del proceso con la finalidad de reducir el impacto de los fallas. Dentro de las principales aportaciones realizadas por esta tesis doctoral tenemos: ha definido un modelo de procesos, se han identificado los principales roles que participan en un entorno de verificación y validación de software, se ha definido una solución flexible, escalable, y adaptable a las necesidades de cada organización.

Aporte: Uno de los procesos involucrados en el desarrollo de sistemas de información, es el proceso de prueba de software, razón por la cual esta tesis es de

vital relevancia para evidenciar como es la estructura y el proceso de validación y pruebas de software y sus propuestas de mejora en el proceso de desarrollo de sistemas de información.

Palabras clave: Pruebas Software – Modelo de procesos - Desarrollo de Software - Normas ISO - Validación de Software –Calidad de Software

Trabajos de Grado de Maestría (TGM)

Farez (2015). En su trabajo de grado de Maestría titulado “MODELO DE CALIDAD PARA LA GESTIÓN DE PROCESOS DE DESARROLLO DE SOFTWARE APLICADO AL SISTEMA INFORMÁTICO EN USO DE LA UNIVERSIDAD TÉCNICA DE MACHALA”, El objetivo de esta TGM de tipo investigativa fue elaborar un Modelo de Calidad para la Gestión del Proceso de Desarrollo de Software en la Unidad de Sistemas de la Universidad Técnica de Machala por medio de un análisis de los diferentes estándares de calidad, para finalmente proponer un modelo de calidad que permita optimizar la eficiencia y eficacia en el sistema desarrollado. Dentro de los logros más importantes de esta investigación se evidencia la creación de una estructura para la elaboración del modelo de calidad en el proceso de desarrollo software, dividido en tres áreas tales como: Gestión de proyectos, Ingeniería y Soporte; y al aplicar el modelo se logró que la Unidad de Sistemas aceptara el modelo de calidad en un 97%; se demuestra con la contribución que ha tenido el modelo aplicado de alguna manera para mejorar el proceso de desarrollo, para el diseño del modelo se hizo un análisis comparativo de los modelos y estándares de calidad aplicadas al proceso de desarrollo de software: CMMI (Capability Maturity Model), Estándar ISO 12207:2008, Norma ISO 15504 , ISO 12207.

Aporte: En este TGM se aporta un análisis comparativo de los diversos modelos o estándares aplicados a la calidad de sistema para mejorar el proceso de desarrollo de sistema, plantea un modelo de calidad que se ajuste a las necesidades de

garantizar la calidad de los procesos de desarrollo de sistema de un área específica. Analiza el proceso de desarrollo de sistemas.

Palabras clave: Software - Estándares de Calidad - Desarrollo de Software - Normas ISO - Normas IEEE - Universidad Técnica de Machala

García (2015). Una propuesta para el uso del paradigma guiado por modelos (MDE) para la definición y ejecución de procesos de negocios. La solución que plantea esta tesis doctoral es una solución práctica a la formalización de la definición y despliegue del proceso. Dicha formalización permitirá: almacenar el conocimiento y la experiencia de los ingenieros software que han usado el proceso en proyectos anteriores; y, la difusión, búsqueda, uso y reutilización de los activos de procesos y proyectos. La solución propuesta se basa en la reutilización del conocimiento y en mecanismos de colaboración que facilitarán el despliegue de los procesos definidos a través de un artefacto de conocimiento, los patrones y una estrategia .

Aporte: En este TGM se aporta un análisis comparativo de los diversos modelos o estándares aplicados a la calidad de sistema para mejorar el proceso de desarrollo de sistema, plantea un modelo de calidad que se ajuste a las necesidades de garantizar la calidad de los procesos de desarrollo de sistema de un área específica. Analiza el proceso de desarrollo.

Palabras clave: Proceso de Software – BMP –Procesos de Negocio – UML

Conlago (2014). PLAN DE MEJORA PARA EL PROCESO DE DESARROLLO DE SOFTWARE DE INTELLECTUS SYSTEM S.A. El objetivo de esta investigación fue Realizar un plan de mejora para el proceso de desarrollo de software de la empresa Intellectus utilizando el modelo CMMI. En esta Tesis se utilizó una investigación por

su naturaleza denominada aplicada y una técnica de campo, esta metodología permitió aumentar el conocimiento sobre el estado del arte para conocer cada uno de los elementos, sus características y cómo se interrelacionan entre sí, adicionalmente se utilizó el método inductivo que permitió, la observación de un caso, en la realidad concreta. En cuanto a las técnicas de investigación se aplicó una técnica de campo, la cual permitió observar y estar en contacto directo con el objeto de estudio planteado, se concluye que en el plan de mejora para el proceso de desarrollo de software de la empresa Intellectus se definieron las metas e indicadores de mejora para las áreas de proceso que presentan debilidades y fortalezas, áreas de proceso a implementar y para la meta genérica de Institucionalizar un proceso gestionado. El beneficio más importante es mejorar el proceso de desarrollo de software, lo que permitirá ejecutar el proceso, fundamentándose en un modelo específico y así cumplir con el tiempo de entrega, la utilización planificada de los recursos y satisfacer al cliente con el producto final.

Aporte: Se utilizan métodos y técnicas (CMMI y CMMI-DEV) para evaluar el nivel de madurez de los procesos de desarrollo de software en la empresa estudiada. Se diseñan metas e indicadores para el plan sugerido, las cuales sirven de referencia para este proyecto de tesis.

Palabras clave: CMMI, Nivel de Madurez, Proceso de Desarrollo, Producto de Calidad, Evaluación del Proceso.

Calderón (2013). Madurez y planificación estratégica de proyectos BPM en el sistema financiero peruano, cuyo objetivo fue Conocer el nivel de madurez de BPM, su adopción y en qué medida impactan en la excelencia operativa de las Entidades del Sistema Financiero Peruano (SFP), con el fin de impulsar un Mapa de Ruta a seguir hacia la madurez; todo esto; mediante el uso de un Modelo de Análisis de Madurez, una Planificación Estratégica de Proyectos BPM y la formulación de una guía de implementación de proyectos BPM. El diseño empleado fue no experimental, más bien observación directa.

Aporte: A pesar de que el estudio es para el sector financiero, conceptualmente aplica para el sector PYMES objeto de estudio, y aporta un material novedoso en cuanto a la forma como es adoptado el BPM, las preguntas que realiza en la investigación para hacer el análisis, pueden perfectamente ser formuladas para el sector PYMES en Venezuela; y partiendo de la premisa que la realidad se compone de Procesos, no de hechos aislados, constituye una base sólida para la integración de procesos.

Palabras clave: Gestión de Procesos de Negocio, Financiero (Perú).

Santamaría (2012). ESTUDIO PARA LA IMPLEMENTACIÓN DE ADMINISTRACIÓN DE PROCESOS DE NEGOCIO (BPM) EN LA FUERZA AÉREA COLOMBIANA. El Objetivo de esta investigación fue desarrollar una propuesta para aplicar la metodología BPM en la Fuerza Aérea colombiana con el fin de rediseñar los procesos de alto impacto estratégico actualmente contemplados en su Sistema de Gestión Integrado (SGI). Se desarrolla un estudio para la implementar BPM, para determinar el nivel de madurez en que se encuentra la Institución con respecto a la gestión de sus procesos. Se concluye que al implementar las alternativas de mejora propuestas, la FAC podría tener ahorros significativos en costos y beneficios en productividad.

Aporte: Esta investigación aporta un estudio para la implementación de BPM, establece un plan a seguir para que los procesos puedan pasar al siguiente nivel de madurez con CMMI. Aporta diseños que describen la lógica de los pasos de un proceso de Negocio utilizando la notación BPMN 2.0 (Business Process Model and Notation). Se utiliza esta notación en el desarrollo de la tesis.

Palabras clave: CMMI, BPM, BPMN, Procesos de Negocio, Mejora de Procesos.

Martínez (2012). Propuesta Integral de un Modelo de Gestión por Procesos de negocio (PIM-GPN). El objetivo de esta investigación fue Diseñar una metodología en base a las mejores prácticas que han demostrado un éxito sostenido. En la investigación primero se realiza un análisis de la situación actual de BPM en diferentes sectores de las organizaciones, luego se analizan las buenas prácticas propuesta por varios autores, luego se describe la metodología a usar PIM-GPN y finalmente aplican la metodología. Se concluye con la frase vale la pena centrarse en los procesos y en la gente, para lograr equipos de alto desempeño.

Aporte: En este TGM utilizan el estándar BPMN para modelar los procesos y explica detalladamente cada uno de los símbolos que se utilizan, se diseñan los procesos utilizando esta notación, lo cual sirve de referencia para este trabajo de grado.

Palabras clave: CMMI, BPM, BPMN, Procesos de Negocio, Mejora de Procesos

Artículos Técnicos

Aguinaga & Araujo (2013). Es un artículo arbitrado que habla del proceso de madurez del proceso de desarrollo de software respecto a CMMI, luego hacen un contraste con la Norma ISO 12207:2008, se realiza un mapeo de los procesos para evaluarlos y así poder determinar los procesos críticos, para finalmente proponer un plan de mejoras basado en las mejores prácticas de ITIL v3.

Aporte: En este artículo se visualiza como es evaluado el proceso de desarrollo de software con la norma bajo la norma ISO 12207 y ITIL v3; así como también elaboran un proceso de integración de software, los cuales constituyen una guía de mejores prácticas para los procesos de esta investigación.

Palabras clave: ISO 12207, ITIL, CMMI, Modelo Desarrollo, Proceso.

Orna (2013). Modelo de calidad para la etapa de planificación de Proyectos Medianos de Software. García (2015) las empresas creadoras de software, que en ocasiones genera insatisfacción del cliente, retrasos en las entregas e incumplimiento de lo acordado con el producto software desarrollado, se ha

determinado que la mayor razón para este suceso es que solo se ha limitado a codificar y corregir pasando por alto la utilización de normativas que regulan tanto el proceso como el producto final para de esta manera aportar calidad al producto entregado.

Aporte: Se describe el proceso general para la Planificación de Proyectos de Software, estándares para calidad de software ISO 9126-1, ISO/IEC 12207, ISO/IEC 15504, ISO 9126 y un estudio comparativo de las normas de planificación de Proyectos, valores agregados para ser aplicados en esta investigación.

Palabras clave: Estándares de calidad, Planificación de Proyectos, Software, Proceso.

2.2. Fundamentos Teóricos

2.2.1. Definición de Modelo

Según Carvajal (2002), un modelo es una representación ideal o aproximada de la realidad, de un objeto o cosa. Los modelos se construyen para representar un fenómeno estudiado, ayudan a resolver cualquier problema de la vida cotidiana. Es una razón por la que se decide en este trabajo de grado construir un modelo. En su artículo Teorías y Modelos: formas de representación de la realidad, indica que los modelos pueden expresarse de manera física, lingüística, simbólica o matemática. En la tabla N° 2 se definen los modelos desde el punto de vista epistemológico, se observa que el modelo analógico es un modelo que se construye en base a analogía de un proceso o sistema que se estudia, definición que aplica para el estudio del proceso de Desarrollo de Software que se llevan a cabo en las Pequeñas y Medianas Empresas en Venezuela.

Tabla N° 2. Tipos de Modelo

Tipo	Descripción	Ejemplo
Icónico	Este tipo de modelo se asemeja “directamente” a una propiedad o conjunto de propiedades de un hecho, proceso, sistema, entre otros. Puede ser de dos tipos: (a) una representación pictórica bidimensional y a escala. (b) una representación física total o parcial y a escala de lo que se quiere representar.	a) Dibujos, planos, mapas, imágenes, entre otros. (b) La maqueta de un edificio, la representación física del átomo o del ADN).
Analógico	Estos modelos pretenden una abstracción mayor que la anterior, y se construyen a partir de la representación por analogía: (a) un conjunto de cualidades o elementos, (b) una estructura y (c) un proceso, un fenómeno o sistema que se estudia. También, cuando se toma un modelo de una disciplina para aplicarlo a otra.	Se expresa lingüísticamente 1. El modelo del sistema Planetario. 2. El modelo del sistema planetario aplicado a la representación del átomo. 3. El proceso de desarrollo de una planta desde que nace hasta que muere aplicado a la historia de las culturas.
Topológico	La colocación de elementos en un plano de tal forma que se ordenan los elementos del sistema que se intenta representar, puede indicar las direcciones en un sistema de flujo.	Diagramas, cuadros, esquemas, mapas conceptuales, entre otros.
Simbólico	Este tipo de modelos, según Bisquera, tienen que ver con la fórmula o representación, por medio de símbolos del sistema, proceso o fenómeno que se estudia. El símbolo H_2O para representar el agua estudia (Bisquera; 1989: 44), Lo cual difiere de manera radical con la idea de “interconexión significativa de Conceptos” de Sierra (1984: 134) , en la cual se pierde la noción de isomorfismo, pues el modelo se construye a partir de otros conceptos.	Diagramas, cuadros, esquemas, mapas conceptuales, entre otros.
Matemático	Son representaciones aritméticas, esto es, un conjunto de proposiciones matemáticas; por tanto, utiliza los símbolos y las reglas de dicha disciplina. Este modelo permite inferir teoremas a partir de unas suposiciones o postulados.	Una ecuación o un algoritmo

Fuente: Carvajal (2002)

2.2.2 Los Procesos

2.2.2.1 Definición de Procesos

La palabra Proceso proviene del latín PROCESSUS que significa: avance, progreso. Macías (2007), define “proceso” como un conjunto de actividades y recursos interrelacionados que transforman elementos de entrada en elementos de salida aportando valor añadido para el cliente o usuario. Los procesos transforman elementos de entrada en elementos de salida. Los recursos pueden incluir personal, finanzas, instalaciones, equipos, técnicas y métodos. En el área de sistemas todo proceso tiene una entrada o conjunto de entradas, y una salida o conjunto de salidas. En Desarrollo de Software las entradas están representadas por los requerimientos de los usuarios, y las salidas están representadas por el producto del proceso, es decir, lo que el cliente solicitó. La entrada puede ser entonces, la solicitud de elaboración de un Reporte que muestre el listado de clientes de una determinada empresa, la salida esperada será un listado de los clientes. El proceso debe ser capaz entonces de procesar los insumos (las entradas), a través del uso de las tecnologías para producir el resultado esperado (las salidas). En la figura N° 3 se puede observar cómo se Representación gráficamente un Proceso.

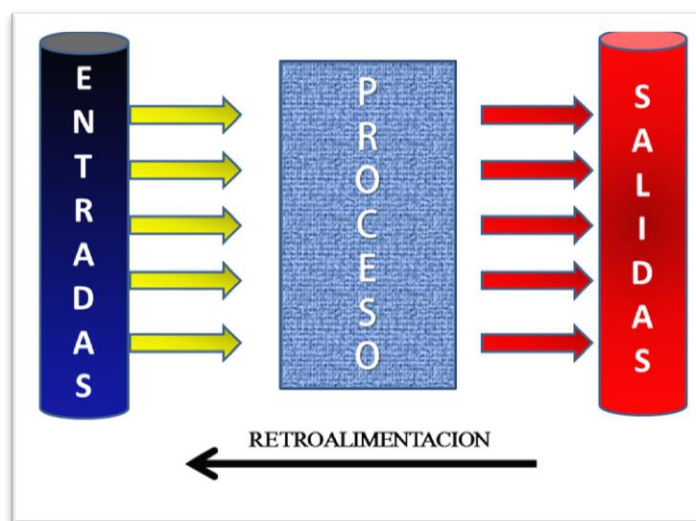


Figura N° 3. Representación General de un Proceso

Si se analiza una unidad administrativa o un equipo de trabajo en una organización, bajo este concepto, se puede decir que éstas reciben solicitudes (entradas), que son tramitadas (proceso) y finalmente suministran respuestas a las solicitudes (salidas). En la figura N° 4 se ilustra como los procesos mantienen unidos a las tres dimensiones fundamentales que soporta a toda organización: las personas, los métodos y procedimientos, y, el equipamiento y herramientas. El Instituto de Ingeniería de Software, (SEI) Software Engineering Institute (2010); explica que un enfoque en los procesos de la organización promueven una mejor productividad de las personas y un mejor uso de las tecnologías para ser más competitivos.

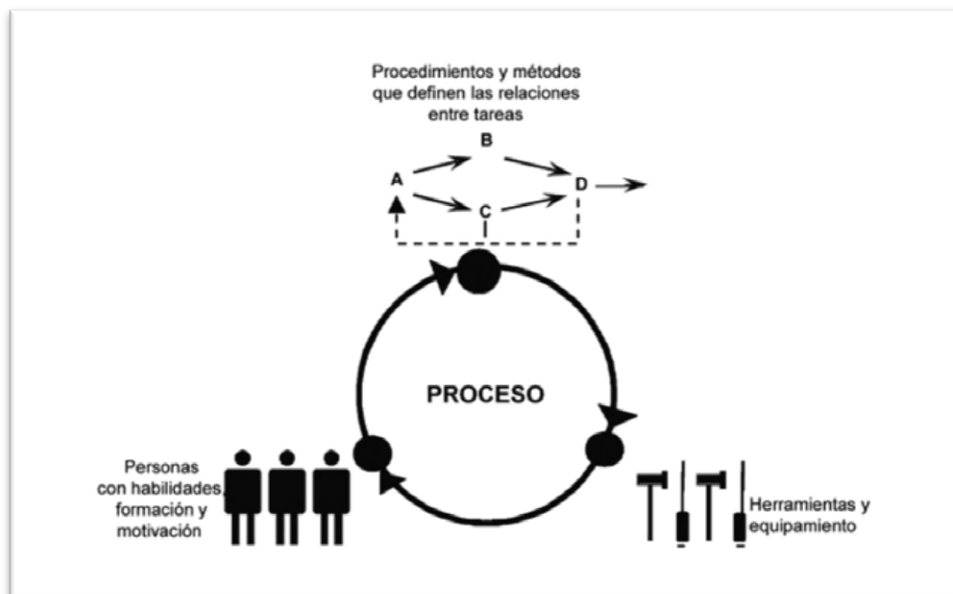


Figura N° 4. Enfoque en los Procesos
Fuente: Software Engineering Institute (2010)

Todo proceso tiene algunas características importantes de destacar:

1. Nunca dos resultados son idénticos. Cada vez que se repite el proceso hay ligeras variaciones en las distintas actividades realizadas.
2. Los procesos se crean para producir un resultado e intentar repetir ese resultado una y otra vez. A más repeticiones, más experiencia y mejores resultados. Crean una base de conocimientos.

3. El proceso tiene que ser fácilmente comprendido por cualquier persona de la organización.
4. Un proceso responde a la pregunta "QUE".
5. Es importante validar si el resultado del proceso es el esperado, de no serlo debe repetirse y mejorarse hasta obtener el resultado esperado. Hay que trabajar sobre el proceso y mejorarlo: Retroalimentación.

2.2.2.2 La Gestión Integrada por Procesos

Llanes-Font (2013) define La Gestión Integrada por procesos (GIP) como el proceso de integrar las etapas de planificación, realización, control y mejora de los procesos integrados del sistema organizacional, a su adaptación constante al entorno organizacional, a través del cumplimiento de los requisitos demandados por los clientes y de otras partes interesadas, a largo plazo y de modo equilibrado, con vistas a incrementar el nivel de integración del sistema. De la misma manera indica que la gestión por procesos tiene como objetivo mejorar los niveles de calidad y satisfacción de los clientes y usuarios; aumentar la productividad a través de la reducción de los tiempos y de los costos internos eliminando actividades que no agregan valor agregado al resultado. La norma ISO 9001:2015 habla de El enfoque basado en procesos, que permite a una organización planificar sus procesos y sus interacciones. Este enfoque, incorpora el ciclo Planificar-Hacer-Verificar-Actuar (PHVA).

En las organizaciones los procesos en una unidad o área administrativa no funcionan de manera aisladas, por lo general interactúan los unos con los otros, sin embargo, tienden a crearse "islas" que impiden la comunicación fluida entre los actores, trayendo como consecuencia la falta de integración de los procesos organizacionales. Uno de los grandes retos es, romper el paradigma de las "islas"; es por ello de vital importancia que se promueva la gestión integrada de procesos, donde unas áreas de la organización, conozcan e interactúen con las otras áreas, con la finalidad de prestar servicios de calidad tanto a clientes internos como clientes externos a la organización.

**Tabla N° 3 Principales diferencias entre Gestión por Procesos y
Gestión Integrada por Procesos.**

GESTIÓN POR PROCESOS	GESTIÓN INTEGRADA POR PROCESOS
El propósito es mejorar la eficiencia y eficacia de la organización para alcanzar los objetivos definidos	El propósito es mejorar la integración del sistema organizacional para alcanzar los objetivos definidos.
El documento básico es la ficha de proceso que describe esencialmente la etapa de planificación del proceso. No se incorporan los resultados del control y mejora del proceso	El documento básico es el expediente del proceso, que integra con la aplicación de herramientas informáticas, las etapas de planificación, operación, evaluación y mejora del proceso, es decir, se registra la historia del ciclo de vida del proceso.
En la descripción de las actividades del proceso se utilizan los esquemas "qué" y "quién- qué"	En la descripción de las actividades es indispensable la utilización del esquema "quién- qué" y la identificación para su gestión de los agujeros negros (se interpretan como desconexiones en las interfaces funcionales, que se manifiesta luego, en un sistema de procesos no armonizado), a través de los mapas funcionales integrados.
Se establecen para los objetivos del proceso, las categorías de medición de eficacia y eficiencia	Se establecen para los objetivos del proceso, las categorías de medición de eficacia, eficiencia y flexibilidad.
Prevalece el enfoque a sistema.	Prevalece el enfoque a sistema y el enfoque de "competencias integradoras" de los actores (roles) involucrados en el proceso.
Generalmente en las entradas del proceso, se identifican los requisitos aplicables (RA) del sistema de gestión a implementar.	En las entradas del "proceso integrado", se identifican todos los requisitos unificados aplicables (RUA), del marco legal- técnico - normativo que regula el proceso.
Prevalece la aplicación de las herramientas clásicas de la calidad.	Prevalece la aplicación de las herramientas clásicas de la calidad y las de la GIP, que propician la integración e informatización del proceso integrado.
Está sustentado en los ocho principios de gestión de la calidad.	Está sustentado en los ocho principios de gestión de la calidad y en el principio de gestión de los agujeros negros.

Fuente: Llanes-Font (2013)

2.2.2.3 Mapas de Procesos

La Norma ISO 9001:2015 sugiere el uso de esta herramienta por su enfoque basado en procesos, la cual facilita la redacción de procedimientos y el diseño de otros documentos, el conocer bien los procesos permite la concordancia entre los documentos. Los Mapas de Procesos son representaciones gráficas que permite a

las organizaciones identificar con cierto nivel de detalle, las tareas o pasos que afectan el resultado final de las actividades, en su diseño usualmente se captan elementos de entrada, proveedores, proceso, clientes y salidas. Para su elaboración es necesario conocer cuáles son las Entradas/Salidas, Proveedores/Clientes, Unidades de Apoyo o Habilitadores, Soportes o Recursos utilizados, así como las actividades detalladas que se realizan. Deben realizarse de lo macro a lo micro, es decir, de los más complejo y extenso a lo más sencillo y detallado.

Beneficios de utilizar Mapas de Procesos: Macías (2007):

- ✓ Permite comunicar y comprender de manera visual la información y reduce las confusiones frecuentes en una descripción narrativa
- ✓ El lector puede entender con facilidad, precisión y rapidez un proceso
- ✓ Establece su secuencia mediante (quién hace qué, cuándo, cómo, por qué y para qué).



Figura N° 5 Diagrama de un Mapa de Proceso

Fuente: Macías (2007)

Simbología: Los símbolos son signos o figuras con determinado significado (documento, decisión, inicio, etc.), que se interrelacionan para representar operaciones manuales o de cómputo de un proceso o procedimiento. Aunque se puede utilizar cualquier simbología, es necesario estandarizarla para facilitar la

lectura e interpretación de un diagrama, por lo general se apoya en el uso de los símbolos que se utilizan para construir diagramas de flujo.

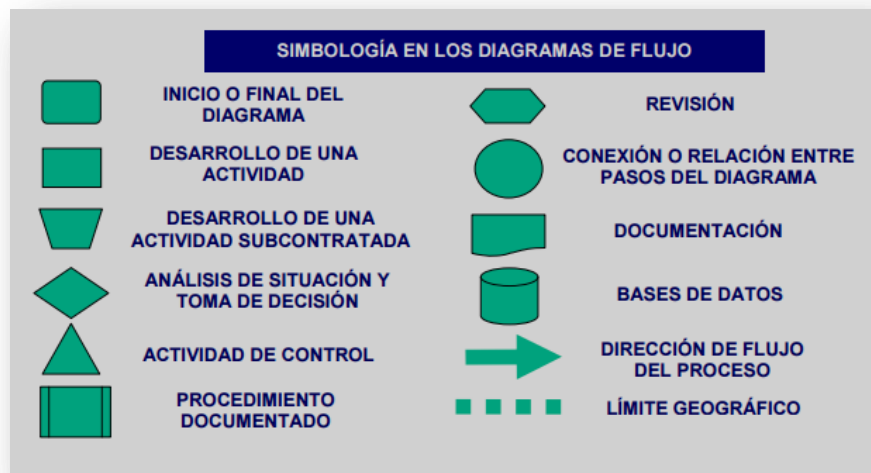


Figura N° 6. Simbología para Construir Mapas de Procesos
Fuente: Macías (2007)

2.2.2.4 Rediseño de Procesos de Negocio

Según Laudon & Laudon (2012), la tecnología de la Información permite cuatro tipos de cambios estructurales en las organizaciones: automatización, racionalización, rediseño del proceso de negocios y cambios de paradigma.

En la figura N° 7 se lee que los cambios de paradigmas tienen un alto riesgo, más sin embargo, el rendimiento es alto, por lo que pareciera recomendable hacer cambios de paradigmas para hacer a las organizaciones más rendidoras. Es importante tener en cuenta que un cambio de paradigma implica el uso de nuevas técnicas, nuevas herramientas y nuevas metodologías para mejorar y optimizar los procesos existentes, para cumplir con esta demanda hoy día, las organizaciones se apoyan en la administración del proceso de negocios, por sus siglas en inglés BPM (Business Process Management). La BPM impone desafíos, explicados con detalle más adelante.

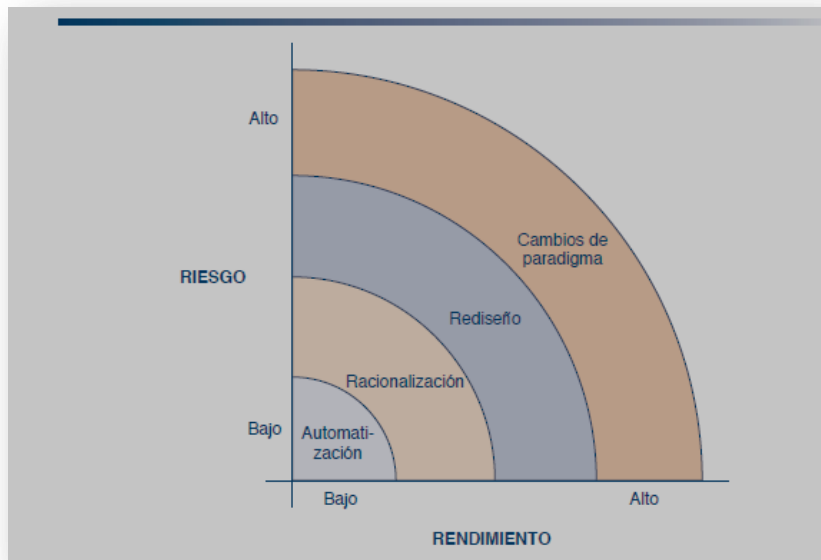


Figura N° 7. Riesgos del Cambio Organizacional
Fuente: Laudon & Laudon (2012)

Las compañías que desean rediseñar sus Procesos de Negocio deben pasar por las siguientes etapas:

- ✓ Identificar los procesos a cambiar
- ✓ Analizar los procesos existentes
- ✓ Diseñar el nuevo proceso
- ✓ Implementar el nuevo proceso
- ✓ Medición continua

2.2.3 Proceso de Desarrollo de Software

Sommerville (2005). Un proceso de Software es un conjunto de actividades cuya meta es el desarrollo o evolución del software.

Pressman (2010). En el contexto de la ingeniería de software, un proceso no es una prescripción rígida de cómo elaborar software de cómputo. Por el contrario, es un enfoque adaptable que permite que las personas que hacen el trabajo (el equipo de software) busquen y elijan el conjunto apropiado de acciones y tareas para el trabajo.

El proceso de desarrollo de software es propio de la Ingeniería Informática o Ingeniería de Software, como todo proceso, necesita de unos insumos (entradas), denominadas especificaciones del sistema, el procesamiento de estas especificaciones se hacen a través de líneas de código, configuración de ambientes, ejecución de al procedimiento o cualquier otra actividad técnica, que produce como resultado (salida) la satisfacción del cliente.

Tabla N° 4. Elementos del Proceso de Desarrollo de Software	
Comunicación	Se busca entender los objetivos de los participantes respecto del proyecto, y reunir los requerimientos que ayuden a definir las características y funciones del software.
Planeación	Llamado Plan del Proyecto de Software, define el trabajo de ingeniería de software al describir las tareas técnicas por realizar, los riesgos probables, los recursos que se requieren, los productos del futuro trabajo y una programación de las actividades.
Modelado	Crea un “bosquejo” del objeto por hacer a fin de entender el panorama general.
Construcción	Esta actividad combina la generación de código (ya sea manual o automatizada) y las pruebas que se requieren para descubrir errores en éste.
Despliegue	Se entrega al consumidor que lo evalúa y que le da retroalimentación, misma que se basa en dicha evaluación.
Fuente: Pressman (2010)	

2.2.3.1 Definición de Software

Pressman (2010), define el Software como la parte intangible del computador, como el elemento lógico de un sistema de cómputo y lo clasifica en software de sistemas operativos, software de redes, herramientas para desarrollo de software y ambientes de software. Desde el punto de vista de procesos se puede definir el software como un elemento lógico que procesa solicitudes (de una persona, de una máquina o de otro software), dichas solicitudes son procesadas bajo las reglas del negocio para el cual fue diseñado, y, arroja resultados y respuestas dado el procesamiento; en términos generales es un transformador de datos en información útil para quien la

requiere. Otra definición indica que es un conjunto de programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación.

2.2.3.2 Desarrollo de Software

Según Pressman (2010), no es más que un proceso, un conjunto de métodos y un arreglo de herramientas que permite a los profesionales elaborar software de cómputo de alta calidad. En la actualidad la necesidad cambiante del mercado exige cada día más desarrollar nuevas aplicaciones o modificar las existentes, es por ello que las personas se interesan cada día más en aprender y practicar el Desarrollo de Software como una actividad profesional. El producto final del desarrollo de un software por lo general resulta ser un Sistema de Información. Desarrollar un Software implica programar en un lenguaje entendible por la computadora, consiste en escribir una serie de comandos, que no son más que los requerimientos del cliente traducidos al lenguaje de “Máquina”.

2.2.3.3 Ingeniería de Software

Según, Pressman (2010), define la ingeniería de software como una tecnología con varias capas, y su fundamento se apoya en el compromiso con la calidad, la Ingeniería de Software es el establecimiento y uso de principios fundamentales de la ingeniería con objeto de desarrollar en forma económica software que sea confiable y que trabaje con eficiencia en máquinas reales. El IEEE ha desarrollado una definición más completa, como sigue: La ingeniería de software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software; es decir, la aplicación de la ingeniería al software. Finalmente, Sommerville (2005) define La Ingeniería de Software como una disciplina de la Ingeniería que abarca todos los aspectos de la producción de Software, desde su concepción hasta el mantenimiento luego de que ya está siendo utilizado por los usuarios finales.



Figura N° 8. Capas de la ingeniería de software
Fuente: Pressman (2010)

De la figura N° 8 se interpreta que la ingeniería de software se basa en un compromiso organizacional con la calidad. La capa del proceso define una estructura que debe establecerse para la obtención eficaz de tecnología de ingeniería de software. Los métodos de la ingeniería de software proporcionan la experiencia técnica para elaborar Software con calidad, incluye actividades de modelación y otras técnicas descriptivas. Las herramientas de la Ingeniería de Software proporcionan un apoyo para el proceso y los métodos. Cuando se integran las herramientas de modo que la información creada por una pueda ser utilizada por otra, queda establecido un sistema llamado ingeniería de software.

2.2.3.4 Sistemas de Información

Laudon & Laudon (2012), define un sistema de información como un conjunto de componentes interrelacionados que recolectan, procesan, almacenan y distribuyen información para apoyar los procesos de toma de decisiones y de control en una organización. Bajo esta definición se puede inferir que un Sistema de Información es un Software de uso específico, que permite ejecutar transacciones de ingreso, actualización y eliminación de datos; cumple con la función de almacenar datos, está orientado al uso administrativo, soportan la operatividad diaria en las organizaciones; por lo que tienen un alto impacto en el éxito de los negocios. De esta manera si el negocio es emitir pólizas de seguro, el sistema de información debe permitir ingreso de datos del cliente, la actualización de datos acerca del cliente, y si por ejemplo, es

una póliza de automóvil, debe permitir actualizar datos del vehículo asegurado, o emitir una póliza para otro vehículo del mismo cliente.

Actualmente las Pequeñas y Medianas Empresas de Venezuela, invierten grandes sumas de dinero en desarrollo de sistemas que le permitan obtener ventajas competitivas en el mercado, adaptarse a las regulaciones legales y mejorar la calidad de sus productos; inversión que es necesaria prever en los planes de crecimiento de las empresas. En la figura N° 9 Se muestra la inversión en capital de tecnología de la información, definida como hardware, software y equipo de comunicaciones, aumentó del 32 al 52 por ciento de todo el capital invertido entre 1980 y 2009.

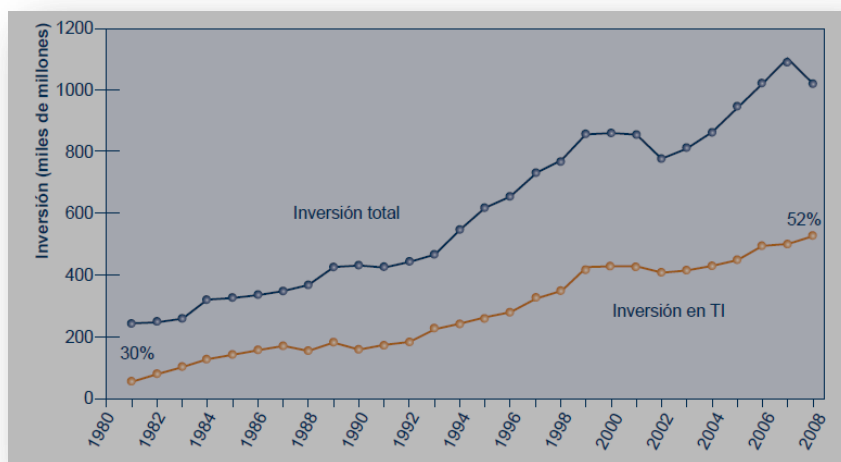


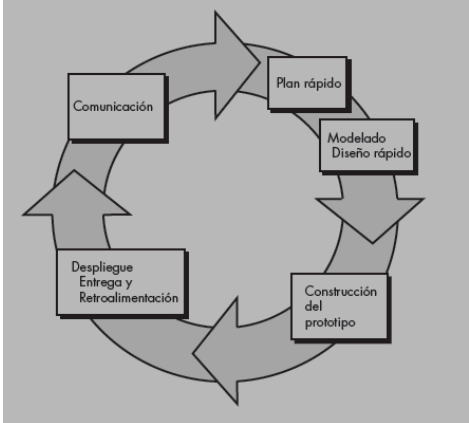
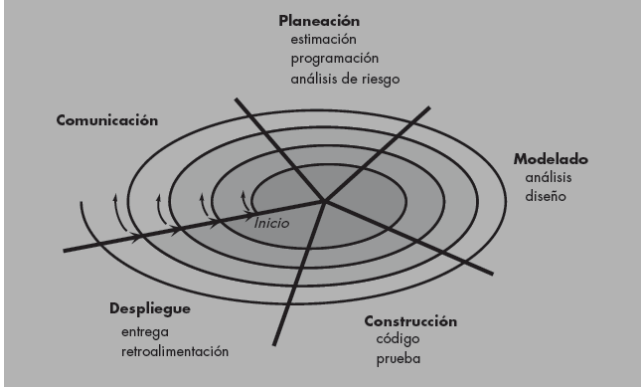
Figura N° 9. Inversión en capital de Tecnología de la Información
Fuente: Laudon & Laudon (2012)

2.2.3.5 Modelos de Desarrollo de Software

Según Pressman (2010), el término ciclo de vida del software describe el desarrollo de software desde la fase inicial hasta la fase final. Inicialmente se usaron los modelos de proceso prescriptivo para poner orden en el caos del desarrollo de software. A través del tiempo se ha intentado dar cierta estructura para constituir modelos que ayuden a los equipos de software a realizar su trabajo de manera eficiente. Sin embargo, se puede observar que hay muchos desarrolladores que no

utilizan esquemas, trabajan al borde del caos, trayendo como consecuencia que el producto final no tenga la calidad esperada. A continuación se muestran algunos de los modelos.

Tabla N° 5. Modelos Evolutivos para Desarrollo de Software	
Modelo Cascada Se comenzó a diseñar en 1966 y se terminó alrededor de 1970. Se define como una secuencia de fases donde al final de cada una de ellas se reúne la documentación para garantizar que cumple las especificaciones y los requisitos antes de pasar a la fase siguiente	<pre> graph TD A[Requerimientos] --> B[Diseño] B --> C[Implementación] C --> D[Integración] D --> E[Validación] E --> F[Instalación] </pre>
Modelo en V El modelo de ciclo de vida V proviene del principio que establece que los procedimientos utilizados para probar si la aplicación cumple las especificaciones ya deben haberse creado en la fase de diseño	<pre> graph TD A[Modelado de los requerimientos] --> B[Diseño de la arquitectura] B --> C[Diseño de los componentes] C --> D[Generación de código] D --> E[Software ejecutable] F[Pruebas de aceptación] --> G[Pruebas del sistema] G --> H[Pruebas de integración] H --> I[Pruebas unitarias] I --> E A --> F B --> G C --> H D --> I </pre>
Modelo Incremental El modelo incremental aplica secuencias lineales en forma escalonada a medida que avanza el calendario de actividades. Cada secuencia lineal produce “incrementos” de software susceptibles de entregarse.	Funcionalidad y características del software Calendario del proyecto Incremento # 1, Incremento # 2, Incremento # n entrega del primer incremento, entrega del segundo incremento, entrega del n-ésimo incremento Comunicación, Planeación, Modelado (análisis, diseño), Construcción (código, prueba), Despliegue (entrega, retroalimentación)

Modelo Evolutivo (Prototipos) El prototipo sirve como el primer sistema. Lo ideal es que el prototipo sirva como mecanismo para identificar los requerimientos del software. Si va a construirse un prototipo, pueden utilizarse fragmentos de programas existentes	 <pre> graph TD A[Comunicación] --> B[Plan rápido] B --> C[Modelado Diseño rápido] C --> D[Construcción del prototipo] D --> E[Despliegue Entrega y Retroalimentación] E --> A </pre>
Modelo Espiral Es un modelo evolutivo del proceso del software y se acopla con la naturaleza iterativa de hacer prototipos con los aspectos controlados y sistémicos del modelo de cascada.	
Fuente: Pressman (2010)	

Existen otros modelos para Desarrollo de Software, denominados por Pressman (2010) como modelos de proceso especializado, que poseen muchas de las características de los modelos evolutivos, entre ellos tenemos:

- ✓ *Desarrollo basado en componentes:* El objetivo principal del desarrollo basado en componentes es la reutilización del software, lo cual aporta varios beneficios en cuanto a la reducción en el ciclo de tiempo del desarrollo como el costo del proyecto.
- ✓ *Desarrollo de software orientado a aspectos:* es un paradigma de ingeniería de software relativamente nuevo que proporciona un proceso y enfoque metodológico para definir, especificar, diseñar y construir aspectos:

mecanismos más allá de subrutinas y herencia para localizar la expresión de una preocupación global”.

- ✓ *Proceso Unificado*: Utiliza UML, lenguaje de modelado unificado, para para describir el punto de vista del usuario respecto de un sistema. Es impulsado por el caso de uso, centrado en la arquitectura, iterativo e incremental.
- ✓ *Proceso Personal del Software (PPS)*: el PPS responsabiliza al profesional acerca de la planeación del proyecto, pone el énfasis en la medición personal tanto del producto del trabajo que se genera como de su calidad.
- ✓ *Proceso del Equipo de Software (PES)*: El objetivo de éste es construir un equipo auto dirigido para el proyecto, que se organice para producir software de alta calidad.

Luego de presentar todos los modelos anteriores, se puede concluir, que es importante seleccionar el modelo de Desarrollo de Software que mejor se adapte a las características propias de proyectos pequeños, de seguro será recomendable utilizar PSP, sin embargo, para proyectos de mayor envergadura, UML siempre será una herramienta idónea. Es importante siempre recordar que la calidad del proceso se ve reflejada en la calidad del producto final, de allí la importancia de hacer una buena elección.

2.2.4 Proceso de Levantamiento de Requerimientos

Bedini (2006), indica que la especificación de requerimientos es una descripción detallada y precisa de la funcionalidad del sistema teniendo en cuenta las restricciones del mismo; este proceso, es de vital importancia, en el ciclo de desarrollo de sistemas, ya que constituye el insumo principal para los desarrolladores de software, por lo tanto debe contener la especificación del tipo de producto de entrada, la especificación del tipo de producto de salida y la especificación del tipo de proceso (proceso de diseño), que permita hacer el trabajo de desarrollo de software fluidamente. Llevar a cabo de manera adecuada el Proceso de Levantamiento de

Requerimientos disminuye considerablemente la probabilidad de fracaso de un proyecto, contribuye a obtener un producto de calidad.

2.2.4.1 Ingeniería de Requerimientos

Sommerville (2005) indica que la Ingeniería de Requerimientos, se define como una actividad que trata en parte de la identificación del propósito del software que se desea construir, el propósito es relativo a actividades humanas, está lejos del software mismo, identifica todas las partes involucradas no sólo el usuario y el cliente. La Ingeniería de Requerimientos desglosa las tareas y técnicas que llevan a entender los requerimientos, responde inquietudes del cómo y donde el sistema será utilizado, se divide en el Desarrollo de Requerimientos y la Gestión de Requerimientos.

Según Sommerville (2005), los requerimientos se pueden clasificar en:

- ✓ **Requerimientos Funcionales:** Los requerimientos funcionales son afirmaciones de los servicios que el sistema debe proveer, como el sistema debe reaccionar a entradas particulares y como el sistema debe comportarse bajo situaciones particulares. En algunos casos los requerimientos funcionales deben describir de manera explícita, lo que el sistema no debe hacer.
- ✓ **Requerimientos NO Funcionales:** Son los requerimientos necesarios para que el sistema opere pero no necesariamente son parte de los requerimientos del usuario; no son funciones específicas que entrega el sistema, definen las restricciones del sistema se refieren a las propiedades emergentes de éste como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. Por ejemplo: el sistema debe correr en sistema operativo Linux, la conexión a internet debe ser superior a 128 Kbps.

Desde el punto de vista de procesos Levantar un Requerimiento es un proceso donde existen entradas de datos proporcionada por el usuario solicitante, cuya salida es el resultado escrito en forma narrativa y grafica de lo que expresó verbalmente el usuario solicitante.

Tabla N° 6. Principios para Levantamiento de Requerimientos	
Separar funcionalidad de implementación.	Las especificaciones deben describir que se desea realizar, no cómo se va a realizar.
Usar lenguaje de especificación orientado a procesos.	Usar una descripción orientada al proceso, en la cual la especificación se obtiene mediante un modelo de comportamiento deseado en términos de respuestas funcionales ante distintos estímulos del entorno.
Abarcar el sistema del cual el software es un componente.	Sólo dentro del contexto del sistema completo y de la interacción entre sus partes puede ser definido el comportamiento de un componente específico.
Abarcar el entorno en el cual opera el sistema.	Es necesario reconocer que el propio entorno es un sistema compuesto de objetos que interactúan, pasivos y activos.
Debe ser un modelo cognitivo.	Debe describir un sistema tal como es percibido por su comunidad de usuarios y los objetivos que manipula deben corresponderse con objetivos reales de dicho dominio.
La especificación debe ser operacional.	La especificación debe ser completa y lo suficientemente formal para que pueda usarse para determinar si una implementación propuesta satisface la especificación.
La especificación debe ser tolerable con la incompletitud.	Las herramientas de análisis que apoyan y prueban la especificación deben ser capaces de tratar con la incompletitud.
Debe ser localizada y débilmente acoplada.	Los principios anteriores tratan con la especificación como una entidad estática, la realidad es que debe servir como base al diseño y la implementación
Fuente: Bedini (2006)	

Según Pressman (2010), La ingeniería de Requerimientos incluye siete actividades diferentes: concepción, indagación, elaboración, negociación, especificación, validación y administración.

- ✓ *Concepción:* ¿Cómo inicia un proyecto de software? La mayor parte de proyectos comienzan cuando se identifica una necesidad del negocio o se descubre un nuevo mercado o servicio potencial.

- ✓ *Indagación:* Cómo va a usarse el sistema o producto de software en las operaciones cotidianas; preguntar al cliente, a los usuarios y a otras personas cuáles son los objetivos para el sistema o producto.
- ✓ *Elaboración:* Esta tarea se centra en desarrollar un modelo refinado de los requerimientos. Cada escenario de usuario se enuncia con sintaxis apropiada para extraer clases de análisis, elabora modelos conceptuales.
- ✓ *Negociación:* se evalúa su costo y riesgo, y se enfrentan los conflictos internos; algunos requerimientos se eliminan, se combinan o se modifican de modo que cada parte logre cierto grado de satisfacción. Se pide a clientes, usuarios y otros participantes que ordenen sus requerimientos según su prioridad y que después analicen los conflictos.
- ✓ *Especificación:* Se elige ó diseña un formato de especificaciones de Requerimientos de Software. Una especificación puede ser un documento escrito, un conjunto de modelos gráficos, un modelo matemático formal, un conjunto de escenarios de uso, un prototipo o cualquier combinación de éstos.
- ✓ *Validación y administración:* ¿Los requerimientos están enunciados con claridad? ¿Podrían interpretarse mal? La validación de los requerimientos analiza la especificación. Asegura la calidad de los productos del requerimiento.

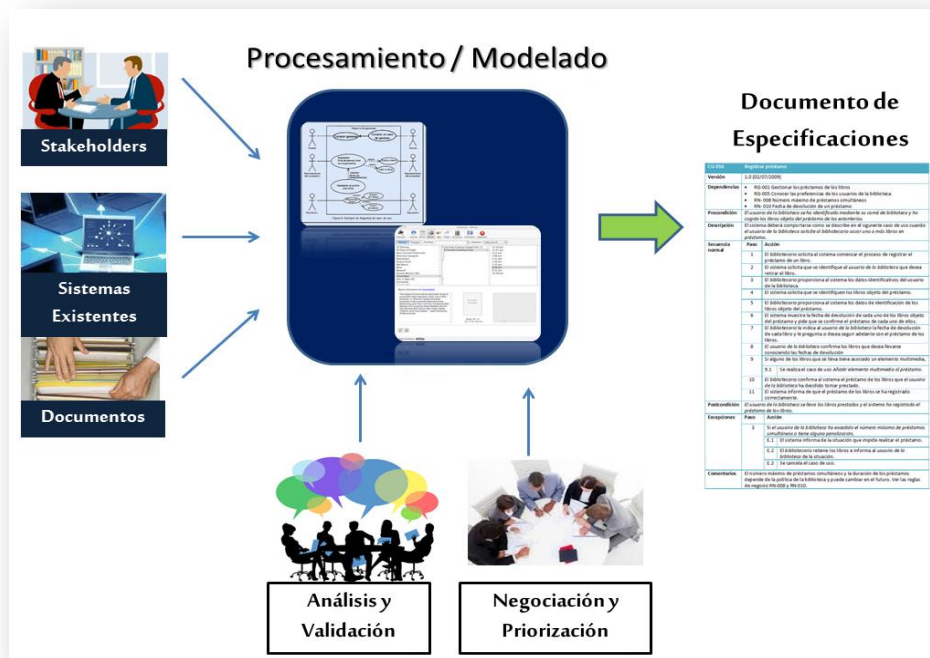


Figura N° 10. Proceso General de Levantamiento de Requerimientos.

2.2.4.2 Documento de Especificación de Requerimientos (ERS)

IEEE (2010). El estándar IEEE 830-1998 para el SRS (por sus siglas en inglés, Software Requirements Specification), define La Especificación de Requerimientos de Software como un conjunto de recomendaciones para la especificación de los requerimientos o requisitos de software el cual tiene como producto final la documentación de los acuerdos entre el cliente y el grupo de desarrollo para así cumplir con la totalidad de exigencias estipuladas. La persona encargada de Especificar los Requerimientos del sistema debe poseer un conjunto de habilidades que le permitan desempeñar eficientemente el trabajo que se le ha encomendado, alguna de ellas son: saber entrevistar, escuchar, cuestionar (pensamiento crítico), modelar, analizar, facilitar discusiones y negociaciones, observar, comunicar de manera verbal y escrita, relacionarse con gente, innovar, identificar muy bien el problema para poder describir una solución acorde a los esperado.

2.2.5 Proceso de Calidad de Software

A continuación se describen los términos de: Calidad de Software, Estándar ISO 9126 y ISO/IEC 25000, para conocer como cada uno de ellos orienta la calidad desde el punto de vista del software.

2.2.5.1 Calidad de Software

El vocabulario de la Norma ISO 9000:2015, en su apartado 2.2.1 define la calidad de los productos y servicios de una organización está determinada por la capacidad para satisfacer a los clientes. La calidad de los productos y servicios incluye no solo su función y desempeño previstos, sino también su valor percibido y el beneficio para el cliente.

Según los autores Aguirre & La Serna (2015), La Calidad de Software (SQA-Software Quality Assurance) es un proceso donde los esfuerzos de Verificación y Validación (V & V) tienen como objetivo asegurar que el software satisfaga los requisitos del usuario y la calidad esperada. Suárez & Garzás (2014) indican que la calidad del software es un aspecto básico a tener en cuenta por las organizaciones que desarrollan y compran aplicaciones informáticas. Debe cumplir con un nivel adecuado de usabilidad, fiabilidad y seguridad. Se puede clasificar la calidad del software básicamente en tres tipos de calidad:

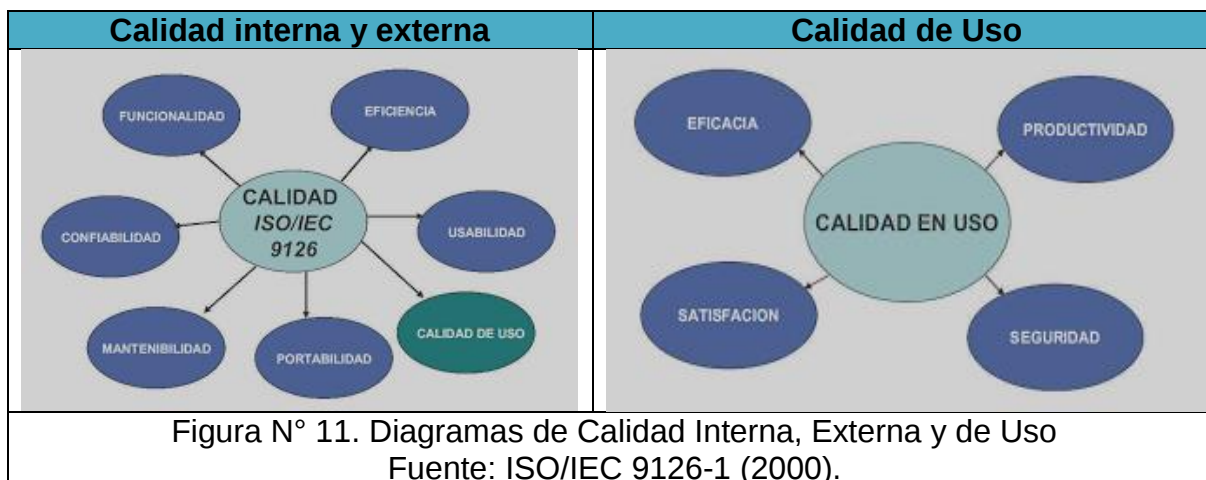
- ✓ *Calidad del proceso*: se refiere al cómo fue elaborado el software, como se desarrollaron las actividades y el proceso para lograr el producto. De allí la importancia de tener documentados los procedimientos de desarrollo de software.
- ✓ *Calidad del producto*: Se refiere al producto final a los atributos estéticos, velocidad de respuesta, La calidad del insumo principal (requerimientos) determina sustancialmente la calidad del producto.
- ✓ *Calidad del Uso*: Se refiere al ambiente donde se emplea el producto final, respondería a las siguientes preguntas: ¿es fácil de usar? ¿se puede usar el aplicativo o sistema bajo la infraestructura actual de la empresa?

Suárez & Garzías (2014). La calidad vista desde el mundo de los procesos nos dice que la calidad del producto software está determinada por la calidad del proceso. Por proceso se entienden las actividades, tareas, entrada, salida, procedimientos, etc., para desarrollar y mantener software. Según la literatura, recopilado de varios autores se puede decir que la importancia de adoptar Estándares de Calidad del Software son:

- ✓ Permite corregir los procesos de software que se hayan desajustado con el tiempo.
- ✓ Genera una cultura organizacional enfocada a cumplir con los requisitos de los clientes.
- ✓ Reducir los costos en todos los procesos, aumentar la productividad, efectividad y utilidad de la empresa.
- ✓ Asegurar la satisfacción de los clientes internos y externos, suministrando productos y servicios de software y con valor agregado.
- ✓ Promueve la medición y generación de indicadores, que permiten, para conocer fortalezas y debilidades de la empresa; y establecer las estrategias necesarias para realizar mejoras.

2.2.5.2 Estándar ISO 9126

ISO 9126 es un estándar internacional para la evaluación de la calidad del software, fue publicada en 1992, también conocido como (o ISO/IEC 9126), define la Calidad del Software como: “La totalidad de características de un producto de software que se manifiesta en su habilidad para satisfacer necesidades establecidas o implícitas”. Este estándar describe 6 características generales: Funcionalidad, Confiabilidad, Usabilidad, Eficiencia, Mantenibilidad, y Portabilidad.



La Calidad interna y externa: Especifica 6 características para calidad interna y externa, las cuales, están subdivididas. Estas divisiones se manifiestan externamente cuando el software es usado como parte de un sistema Informático, y son el resultado de atributos internos de software.

Calidad en uso: Calidad en uso es la calidad del software que el usuario final refleja, la forma como el usuario final logra realizar los procesos con satisfacción, eficiencia y exactitud. La calidad en uso debe asegurar la prueba o revisión de todas las opciones que el usuario trabaja diariamente y los procesos que realiza esporádicamente relacionados con el mismo software.

Tabla N° 7. Resumen Estándar ISO 9126	
Característica	Subcaracterísticas
Funcionalidad	Adecuacion, Exactitud Conformidad de la funcionalidad Interoperabilidad, Seguridad
Confiabilidad	Madurez, Recuperabilidad Conformidad de Confiabilidad Tolerancia a errores
Usabilidad	Entendimiento, Aprendizaje Conformidad de Usabilidad Atracción, Operabilidad
Eficiencia	Comportamiento de tiempos Conformidad de Eficiencia Utilización de Recursos
Capacidad de Mantenimiento	Capacidad de ser analizado Conformidad de Mantenimiento Facilidad de Prueba, Estabilidad, Cambiabilidad
Portabilidad	Adaptabilidad, Coexistencia Conformidad de Portabilidad Reemplazabilidad, Facilidad de Instalacion
Fuente: ISO/IEC 9126-1 (2000).	

2.2.5.3 ISO/IEC 25000

La familia de normas ISO/IEC 25000 es conocida como SQuaRe (Software Product Quality Requirements and Evaluation), tiene por objetivo la creación de un marco de trabajo para evaluar la calidad del producto software, sustituye a las anteriores ISO/IEC 9126 e ISO/IEC 14598 y convirtiéndose así en el referente a seguir. Según Marcos, Arroyo, Garzás, & Piattini (2008), la ISO/IEC 25000 se encuentra compuesta de varias partes o divisiones, entre las que podemos destacar:

- ✓ *La ISO/IEC 25010*: determina las características de calidad del producto software que se pueden evaluar. En total son 8 las 47 características de calidad que identifica: funcionalidad, rendimiento, compatibilidad, usabilidad, fiabilidad, seguridad, mantenibilidad y portabilidad.

- ✓ *La ISO/IEC 25040*: define el proceso de evaluación de la calidad del producto software, compuesto por cinco actividades:
 - Establecer los requisitos: para determinar cuáles son los requisitos de calidad que se deben considerar a la hora de evaluar el producto.
 - Especificar la evaluación: indicando las métricas, criterios de medición y evaluación a tener en cuenta.
 - Diseñar la evaluación: definiendo el plan de actividades que se diseñaron para evaluar el producto.
 - Ejecutar la evaluación: realizando las actividades de medición y evaluación del producto, considerando los criterios identificados en las fases previas. o Concluir la evaluación: elaborando el informe de evaluación y realizando la disposición de resultados e ítems de trabajo.
- ✓ La ISO/IEC 25020 que será la encargada de definir las métricas de calidad del producto software y que todavía está en pendiente de publicación, por lo que no existe un acuerdo respecto a los indicadores y umbrales que se deben considerar para poder determinar la calidad de un producto software de manera estandarizada.

2.2.6 Proceso de Gestión de Proyectos

2.2.6.1 Definición

El Project Management Institute, PMI (2013) define la dirección de proyectos como la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades del proyecto para cumplir con los requisitos del mismo. Se logra mediante la aplicación e integración adecuadas de los 47 procesos de la dirección de proyectos, agrupados de manera lógica, categorizados en cinco Grupos de Procesos, que son:

- ✓ Inicio
- ✓ Planificación
- ✓ Ejecución
- ✓ Monitoreo y Control
- ✓ Cierre

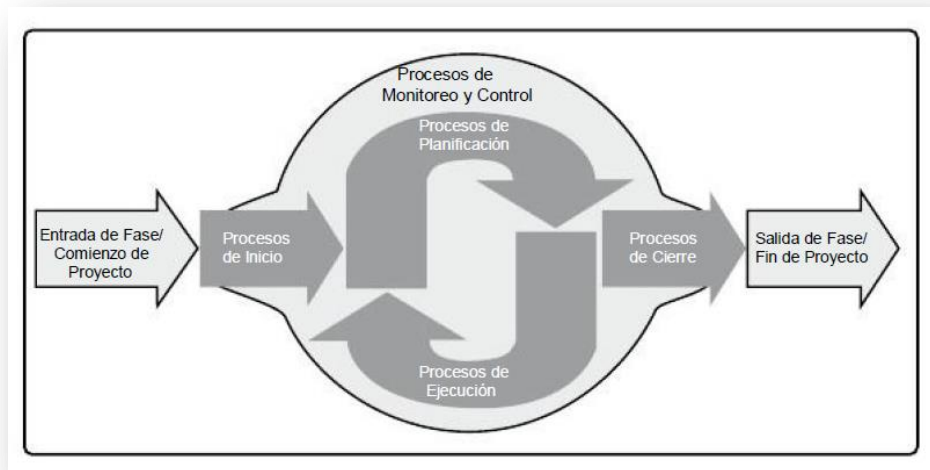


Figura N° 12. Grupos de Procesos de la Dirección de Proyectos
Fuente: PMI (2013)

Un proyecto es entonces, un esfuerzo temporal pero continuo, que se lleva a cabo para crear un producto, un servicio o resultado único. Los proyectos tienen un principio y un final definidos. El final del proyecto llega cuando: se logran los objetivos del proyecto, cuando sus objetivos no se cumplen o no pueden ser cumplidos, o cuando ya no existe la necesidad que dio origen al proyecto.

2.2.6.2 Proyecto de Software

En los proyectos de Desarrollo de Software las actividades de gestión son aquellas relacionadas con la administración de las personas, sistemas y procedimientos comprometidos en el proceso de planificación y construcción del sistema. Este tipo de proyectos se diferencia de los proyectos tradicionales en que el producto es intangible. El administrador de proyectos depende de la documentación disponible para revisar el progreso. Los principales objetivos de la planificación en proyectos de desarrollo de software es ordenar las actividades del proyecto, establecer hitos, asignar adecuadamente los recursos y tareas, medir el avance, mejorar la comunicación para cumplir los objetivos propuestos del proyecto. (Bedini González, 2006).

Para garantizar una gestión eficiente de Control y Seguimiento de Proyectos que permita organizar y administrar los recursos de manera tal que se pueda culminar todo el trabajo requerido en el proyecto dentro del alcance, tiempo y coste definidos debe construirse un Plan del Proyecto de Desarrollo de Software, documentado, distribuido, entendido y comprometido; que resulta del proceso de planificación, el cual debe contener:

2.2.7 Qué es una Métrica

El IEEE Standard Glossary of Software Engineering Terminology define métrica como una medida cuantitativa del grado en el que un sistema, componente o Proceso posee un atributo determinado. La medición es un mecanismo clave en cualquier proceso de ingeniería. En Ingeniería de Software, se establecen métricas para medir tiempos de desarrollo de software, cantidad de errores en el software, nivel de satisfacción del cliente y para medir la calidad del proceso y del producto final. Existen un conjunto de Métricas que aportaran valor agregado en los cuatro procesos objeto de estudio de este trabajo de investigación de Desarrollo de Software, Calidad de Software, Gestión de Proyectos y Requerimientos de Software.

2.2.7.1 Métricas de Requerimientos de Software

Suárez & Parra (2014) Las métricas de requerimientos de software proveen una forma cuantitativa para valorar la calidad de los Requisitos de Software, desde la perspectiva del proceso que se lleva a cabo para poder hacer el levantamiento de información y desde la perspectiva de la calidad del Requerimiento levantado.

2.2.7.2 Métricas de Desarrollo de Software

Suárez & Parra (2014) indican que familia de normas ISO/IEC 25000 todavía no ha definido el conjunto de métricas e indicadores que afectan las métricas de software, sin embargo, se conoce que proveen una forma cuantitativa para valorar la calidad del software en su construcción. Una métrica de software debe ser simple y calculable, convincente, congruente y objetiva; debe ser independiente del lenguaje

de programación y ofrecer retroalimentación efectiva .A continuación se mencionan algunas de ellas son:

Tabla N° 8. Métricas de Desarrollo de Software	
Para cuantificar la utilidad	✓ Modularidad ✓ Reusabilidad ✓ Mantenibilidad ✓ Tamaño del software
Para cuantificar la cantidad de funcionalidad	✓ Entradas de usuario ✓ Salidas de usuario ✓ Interfaces externas ✓ Archivos
Para cuantificar el tamaño del software	✓ Líneas de Código ✓ Cantidad de Objetos de base de datos ✓ Cantidad de objetos de aplicación ✓ Cantidad de reportes ✓ Cantidad de data generada
Para cuantificar las fallas en el proceso de desarrollo	✓ Número de veces que fue modificado un objeto en el proceso de pruebas
Fuente: Suárez & Parra (2014)	

2.2.7.3 Métricas de Calidad de Software

Las métricas de calidad proporcionan los atributos a medir y las variaciones permitidas. Un software con calidad no debería arrojar errores, sin embargo, siempre quedan brechas en funcionalidades que no son desarrolladas por falta de especificadores con un nivel de detalle muy elevado, o por una pobre interpretación del requerimiento; en este escenario es aceptable entonces que surjan en las pruebas de calidad ampliación de especificaciones o cambios de alcances.

Tabla N° 9. Métricas de Calidad de Software	
Para cuantificar la cantidad de errores del aplicativo	✓ Defectos detectados ✓ Defectos que no aplican
Para cuantificar las pruebas	✓ Número de casos de prueba en el plan ✓ Frecuencia de prueba de un proceso o funcionalidad ✓ Tiempo estimado para completar una tarea
Para Evaluar Requisitos Funcionales	✓ Número de funcionalidades que no cumplieron con los requisitos
Para cualificar el producto	✓ Diseño ✓ Confiabilidad ✓ Velocidad de respuesta
Fuente: Suárez & Parra (2014)	

2.2.7.4 Métricas de Gestión de Proyectos

Permiten analizar la eficiencia del proceso, durante el periodo de vida del proyecto. Controlar la Calidad es el proceso de monitorear y registrar los resultados de la ejecución de las actividades de calidad, a fin de evaluar el desempeño y recomendar los cambios necesarios. Al culminar un Proyecto de Desarrollo de Software se puede evaluar desde diferentes aristas: en el periodo de vida del proyecto o luego de culminado el proyecto.

Tabla N°10. Métricas de Gestión de Proyecto de Software	
Durante periodo de vida del proyecto	✓ Porcentaje de Avance ✓ Porcentaje de Ejecución ✓ Cantidad de Recursos Humanos
Ya finalizado el proyecto	✓ ¿Se terminó el proyecto en el tiempo establecido? ✓ ¿Cuánto tiempo más? ¿tiempo menos? ✓ Oportunidades de mejora ✓ Costo del Proyecto
Fuente: Suárez & Parra (2014)	

2.2.8 BPM – Gerencia de Procesos de Negocio

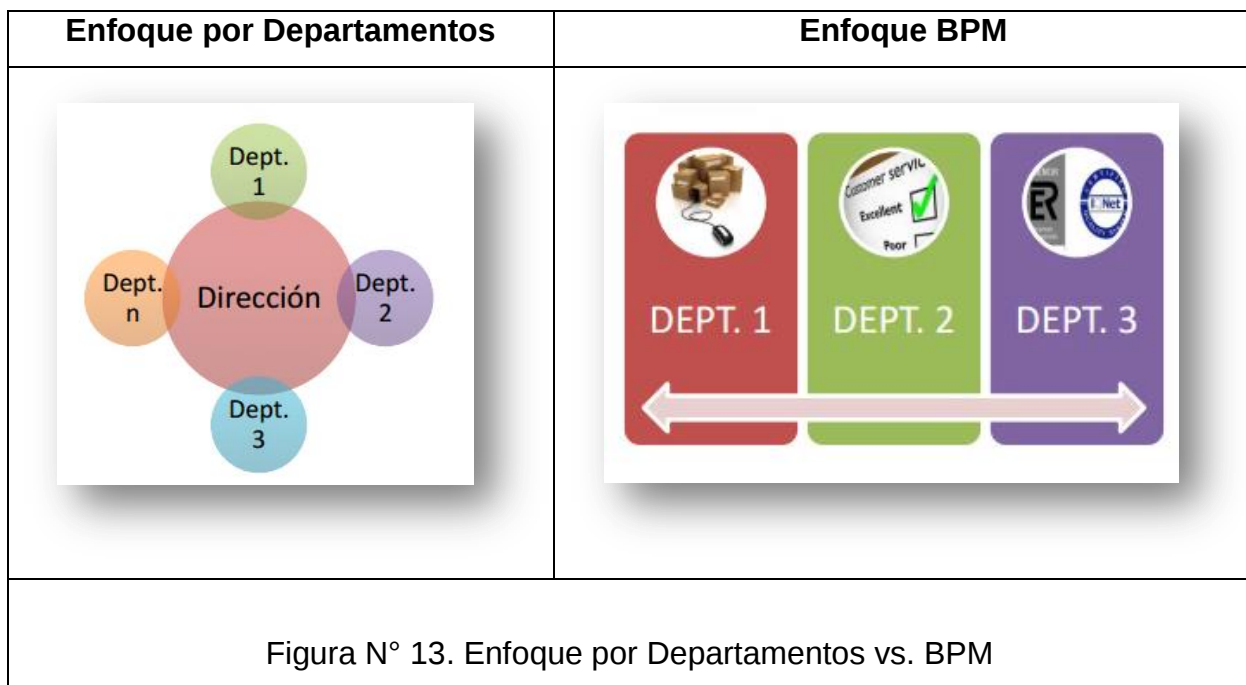
2.2.8.1 Definición BPM

BPM (2011) - Business Process Management, que traducido al español significa Gerencia de Procesos de Negocio. Se entiende que los procesos de negocio son una secuencia de actividades que deben dar soporte a la estrategia y analizar la efectividad. En un flujo de principio a fin este enfoque de gestión, le permite a las organizaciones monitorizar los procesos de negocio.

Robledo & Farrington (2011). BPM, va más allá del aspecto tecnológico, es un sistema de gestión enfocado a perseguir la mejora continua del funcionamiento de las actividades empresariales mediante la identificación y selección de procesos y la descripción, documentación y mejora de los mismos, partiendo del despliegue de la estrategia de la organización, asegurando la misión empresarial y alineada a la visión de la empresa.

Tradicionalmente las empresas cuentan con una estructura organizativa, que define como están operan los procesos de cada unidad administrativa o gerencia. El paradigma de BPM es hacer que los procesos de cada una de esas unidades administrativas sean transversales y menciona Freund, Rucker & Hitpass (2014) que debe tener objetivos bien claros y definidos:

- ✓ Hacer el negocio más ágil, siendo capaz de adaptarse rápidamente a los cambios externos, a través de los cambios de sus procesos integrados.
- ✓ Lograr mayor “Eficacia”: entendiéndose por eficacia como la capacidad que tiene la organización para lograr los objetivos estratégicos del negocio.
- ✓ Mejorar los niveles de Eficiencia: lo que indica una mejora intrínseca de la productividad, que nos es más que la relación directa entre el tiempo y los costos.



El rediseño del proceso de negocios produce cambios exitosos en las organizaciones, por lo que reorganiza los flujos de trabajo; combina los pasos para reducir el desperdicio y eliminar las tareas repetitivas que requieren de mucha papelería; este cambio de paradigma puede fracasar debido a que es muy difícil de orquestar el cambio organizacional.

2.2.8.2 Beneficios de Implementar BPM

El libro del BPM (BPM, 2011), menciona que con BPM se consigue reducir costes, reducir tiempos, mejorar la calidad, mejores rendimientos, mayor motivación de los empleados y satisfacción del cliente, porque se poseen mejores procesos.

Además, el uso de la Gestión por Procesos en la empresa nos va a permitir fundamentalmente a:

- ✓ Aumentar los niveles de productividad y competitividad a través de la automatización y gestión de sus procesos de negocio.
- ✓ Proporcionar agilidad para adaptarse a los cambios del mercado y el entorno.

- ✓ Facilitar la adopción y el cumplimiento de una estrategia, permitiendo tomar acciones preventivas o correctivas en el momento oportuno.
- ✓ Mejorar la eficiencia y la efectividad.
- ✓ Disponer de la agilidad y la flexibilidad adecuadas para la innovación y adaptación al cambio.
- ✓ Disponer de una trazabilidad completa de las actividades ofreciendo un control absoluto de lo que ocurre en cada proceso.
- ✓ Aumentar la productividad al detectar y corregir el uso ineficiente de los recursos (tiempos muertos, cuellos de botella).
- ✓ Satisfacer al cliente interno y externo al mejorar la atención y resolviendo de forma proactiva las incidencias.
- ✓ Tomar decisiones más rápidas con información de cada proceso en tiempo real.
- ✓ Cumplir con las nuevas regulaciones (leyes y normativas).

Por otro lado, las organizaciones que manejan sus procesos con el paradigma BPM son capaces de determinar rápidamente:

- ✓ Quién es el responsable del proceso.
- ✓ Dónde se encuentra parado un proceso.
- ✓ Porqué, cómo y cuándo se arranca un proceso.
- ✓ Que ocurre cuando alguien no complementa algún dato de un proceso
- ✓ Cómo, cuándo y a quién se avisa cuando no se cumple un plazo en una actividad.
- ✓ Cómo interactúan los procesos entre sí.

2.2.8.3 Recomendaciones al Implementar BPM

El Club BPM (2011) menciona que antes de iniciar a modelar los procesos del negocio con cualquier herramienta comercial seleccionada, es necesario estudiar y analizar bien el proceso desde la abstracción y evaluar como un conjunto de entradas y salidas, lo cual ayuda a entender cuáles son los insumos y como deben ser utilizados, y cuáles son las salidas esperadas. Para ello:

- ✓ Hay que Definir el Producto: es recomendable hacer una lista cuales son las actividades que lo constituyen
- ✓ Desarrollar una narrativa escrita del proceso: describiendo el “qué” en términos de los resultados esperados, el “porqué” y el “cómo” del sistema
- ✓ Identificar las partes interesadas y las interacciones con el sistema.
- ✓ Coloque al usuario en el centro del diseño: haga entrevistas personalizadas, la idea es involucrarlo desde el principio
- ✓ Identifique sus herramientas de trabajo, por ejemplo las hojas de cálculo
- ✓ Hacer un mapa de procesos
- ✓ Identificar Objetivos, Resultados, Recursos y Participantes
- ✓ Definir Métricas de Rendimiento y Criterios de Éxito



Figura N° 14. Siete (7) pasos para el éxito de BPMS

Fuente: Club BPM (2011).

2.2.8.4 Business Process Management Suite (BPMS)

En el Libro del BPM, BPM (2011), describe BPMS como el conjunto de herramientas de software que soporta BPM, y que debe contener, al menos, las siguientes funcionalidades: Diagramador de Proceso para dibujar el proceso, Motores de

Orquestación para coordinar la secuencia de actividades según los flujos y reglas de las Clases de Proceso, Diseñador de formularios para definición de formularios e informes que no están disponibles en los componentes que se necesitan para apoyar el proceso de diseño. Inteligencia de Procesos: Herramientas BAM (Business Activity Monitoring), BI (Business Intelligence), Cuadro de Mandos, KPIs (Key Performance Indicators), etc.

BPMS, tiene un enfoque cien por ciento práctico por lo que exige que se afine tanto que deben determinarse todas las posibilidades que puedan darse, es recomendable antes de ir la practica tener consolidados muy bien los conocimientos teóricos de Modelaje de Procesos y determinación de Reglas de Negocio. Con esto se concluye que para poder implantar BPM es recomendable contar con la suite BPMS.

BPMS construye, transparentemente para el usuario, el código de programación para el motor que conducirá la ejecución del proceso. No es necesario ser programador ni ser técnico de TI, solo conocer de procesos y la notación de la herramienta. BPM (2011).

2.2.8.5 BPMN -Business Process Management Notation

Según, Club BPM (2011) es el nuevo estándar para el modelado de procesos de negocio, usa una notación gráfica a través de la cual se expresan los procesos de negocio en un diagrama de procesos de negocio (BPD), dentro de las características principales de BPMN tenemos:

- ✓ Posee un lenguaje gráfico común, con el fin de facilitar su comprensión a los usuarios de negocios.
- ✓ Integra las funciones empresariales. Utiliza una Arquitectura Orientada por Servicios (SOA), con el objetivo de adaptarse rápidamente a los cambios y oportunidades del negocio.
- ✓ Combina las capacidades del software y la experiencia de negocio para optimizar los procesos y facilitar la innovación del negocio.

Los elementos de los diagramas disponibles para construir los procesos del negocio, se pueden dividir en cuatro categorías básicas de elementos son los representados en el siguiente cuadro:

Tabla N° 11. Categorías básicas de elementos BPMN		
Objetos de flujo	Eventos: Es algo que sucede durante el proceso de negocio, y que afecta el flujo del proceso. Se dividen en: Eventos de inicio. Eventos intermedios. Eventos de fin. Actividad: Es un término genérico para el trabajo que se realiza en una compañía Tarea Subproceso ✓ Gateway (compuerta)	     
Objetos conectores	Conectan los objetos de flujo de un proceso, y definen el orden de ejecución de las actividades	 
Swimlanes (canales)	Son un mecanismo empleado para organizar actividades en categorías separadas visualmente	 
Artefactos	Son objetos gráficos que proveen información adicional de los elementos dentro de un proceso, sin afectar el flujo del proceso	

2.3 Marco Referencial

Esta investigación está enmarcada en las Pequeñas y Medianas Empresas de Venezuela. Lemes & Machado, (2007) indicaron en el Segundo Encuentro Internacional sobre Las Medianas, Pequeñas y Micro-Empresas del Siglo XXI que las pequeñas y medianas empresas (PyMES) son la caracterización más elocuente del tejido empresarial de cualquier país, sea desarrollado o subdesarrollado. Su forma de manifestarse varía en función del país en que se encuentra, pero en esencia su núcleo básico es el mismo y, además, se mueve dentro del marco de ventajas y desventajas asociadas a su propio tamaño. Por ello se clasifican de diferentes formas e incluso se agrupan de acuerdo a distintos indicadores.

2.3.1 Historia de las Pequeñas y Medianas Empresas de Venezuela

Las organizaciones en Venezuela, cualquiera sea su naturaleza, tienen un papel muy importante en el desarrollo socioeconómico del país. Se encuentran por un lado las PYMIS que están referidas al sector manufacturero, tienen su fundamento legal en la Ley para la Promoción Desarrollo de la Pequeña y Mediana Industria (Asamblea Nacional de la República Bolivariana de Venezuela, 2002); mientras que las PYMES se constituyen como pequeñas y medianas empresas comercializadoras, lo que incluye a la industria manufacturera, al comercio y a los servicios.

En la presente investigación se habla de Pymes, ya que incluyen a las PYMIS.

En la Reforma de la Ley para la promoción y desarrollo de la pequeña y mediana industria (PYMIS) y unidades de propiedad social, En Gaceta Oficial Extraordinaria N° 6.151 de fecha 18 de noviembre de 2014, mediante el decreto N° 1.413 fue publicada La Reforma del Decreto con Rango, Valor y Fuerza de Ley para la Promoción y Desarrollo de la Pequeña y Mediana Industria y Unidades de Propiedad Social.

- ✓ En su Artículo 5, define Pequeña y Mediana Industria como toda unidad organizada jurídicamente con la finalidad de desarrollar un modelo económico productivo mediante actividades de transformación de materias primas en

insumos, en bienes industriales elaborados o semielaborados, dirigidas a satisfacer las necesidades de la comunidad. Se considerará Pequeña Industria a aquéllas que tengan una nómina promedio anual de hasta cincuenta (50) trabajadores y con una facturación anual de hasta doscientas mil Unidades Tributarias (200.000 UT). Se considerará Mediana Industria a aquéllas que tengan una nómina promedio anual desde cincuenta y un (51) trabajadores, hasta cien (100) trabajadores y con una facturación anual desde doscientas mil una Unidades Tributarias (200.001 UT) hasta quinientas mil Unidades Tributarias (500.000 UT).

- ✓ Instituto Nacional de Desarrollo de la Pequeña y Mediana Industria (INAPYMI) es un instituto público, que posee patrimonio separado de la República, adscrito al Ministerio del Poder Popular con competencia en materia de industrias, el cual tiene como fin ejecutar las políticas de fomento y financiar los proyectos de inversión de este sector de producción desde la concepción hasta la etapa de comercialización.

2.3.2 El Software de las Pequeñas y Medianas Empresas de Venezuela

Actualmente en Venezuela, existe un número considerable de Pequeñas y Medianas Empresas que han decidido desarrollar sus propias soluciones de software, ya que cuentan con el recurso humano preparado en el área de las Tecnologías de Información, la infraestructura Tecnológica y las herramientas de software para desarrollarlo. La estrategia está dirigida a ahorrar costos, y a empoderarse de los procesos que con facilidad pueden ser modificados por la dinámica del cambiante entorno.

2.4. Bases Legales

- ✓ El Art. 110 de La Constitución de la República Bolivariana de Venezuela (1999) reza “El Estado reconocerá el interés público de la ciencia, la tecnología, el conocimiento, la innovación y sus aplicaciones y los servicios de

información necesarios por ser instrumentos fundamentales para el desarrollo económico, social y político del país, así como para la seguridad y soberanía nacional. Para el fomento y desarrollo de esas actividades, el Estado destinará recursos suficientes y creará el sistema nacional de ciencia y tecnología de acuerdo con la ley. El sector privado deberá aportar recursos para las mismas. El Estado garantizará el cumplimiento de los principios éticos y legales que deben regir las actividades de investigación científica, humanística y tecnológica. La ley determinará los modos y medios para dar cumplimiento a esta garantía”.

- ✓ Artículo 2 del REGLAMENTO INTERNO DEL INSTITUTO NACIONAL DE DESARROLLO DE LA PEQUEÑA Y MEDIANA INDUSTRIA indica que El INAPYMI es una institución cuya creación y funcionamiento auspicia el Estado venezolano para promover el desarrollo de la pequeña mediana industria y las unidades de propiedad social en todos los ámbitos de la República Bolivariana de Venezuela.

CAPITULO III: MARCO METODOLÓGICO

En este capítulo se detalla la metodología que se ha seleccionado para desarrollar la investigación, se explica el “cómo” se hizo la investigación, y “dado que la investigación científica es, en esencia, como cualquier tipo de investigación, sólo que más rigurosa, organizada y se lleva a cabo cuidadosamente” (Hernández, Fernández & Baptista, 2010); es necesario definir el método y técnicas de cómo se llevó a cabo esta investigación. Inicialmente se describe el enfoque seleccionado; seguidamente se define el tipo y el diseño de la investigación, así como las técnicas de recolección de datos, la población y muestra que se utilizaron para construir el Modelo de Integración de Procesos.

La investigación, se le se trabajó bajo un enfoque cuantitativo porque usa la recolección de datos para probar teorías, en base a la medición numérica y el análisis estadístico. El tipo de investigación seleccionado es de campo debido a que se recolectaron datos directamente de la realidad. El diseño es No Experimental porque se observaron los hechos tal y como se presentan en su contexto real. Para esta investigación se ha elegido el nivel Proyectivo porque conducen a inventos, diseños programas o creaciones, basadas en conocimientos anteriores.

3.1 Enfoque de la Investigación

Según Hernández, Fernández & Baptista (2010), el enfoque de la investigación puede ser cualitativo o cuantitativo, y definen el enfoque cuantitativo como un procesos que es secuencial y probatorio, que usa la recolección de datos para probar hipótesis, con base en la medición numérica y el análisis estadístico, para establecer patrones de comportamiento y probar teorías. Esta descripción aplica perfectamente con el enfoque que se le dio a esta investigación, es por ello que ha sido seleccionado. Estos autores enumeran las siguientes características del enfoque cuantitativo:

- ✓ El investigador o investigadora plantea un problema de estudio delimitado y concreto. Sus preguntas de investigación versan sobre cuestiones específicas.

- ✓ Una vez planteado el problema de estudio, el investigador o investigadora considera lo que se ha investigado anteriormente (la revisión de la literatura) y construye un marco teórico (la teoría que habrá de guiar su estudio)
- ✓ La recolección de los datos se fundamenta en la medición (se miden las variables o conceptos contenidos en las hipótesis).
- ✓ Debido a que los datos son producto de mediciones se representan mediante números (cantidades) y se deben analizar a través de métodos estadísticos.
- ✓ La investigación cuantitativa debe ser lo más “objetiva” posible, los fenómenos que se observan y/o miden no deben ser afectados por el investigador.
- ✓ Al final, con los estudios cuantitativos se intenta explicar y predecir los fenómenos investigados, buscando regularidades y relaciones causales entre elementos.
- ✓ La búsqueda cuantitativa ocurre en la realidad externa al individuo.

3.2 Tipo de Investigación

Los autores Palella & Martins (2012), indican que el tipo de investigación se refiere a la clase de estudio que se va a realizar, orienta sobre la finalidad general del estudio y sobre la manera de recoger las informaciones o datos necesarios. Definen y clasifican cuatro tipos de investigación según sigue:

- ✓ **Investigación de campo:** “La Investigación de campo consiste en la recolección de datos directamente de la realidad donde ocurren los hechos, sin manipular o controlar las variables. Estudia los fenómenos sociales en su ambiente natural. El investigador no manipula variables debido a que esto hace perder el ambiente de naturalidad en el cual se manifiesta”. (pag.88)
- ✓ **Investigación documental:** “La investigación documental se concreta exclusivamente en la recopilación de información en diversas fuentes. Indaga sobre un tema en documentos-escritos u orales- uno de, los

ejemplos más típicos de esta investigación son las obras de historia”. (pag.90)

- ✓ **Investigación Pre-Experimental:** “En este tipo de investigación, el grado de control de las variables es mínimo y poco adecuado para el establecimiento de relaciones entre la variable independiente y la dependiente. Es conveniente utilizarla sólo como prueba de experimentos que requieren mayor control. Puede servir en ocasiones como estudio exploratorio, debido a que es útil como un primer acercamiento al problema de investigación.”.(pag.89)
- ✓ **Investigación Cuasi-Experimental:** “se usa cuando no es factible utilizar un diseño experimental verdadero. Es un método de control parcial, basado en la identificación de los factores que pueden intervenir en la validez interna y externa del mismo. Incluye el uso de grupos intactos de sujetos para la realización del experimento, puesto que en un estudio no siempre es posible seleccionar los sujetos al azar”.(pag.89)

Para el desarrollo de este estudio, se utilizó una investigación de campo ya que se hizo una recolección de datos directamente de la realidad donde ocurren los hechos.

3.3 Diseño de la Investigación

Los autores Hernández, Fernández & Baptista (2010), definen el termino diseño como el plan o estrategia concebida para obtener la información que se desea. En el enfoque seleccionado, cuantitativo, el investigador utiliza su diseño para analizar la certeza de las hipótesis formuladas en un contexto en particular o para aportar evidencia respecto de los lineamientos de la investigación. Según los autores Palella & Martins (2012), definen con fines didácticos los siguientes diseños cuantitativos (experimentales y no experimentales):

- ✓ **Diseño experimental:** “El diseño experimental es aquel según el cual el investigador manipula una variable experimental no comprobada, bajo condiciones estrictamente controladas. Su objetivo es describir de qué modo y porque causa se produce o puede producirse un fenómeno. Busca

predecir el futuro, elaborar pronósticos que una vez confirmados, se convierten en leyes y generalizaciones tendentes a incrementar el cúmulo de conocimientos pedagógicos y el mejoramiento de la acción educativa”.
(pag.86)

- ✓ **Diseño no experimental:** “El diseño no experimental es el que se realiza sin manipular en forma deliberada ninguna variable. El investigador no sustituye intencionalmente las variables independientes. Se observan los hechos tal y como se presentan en su contexto real y en un tiempo determinado o no, para luego analizarlos. Por lo tanto en este diseño no se construye una situación específica si no que se observa las que existen”.
(pag.87)

En esta investigación se eligió el Diseño NO Experimental, dado que no se sustituyeron intencionalmente las variables independientes, al contrario se observan los hechos tal y como se presentan en su contexto real.

3.4 Nivel de la Investigación

Según, Palella & Martins (2012), la investigación de campo comprende los niveles Exploratorio, Descriptivo, Explicativo, Evaluativo, Proyectivo, Correlacional, Transversal o transeccional, y Longitudinal. Para esta investigación se ha elegido el nivel Proyectivo, que es descrito por Hurtado de Barrera (2000) como investigaciones que conducen a inventos, diseños programas o creaciones, basadas en conocimientos anteriores. También menciona que este nivel de investigación se ocupa de cómo deberían ser las cosas, para alcanzar unos fines y funcionar adecuadamente, es por ello que es importante la creatividad para poder trascender el presente, para poder concebir el futuro deseado, El investigador puede llegar a ésta mediante vías diferentes, las cuales involucran procesos, enfoques, métodos y técnicas propias. Se elige por contar con las siguientes bondades:

- ✓ Visión Holística que estudia los eventos en su contexto, donde “el todo es más que la suma de las partes”.

- ✓ Relaciones dinámicas: se interesa en los procesos evolutivos y las relaciones dinámicas entre los eventos.
- ✓ Creatividad y participación: toma todos los actores del proceso
- ✓ Actitud hacia el futuro y libertad para transformar los sucesos a partir de acciones voluntarias y dirigidas hacia ciertos fines.

3.5 Población y Muestra

Palella & Martins (2012), definen la población en una investigación cómo el conjunto de unidades de las que se desea obtener información y sobre las que se van a generar conclusiones; y definen la muestra, como la escogencia de una parte representativa de una población, cuyas características reproduce de la manera más exacta posible. Estos autores clasifican el tipo de muestreo en probabilístico y no probabilístico, el primero se aplica si es posible conocer la probabilidad de selección de cada unidad componente de la muestra, el segundo, se usa cuando no se puede determinar la probabilidad. Este último se subdivide en Muestreo accidental, Muestreo intencional y Muestreo por cuotas.

Se seleccionó el tipo de Muestreo Intencional No Probabilístico, ya que el investigador estableció previamente los criterios para seleccionar las unidades de análisis, las cuales reciben el nombre de tipo.

Las Pequeñas y Medianas Empresas de Venezuela, están constituida por una población no cuantificada oficialmente, sin embargo, CONSECOMERCIO indica la existencia de aproximadamente doscientas cincuenta mil empresas. La Muestra seleccionada para esta investigación consta de una empresa del sector Droguerías y otra empresa del sector Tecnología, enfocada específicamente en el estudio de cuatro (4) de los procesos involucrados en el proceso de desarrollo de software: Proceso de levantamiento de Requerimientos, Proceso de Construcción de Software, Proceso de Pruebas de Software y Proceso de Control de Proyectos. La muestra total está compuesta por cuatro (4) procesos de cada una de la empresa de este tipo ubicadas en la ciudad de Caracas.

3.6 Técnicas de Recolección de Datos

Hernández, Fernández & Baptista (2010) indican que recolectar los datos implica elaborar un plan detallado de procedimientos que nos conduzcan a reunir datos con un propósito específico. Para recolectar datos disponemos de una gran variedad de instrumentos o técnicas como observación, entrevista, encuesta, pruebas, entre otras. Advierten además, que en los estudios cuantitativos se busca que el factor humano del investigador se reduzca al mínimo posible. Por lo que la validez, la confiabilidad y la objetividad no deben tratarse de forma separada.

Las herramientas de recolección de datos primarios, definidas por Palella & Martins (2012), son las siguientes:

- ✓ **La observación:** Consiste en el uso sistemático de nuestros sentidos orientados a la captación de la realidad que se estudia. A través de los sentidos, el hombre capta la realidad que lo rodea y luego la organiza intelectualmente, para la actividad científica resulta de primordial valor. En la observación se apoya el investigador para obtener el mayor número de datos. Sin embargo, dado que la presencia del observador puede provocar, por sí sola, una alteración o modificación en la conducta de los observados, se sugiere integrarse a la acción de los observados, participar como un miembro más del grupo. Los pasos son: Determinar el objeto que se va a observar. Concretar el para qué se va a observar. Establecer la forma como se van a registrar los datos. Observar detallada, rigurosa y críticamente. Registrar los datos observados. Analizar e interpretar los datos. Elaborar conclusiones. (pp. 115-116)
- ✓ **La entrevista:** es una técnica que permite obtener datos mediante un diálogo que se realiza entre dos personas cara a cara: el entrevistador "investigador" y el entrevistado; la intención es obtener información que posea este último.
- ✓ **La encuesta:** es una técnica destinada a obtener datos de varias personas cuyas opiniones interesan al investigador. Para ello, a diferencia de la

entrevista, se utiliza un listado de preguntas escritas que se entregan a los sujetos quienes, en forma anónima, las responden por escrito.

- ✓ **Las pruebas tipo ensayo:** pueden ser escritas u orales. Generalmente están construidas con preguntas abiertas y requieren la elaboración de repuestas por parte del investigado. Para estas pruebas se requiere capacidad y destreza cognitiva en aspectos referidos tanto a la argumentación, la organización de ideas, la identificación de hipótesis y de problemas.

Se aplicó en esta Investigación: La Observación, La Encuesta y Las Pruebas tipo Ensayo; adicionalmente una herramienta de recolección de datos secundarios: Internet, ya que se facilita enormemente la tarea de obtener datos secundarios, ya que, permite investigar sobre los temas de interés, hasta dialogar directamente con autores e investigadores, a los cuales sueles pedírsele referencias.

3.7 Técnicas de Análisis de Datos

Una vez finalizadas las tareas de recolección el investigador queda en posesión de un cierto número de datos, a partir de los cuales será posible sacar las conclusiones generales que apunten a esclarecer el problema formulado en los inicios del trabajo. El mismo sugiere que los datos numéricos sean procesados y agrupándolos en intervalos y se tabularán; los datos verbales que se desea presentar como numéricos se codifiquen (agrupar numéricamente); y los datos verbales que habrán de manejarse en forma puramente conceptual (Sabino, 1992) Para el desarrollo de esta investigación se utilizaron algunos de las siguientes técnicas para analizar los datos recopilados a fin de que permitan generar conclusiones.

- ✓ **La tabulación:** La palabra tabulación deriva etimológicamente del latín tabula, y significa hacer tablas, listados de datos que los muestren agrupados y contabilizados. Para ello es preciso contar cada una de las respuestas que aparecen, distribuyéndolas de acuerdo a las categorías o códigos previamente definidos. Cada una de las preguntas de un cuestionario o de una hoja de observación tendrá que ser tabulada independientemente, por lo que es

preciso hacer previamente un plan de tabulación que prepare adecuadamente la tarea a realizar

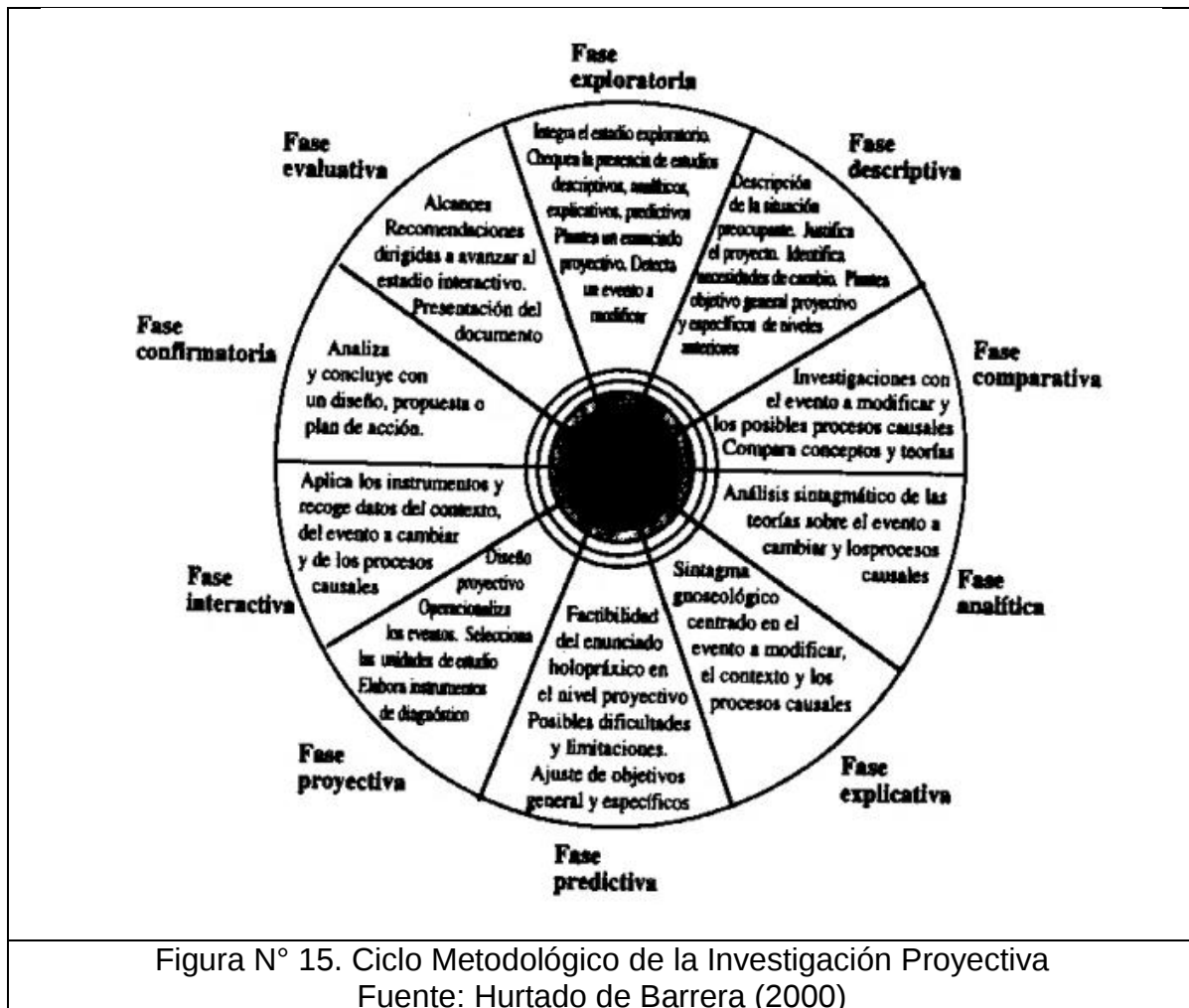
- ✓ **Nivel de medición nominal:** Según Palella & Martins (2012), la distribución de frecuencia es la forma más sencilla para describir una variable. Supone determinar cuántas observaciones están presentes en cada categoría de respuesta para la variable. Las proporciones y los porcentajes, como frecuencias relativas, son extremadamente útiles porque permiten comparar distribuciones de frecuencia de poblaciones de distintos tamaños. Así, se expresan las frecuencias como: $f_1 + f_2 + f_3 + \dots + f_k = N$, donde f_1 se refiere a la frecuencia observada (número de casos u observaciones) de la primera categoría de la variable, f_2 a la frecuencia de la segunda categoría y así sucesivamente hasta llegar a la última (k). La proporción relativa se obtiene ecuación: $P_i = f_i / n$. Se utilizaron histogramas para representar gráficamente el resultado de la medición de las variables.
- ✓ **Matriz DOFA:** son las siglas de Fortalezas, Oportunidades, Debilidades, Amenazas. Ponce (2007) define La matriz FODA, como instrumento viable para realizar análisis organizacional en relación con los factores que determinan el éxito en el cumplimiento de meta. En el contexto de esta investigación se hizo el ejercicio de evaluar las debilidades, fortalezas, oportunidades y amenazas de la unidades involucradas en el proceso de desarrollo, en el función de las respuestas proporcionadas en los instrumentos de medición
- ✓ **Matriz de Evaluación de los Factores Internos (MEFI):** Ponce (2007) sugiere que una vez elaborada la matriz FODA, que enlista los factores internos y externos que influyen en el desempeño de una organización, el siguiente paso es evaluar primeramente la situación interna de la compañía mediante la Matriz de Evaluación de los Factores Internos (MEFI). En el contexto de esta investigación se evalúa la situación interna de la unidades involucradas en el proceso de desarrollo de software.

3.8 Fases de la Investigación

En la investigación proyectiva, la propuesta está basada en un proceso sistemático, que consta de las fases exploratorias, descriptivas, comparativa, analítica, explicativa, predictiva, proyectiva, interactiva, confirmatoria y evaluativa. Hurtado de Barrera (2000). Sin embargo, en esta investigación se utilizaron las siguientes fases:

- ✓ **Fase exploratoria:** La fase exploratoria de un proyecto de investigación consta de varias etapas, a saber:
 - Marco Teórico Conceptual
 - Elaboración del marco teórico
 - Revisión de literatura
 - Adopción de una teoría y desarrollo de una perspectiva teórica.
- ✓ **Fase Diagnóstica:** En esta fase se seleccionó la empresa objeto del estudio, para luego realizar un diagnóstico a través de aplicación de herramientas de recolección de datos e información como son las encuestas y entrevistas. Se tabularon y analizaron los datos de la fase diagnóstica para proceder a analizar los eventos a cambiar y los procesos causales.
- ✓ **Fase Descriptiva:** En esta fase se describieron las actividades que llevan a cabo cada una de las cuatro unidades a fin de poder determinar como llevan a cabo los procesos para finalmente elaborar el mapa de procesos de cada una de ellas, obteniendo resultado de esta fase la descripción general de los procesos objeto de estudio e investigación.
- ✓ **Fase de Diseño:** En esta fase se diseñaron todos los elementos necesarios para construir el modelo: diseño integrado de los procesos, diseño de las actividades del nuevo proceso, diseño del mecanismo de integración. Una vez diseñado todos los elementos, se procedió a construir el Modelo Integrado de Procesos: elaboración de formatos que definen el estándar, construir los indicadores de gestión de cada uno de los procesos involucrados, de cada unidad de trabajo .

- ✓ **Fase evaluativa:** En esta fase, se evaluó el Modelo construido en función de una prueba piloto .
- ✓ **Fase Documental:** En esta fase se procedió a elaborar las conclusiones y recomendaciones de los resultados obtenidos de la fase evaluativa.



3.9 Variables, Definición Conceptual y Operacional e Indicadores

Los autores Palella & Martins (2012), definen las variables como elementos o factores que pueden ser clasificados en una o más categorías, es posible medirlas o cuantificarlas, según sus propiedades o características. las variables pueden adquirir valores tanto cualitativos (bueno, malo, regular, medio, bajo, entre otros) como

cuantitativos (1 ,2, 3 ...). Por la relación que guardan con el propósito de la investigación pueden ser independiente, dependiente. La primera expresa la causa que produce el resultado o efecto observado. La segunda representa el efecto o resultado producido por la variable independiente.

En cuanto a la operacionalización de las variables lo definen como el procedimiento mediante el cual se determinan los indicadores que caracterizan o tipifican a las variables de una investigación. A continuación se muestra la forma de representar la operacionalización de las variables de esta investigación, según estos autores.

Objetivos Específicos	Variable	Dimensión	Indicador
Diagnosticar la situación actual, cómo se llevan a cabo los procesos de Desarrollo de Software en empresas del sector PYMES en Venezuela	Situación actual de Procesos de Desarrollo de Software.	Las cuatro unidades de trabajo de las empresas.	Cantidad de Equipos de trabajos que participan el desarrollo de software. Métricas utilizadas para medir la calidad del trabajo. Mecanismos de Comunicación con las áreas involucradas. Mapas de Procesos. Descripción de Roles y Actividades. Identificación de Entradas y Salidas de cada proceso. Madurez de los procesos actuales.

Objetivos Específicos	Variable	Dimensión	Indicador
Describir los procesos y los equipos de trabajo involucrados en el Desarrollo de Software en empresas del sector PYMES en Venezuela.	Los Procesos y los equipos de trabajo involucrados en el Desarrollo de Software	Las Empresas	Nombres de los cuatro equipos de Trabajo y sus procesos en cada empresa Actividades que realizan los equipo de trabajo dentro del proceso de desarrollo de software. Técnicas y herramientas utilizadas para compartir información con los demás equipos de trabajo Métricas de desempeño del equipo de trabajo
Elaborar un Modelo que integre los Procesos que participan en el Desarrollo de Software para empresas del sector PYMES en Venezuela.	Modelo que integre los Procesos que participan en el Desarrollo de Software.	Las cuatro unidades de trabajo de las empresas.	Indicadores de Gestión de Calidad para los equipos ERF, EDA, ECS y ESP. Formatos para integración de procesos.
		Producto de Software	Nivel de Integración de los procesos Indicadores de Gestión de Calidad del Proceso. Indicadores de Gestión de Calidad del Producto.

Objetivos Específicos	Variable	Dimensión	Indicador
Validar el Modelo propuesto a través de juicio experto.	Modelo Propuesto	Producto de Software	Indicadores de Calidad del Proceso: validar si se construyó siguiendo las mejores prácticas de estándares, metodologías sugeridas. Calidad del Producto: Tiempos de Entrega, Velocidad de Respuesta, Cumple con los requisitos iniciales. Nivel de integración: Agiliza procesos, mejora nivel de comunicación entre las unidades. Arroja las métricas globales del proyecto. Satisfacción del Usuario

3.10 Estructura Desagregada de Trabajo

Según PMI (2013), La EDT/WBS representa todo el trabajo necesario para realizar el producto y el proyecto, e incluye el trabajo de dirección del proyecto. El total del trabajo correspondiente a los niveles inferiores debería corresponder al acumulado para los niveles superiores, de modo que no se omita nada y que no se efectúe ningún trabajo extra. En la figura N° 17 se muestra el EDT/WBS de este proyecto de Investigación.

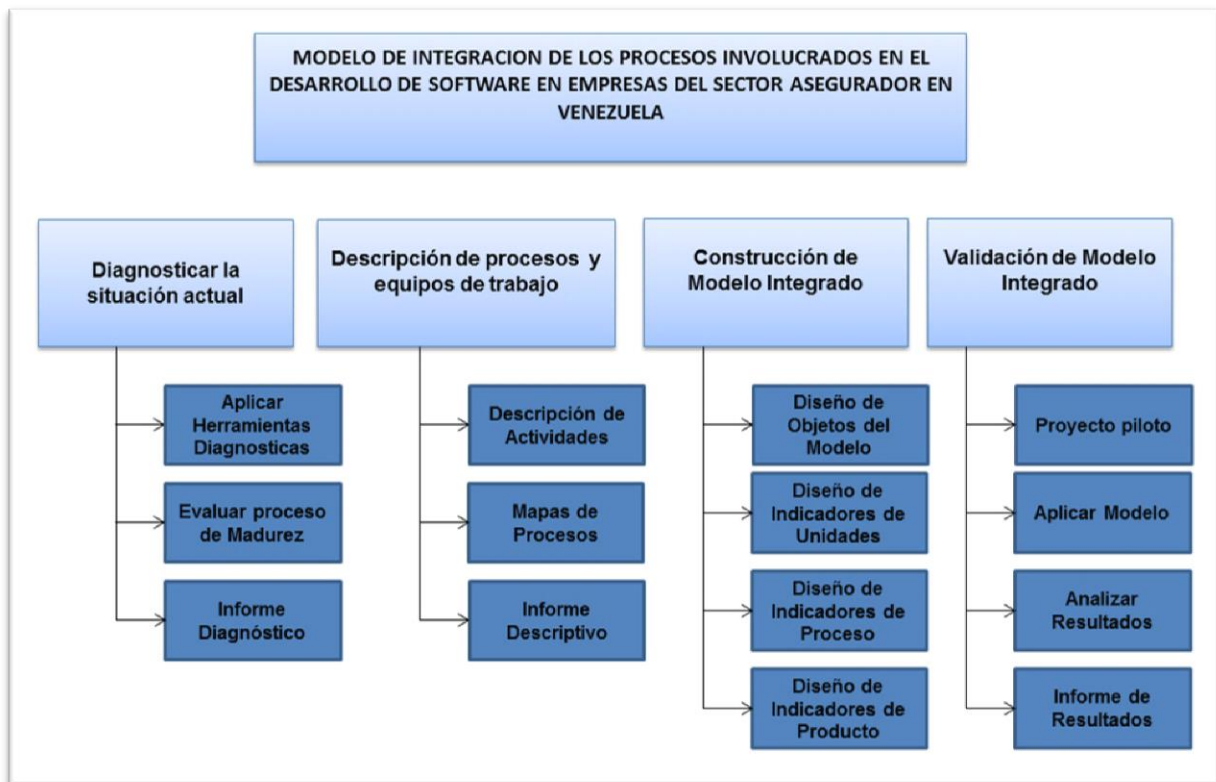


Figura N° 16. EDT Modelo Integración

3.11 Aspectos Éticos

Los códigos de ética tienen una función esencial para caracterizar una profesión, y para que una disciplina adquiera el carácter de profesión debe poseer un código de conducta. El Código de Ética y Práctica Profesional de Ingeniería del Software de la ACM (Association for Computing Machinery) / IEEE Computer Society (1992), en su versión 5.2, establece que contiene ocho Principios clave, relacionados con el comportamiento y las decisiones tomadas por los ingenieros del software profesionales, tanto si son profesionales en ejercicio, educadores, gestores, directivos y responsables, como si se trata de educandos y estudiantes.

Los ingenieros de software debieran obligarse a hacer del análisis, especificación, diseño, desarrollo, pruebas y mantenimiento del software una profesión respetada y

beneficiosa. En el desarrollo de este proyecto se siguió el código de ética que rige a los ingenieros de software según a los Ocho Principios siguientes:

1. Sociedad: Los ingenieros de software actuarán de manera coherente con el interés social.
2. Cliente y Empresario: los ingenieros de software actuarán de manera que produzca el mejor resultado para cliente y empresario, y de manera coherente con el interés social.
3. Producto: los ingenieros de software garantizarán que sus productos y las modificaciones correspondientes cumplen los mayores estándares profesionales posibles.
4. Valoración: los ingenieros de software mantendrán la integridad e independencia en sus valoraciones profesionales.
5. Gestión: los líderes y gestores de ingeniería de software suscribirán y promoverán un enfoque ético en la gestión del desarrollo y mantenimiento del software.
6. Profesión: los ingenieros de software avanzarán en la integridad y reputación de la profesión, de manera consistente con el interés social.
7. Compañeros: los ingenieros del software serán justos y apoyarán a sus compañeros.
8. Personal: los ingenieros del software participarán en el aprendizaje continuo referente a la práctica de su profesión y promoverán un enfoque ético en la práctica de la profesión.

3.12 Cronograma

Se construyó el cronograma de actividades con el software Microsoft Project, con fechas de Inicio y Fin tentativas, según se muestra en la figura N° 17

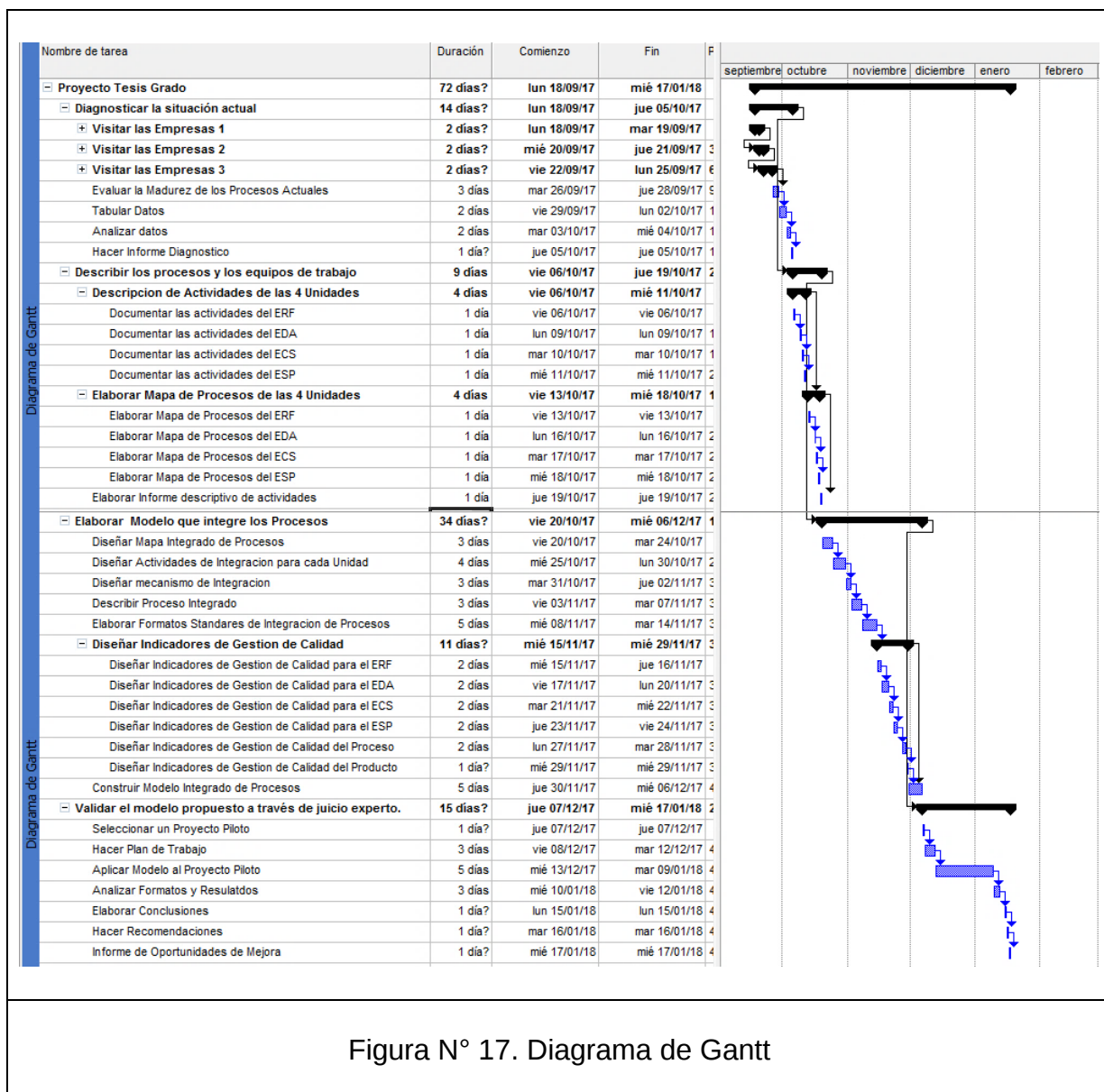


Figura N° 17. Diagrama de Gantt

3.13 Recursos

A continuación se muestra la lista de recurso humanos, materiales e intelectuales que fueron necesarios para llevar a cabo de forma exitosa el desarrollo de este Trabajo de Grado de Maestría.

Tipo de recurso	cantidad	costo del recurso
Humano:		
✓ Horas Hombre	572 horas	
Infraestructura:		
✓ Computador Personal	1	100.000,00
✓ Software Informático	1	0,00
✓ Impresora	1	80.000,00
✓ Oficina	1	0,00
Papelería		
✓ Resmas de Papel	2	120.000,00
Seminarios		
✓ Curso BPMN	1	200.000,00
✓ Curso Calidad	1	160.000,00
TGM		
✓ Encuadernado	3	60.000,00
TOTAL		720.000,00

CAPITULO IV: ANÁLISIS DE DATOS

En este capítulo se detallan las actividades que se llevaron a cabo para obtener los datos y su respectivo análisis, con la finalidad de cumplir con los objetivos planteados en la presente investigación. Se inicia con el Análisis de encuestas y entrevistas.

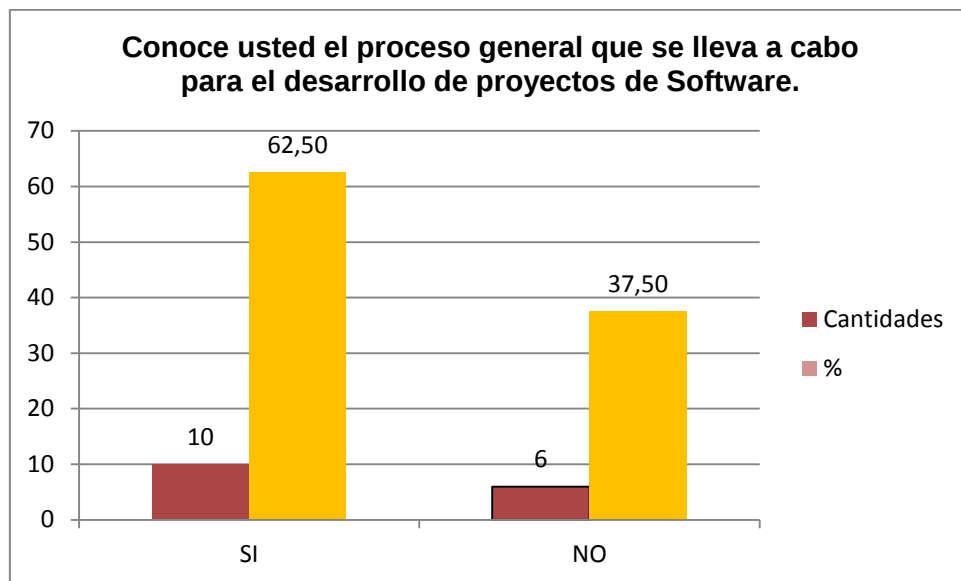
4.1 Análisis de encuestas y entrevistas

Para desarrollar esta sección, se diseñaron siete preguntas orientadas a indagar el nivel de conocimiento de los proceso por parte del recursos humanos que participan en los proyectos de desarrollo de software, a continuación se listan y se posteriormente se analizan:

- ✓ ¿Conoce usted el proceso general que se lleva a cabo para el desarrollo de proyectos de Software?
- ✓ ¿Conoce usted el proceso que debe seguir para cumplir con sus actividades asociadas al desarrollo de proyectos de Software?
- ✓ ¿Ha participado en el desarrollo de proyectos de Software con anterioridad?
- ✓ ¿Al iniciar un Proyecto de desarrollo de proyectos de Software conoce la planificación?
- ✓ ¿Asiste a las reuniones de seguimiento de Proyecto con frecuencia?
- ✓ ¿Participa usted en el diseño de los Requerimientos funcionales?
- ✓ ¿Conoce de la existencia de la documentación de los procesos de desarrollo de Proyectos de Software?

Pregunta Nro. 1: Conoce usted el proceso general que se lleva a cabo para el desarrollo de proyectos de Software.

Respuestas	TOTAL	
	Cantidades	%
SI	10	62,50
NO	6	37,50
TOTAL	16	100

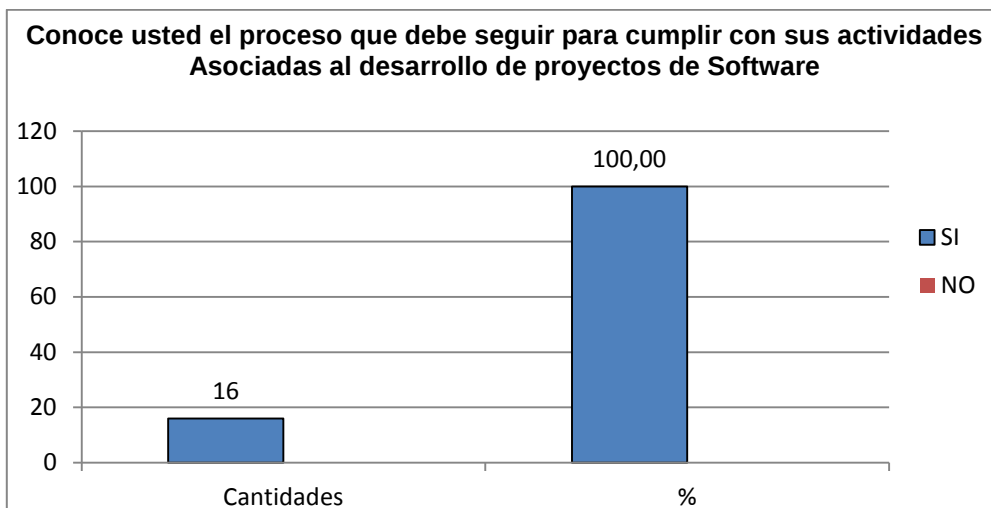


Interpretación: de un total de dieciséis trabajadores encuestados, el 62,5% dice que si conoce el proceso general que se lleva a cabo para el desarrollo de proyectos de Software, mientras que el 37,5% indica desconocerlo.

Análisis: Se puede observar claramente que no todos los encuestados conocen el proceso de desarrollo de Software, a pesar de que trabajan en ellos.

Pregunta Nro. 2: ¿Conoce usted el proceso que debe seguir para cumplir con sus actividades asociadas al desarrollo de proyectos de Software?

Respuestas	TOTAL	
	Cantidades	%
SI	16	100,00
NO	0	0,00
TOTAL	16	100

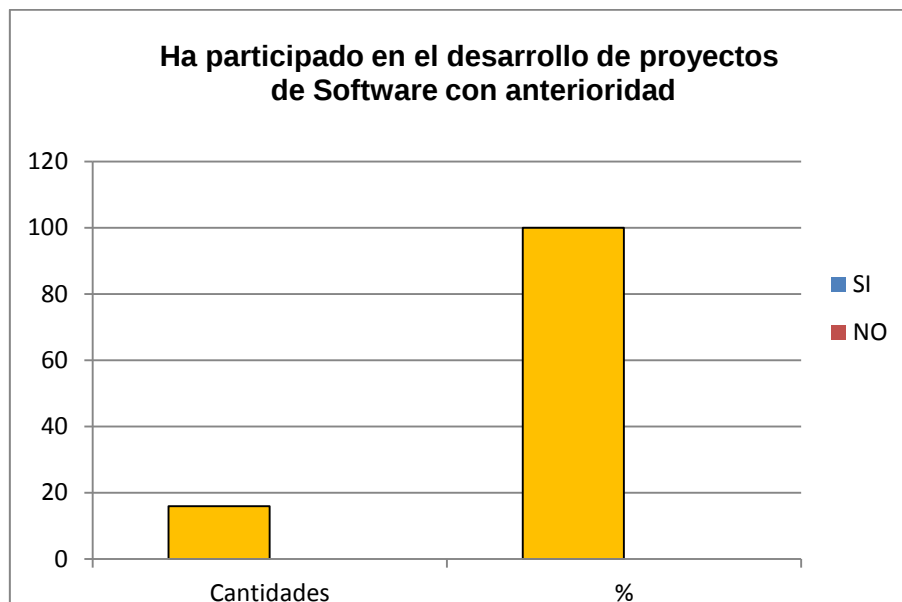


Interpretación: de un total de dieciséis trabajadores encuestados, el 100% dice que si conoce el proceso que debe seguir para cumplir con sus actividades dentro del proyecto.

Análisis: Se puede observar claramente que todos los encuestados conocen el trabajo que deben hacer dentro de un proyecto de desarrollo de Software.

Pregunta Nro. 3: ¿Ha participado en el desarrollo de proyectos de Software con anterioridad?

Respuestas	TOTAL	
	Cantidades	%
SI	16	100,00
NO	0	0,00
TOTAL	16	100

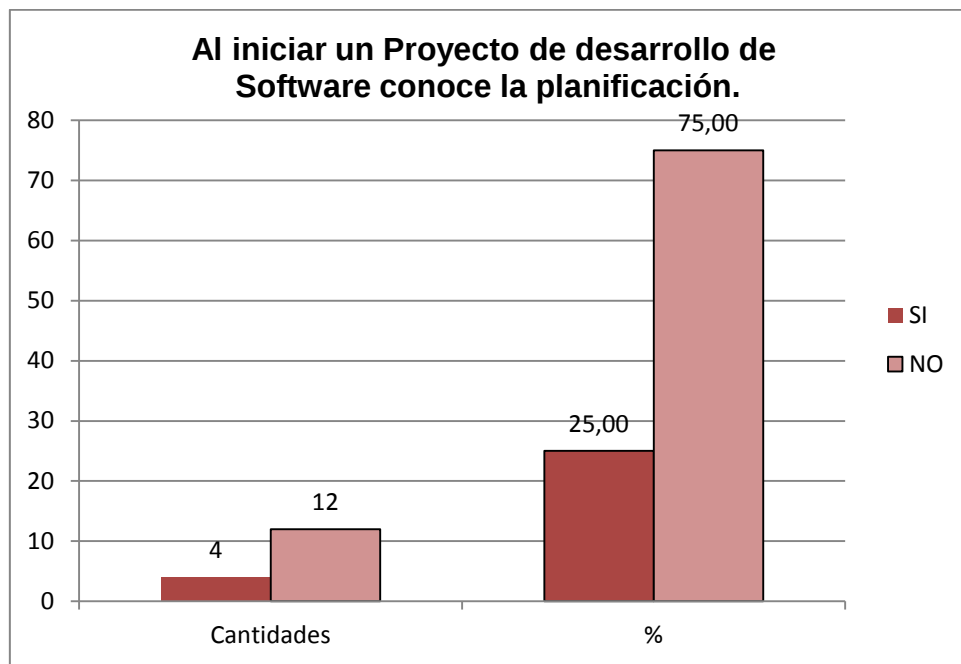


Interpretación: de un total de dieciséis trabajadores encuestados, el 100% dice que si ha participado en proyectos de desarrollo de Software con anterioridad.

Análisis: Se puede observar claramente que todos los encuestados tienen experiencia trabajando en proyectos de desarrollo de Software.

Pregunta Nro. 4: ¿Al iniciar un Proyecto de desarrollo de proyectos de Software conoce la planificación?

Respuestas	TOTAL	
	Cantidades	%
SI	4	25,00
NO	12	75,00
TOTAL	16	100

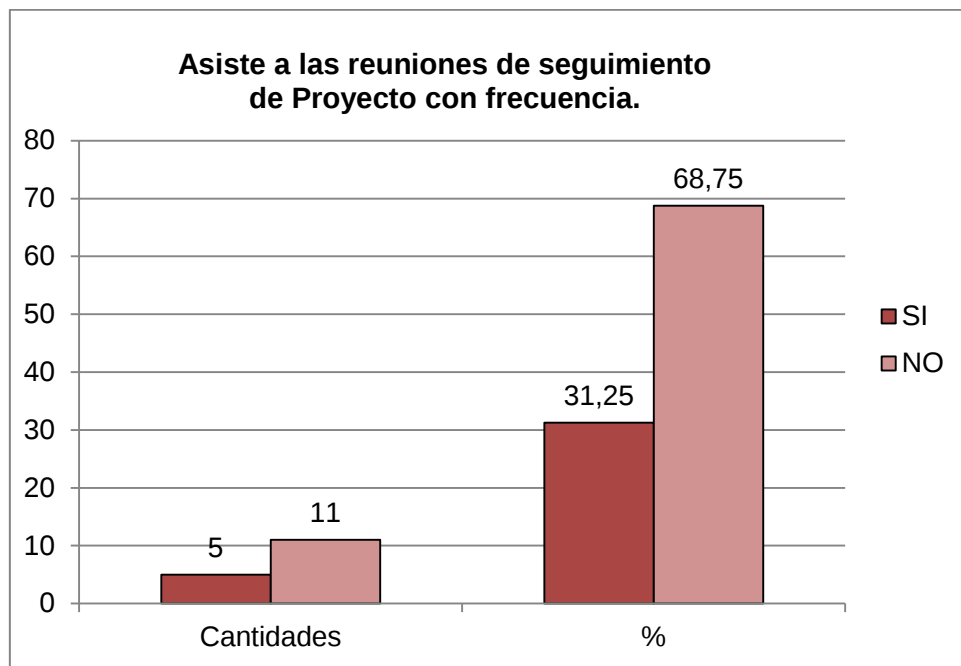


Interpretación: de un total de dieciséis trabajadores encuestados, el 75% dice que NO conoce la planificación inicial del proyecto de desarrollo de Software, mientras que el 25% conocerlo.

Análisis: Se puede observar son muy pocos los trabajadores que conocen la planificación inicial de los proyectos de desarrollo de software.

Pregunta Nro. 5: ¿Asiste a las reuniones de seguimiento de Proyecto con frecuencia?

Respuestas	TOTAL	
	Cantidades	%
SI	5	31,25
NO	11	68,75
TOTAL	16	100

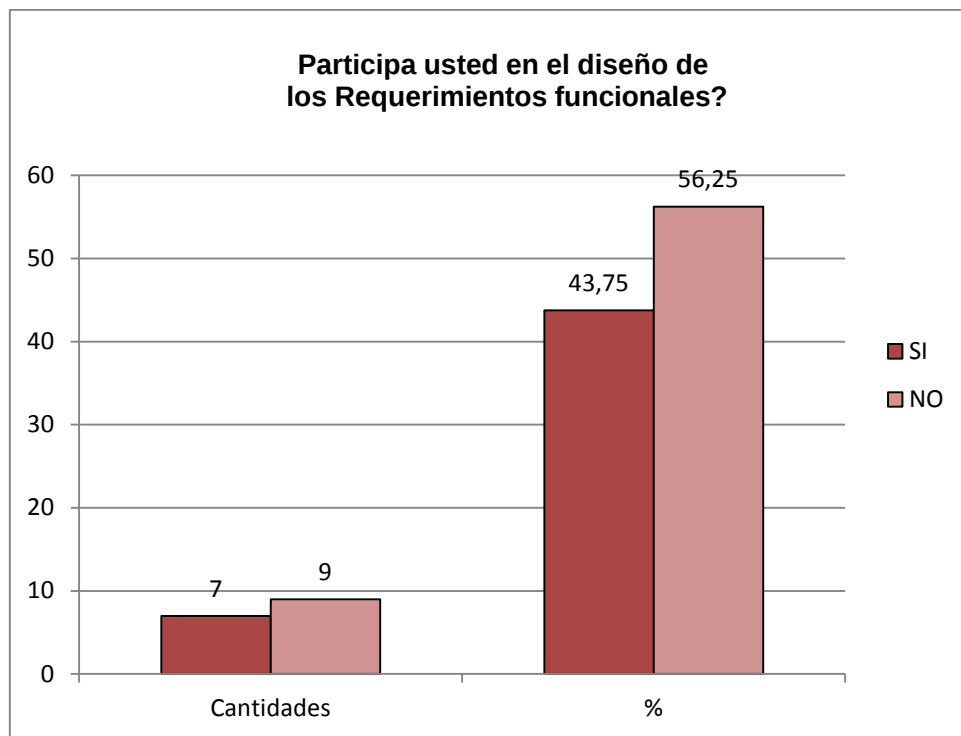


Interpretación: de un total de dieciséis trabajadores encuestados, el 68,75% dice que NO Asiste a las reuniones de seguimiento de Proyecto con frecuencia, mientras que el 31,25% asistir frecuentemente.

Análisis: Se puede observar los trabajadores no están muy interesados en asistir a las reuniones de seguimiento, o puede ser que no sean invitados oportunamente a las reuniones de seguimiento de proyectos de desarrollo de software.

Pregunta Nro. 6: ¿Participa usted en el diseño de los Requerimientos funcionales?

Respuestas	TOTAL	
	Cantidades	%
SI	7	43,75
NO	9	56,25
TOTAL	16	100

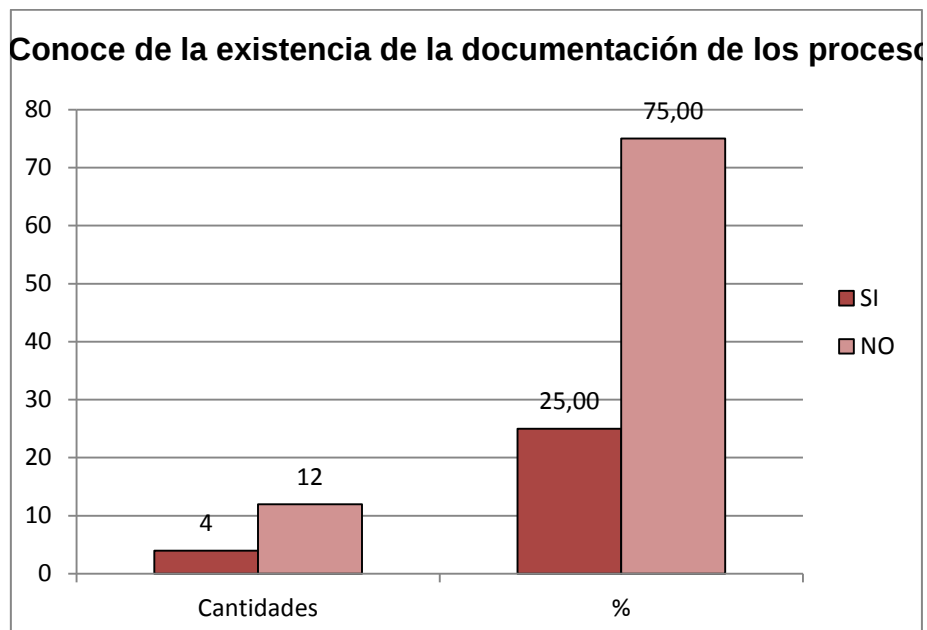


Interpretación: de un total de dieciséis trabajadores encuestados, el 56,25% dice que NO Participa en el diseño de los Requerimientos funcionales, mientras que el 43,75% indica que si participa.

Análisis: Se puede observar que casi la mitad de los trabajadores SI participa en el diseño de los Requerimientos funcionales de proyectos de desarrollo de software.

Pregunta Nro. 7: ¿Conoce de la existencia de la documentación de los procesos de desarrollo de Proyectos de Software?

Respuestas	TOTAL	
	Cantidades	%
SI	4	25,00
NO	12	75,00
TOTAL	16	100



Interpretación: de un total de dieciséis trabajadores encuestados, el 75% dice que desconoce de la existencia de la documentación de los procesos de desarrollo de Proyectos de Software, mientras que el 25% indica que si los conoce.

Análisis: Solo la cuarta parte de los trabajadores encuestados conoce los procesos de desarrollo de proyectos de desarrollo de software, lo cual puede ser desventajoso para los proyectos.

Síntesis General:

De los dieciséis trabajadores encuestados, son pocos trabajadores conocen el proceso general que se lleva a cabo para el desarrollo de proyectos de Software, pocos los que conocen la planificación inicial de los proyectos y no están asistiendo con frecuencia a las reuniones de seguimiento, Sin embargo, no cabe duda de que todos los encuestados tienen experiencia trabajando en proyectos de Desarrollo de Software y conocen el trabajo que deben hacer dentro de un proyecto.

Muchos de estos trabajadores SI participa en el diseño de los Requerimientos funcionales de proyectos de desarrollo de software.

Resultados de la Entrevistas

Se realizaron entrevistas al personal que conforma a Unidad de Desarrollo de Software, Unidad de Calidad de Software, Unidad de Requerimientos Funcionales y Unidad de Seguimiento de Proyectos; con el objetivo de

4.1.1 Resultados de la Entrevistas - Unidad de Desarrollo de Software

En la entrevista aplicada a los Integrantes de los diversos equipos de trabajo, se realizaron las siguientes preguntas las analizadas, y se muestra a continuación el análisis de cada una de ellas:

- a) Explique brevemente como llega a sus manos un nuevo Proyecto de Desarrollo de Software

La mayoría de los desarrolladores de software indican que llega un nuevo proyecto de desarrollo de software a través del supervisor inmediato, los cuales pueden cumplir el rol de Gerente de proyectos o Líderes de proyecto. Otros indican que debido a una necesidad que presenta el usuario o por algún fallo que se presenta en el sistema actual. También puede ocurrir por medio documento que tienen las especificaciones del requerimiento. Los Gerentes de Proyecto indican que es a partir de una necesidad de automatizar un proceso, porque así el usuario lo ha solicitado.

- b) ¿Participa usted en el proceso de levantamiento de Información? ¿de qué manera?

Con las repuestas obtenidas se identifica claramente que hay levantamientos de información para las incidencias del día a día y levantamiento de información para un proyecto. Los desarrolladores de software indican que en las primeras participan normalmente porque es parte de sus actividades diarias, pero no participan en el levantamiento de información para nuevos proyectos. En este caso los líderes de proyectos y gerentes son los que se encargan de ese proceso.

Algunos desarrolladores indicaron que una vez iniciado el proyecto participan esporádicamente, cuando existen dudas de algún aspecto omitido en las especificaciones recibidas.

- c) ¿Le he de utilidad el documento de Especificaciones para el desarrollo de nuevos de proyectos de Software? ¿Qué mejora le haría?

La mayoría de los integrantes del equipo de desarrolladores de software indican que el documento de Especificaciones si les he de utilidad, opinan que es útil porque quedan documentados todos los requerimientos y esto contribuye a que no se alteren durante el desarrollo de las soluciones; sin embargo indican que no abarca todo el detalle requerido para el desarrollo, siempre quedan dudas luego de leerlos, sugieren entonces mejoras en el mismo con más especificaciones y nivel de detalle. A muchos desarrolladores le gustaría formar parte del desarrollo del documento de Especificaciones.

- d) ¿Una vez iniciado el Proyecto está usted informado de la fecha de finalización o entrega del mismo?

La respuesta obtenida de esta pregunta fue casi unánime, todos están informados de la fecha de finalización del proyecto, Hay fechas tentativas por lo que manejan términos como fecha propuesta y fecha definitiva para la realización del mismo. Fechas que se van discutiendo entre las diferentes unidades de trabajo.

- e) ¿Qué nivel de integración tiene su área con la unidad de Levantamiento de Requerimientos? ¿Explique?

En esta pregunta se obtuvo respuestas más diversas, y eso tiene que ver precisamente con el rol del trabajador, se concluye que los líderes de proyectos y gerentes tienen una mayor integración con el área de levantamiento de requerimientos.

- f) ¿Qué nivel de integración tiene su área con la unidad de Aseguramiento de Calidad? ¿Explique?

La mayoría de las respuestas obtenidas de esta pregunta indican que el nivel de integración es bajo, y la comunicación con el equipo de calidad se da mayormente cuando ya los artefactos de software son pasados al ambiente de Pruebas, es decir cuando el equipo de desarrollo ya culminó con su trabajo.

g) ¿Qué nivel de integración tiene su área con la unidad de Seguimiento de Proyectos? ¿Explique?

La mayoría de los encuestados, indica que si hay integración con la unidad de Seguimiento de Proyectos, ya que siempre es necesario dar avances sobre el desarrollo del proyecto, para saber si se está trabajando en los tiempos óptimos o si es necesario levantar alertas por algún retraso en el desarrollo. Muy pocos indican, especialmente los desarrolladores, que solo se preocupan cuando hay presión en los tiempos de entrega y/o usuario solicitante.

h) ¿De qué manera se entera usted de los avances en el proyecto?

La mayoría de los encuestados indican que se enteran a través del supervisor directo. Quienes regularmente asisten a las reuniones de seguimiento semanal son los supervisores. El equipo de Proyectos suele enviar un correo luego de celebrada la reunión, indicando porcentaje de avance del proyecto y puntos tratados, algunos indican que este correo no es muy explícito.

i) ¿Presenta algún tipo de dificultad al momento de Realizar su trabajo?

La mayor parte de los encuestados indica que no presentan dificultad al momento de realizar su trabajo, solo cuando viene la fase de pruebas funcionales, que se presentan retrasos.

j) ¿Cumple usted con los tiempos de entrega según el plan inicial? ¿explique?

Muchas veces es necesario cambiar las fechas de entregas, ya que a medida que se realiza el desarrollo se pueden ir encontrando problemas que pueden cambiar los tiempos, debido a la misma falta de especificaciones detalladas. Este desplazamiento de fechas de entrega se pudiese corregirse si se incluyeran más

recursos al proyecto. Los posibles retrasos se originan por el desvío en los tiempos de pruebas o por retrasos en las aclaratorias requeridas. Son las opiniones de los encuestados.

4.1.2 Resultados de las Entrevistas - Unidad de Calidad de Software

a) ¿En qué momento inician las pruebas de una nueva Aplicación?

Inicialmente hay una fecha tentativa de inicio de las pruebas, luego esta fecha puede verse afectada por varias razones: porque no todos los componentes del desarrollo del proyecto están completos, porque no se ha culminado con las pruebas de un proyecto que está en plena fase de pruebas o porque cambió la prioridad del proyecto con respecto a otro, en términos generales es variable la fecha efectiva de inicio del proyecto.

b) ¿Participa Ud. en la aplicación de calidad a otras áreas cómo el diseño del documento de especificaciones funcionales?

No se aplica calidad al diseño del documento de especificaciones funcionales, ni a otras áreas. Fue la respuesta unísona de los encuestados.

c) ¿Participa en la calidad del proceso de desarrollo de Software?

Todos los encuestados indicaron al unísono que solo se aplica calidad al producto final, no a los procesos intermedios.

d) ¿Cuál es el nivel de errores encontrados en la fase de pruebas? ¿A qué se debe? Existen muchos errores que existían previos al proyecto actual y se detectan en la fase de pruebas. Estos errores se miden por cantidad de defectos, y suelen ser de diversos tipos, errores de validación de datos, errores de pruebas de un proceso completo que involucra varias fases.

e) ¿Qué estándares de calidad son aplicados al software desarrollado?

Las pruebas se realizan según el documento de especificaciones funcionales, es decir, se prueba lo que está escrito en dicho documento, excepto cuando en el transcurso del desarrollo se deja sentado en algún correo que hubo una modificación en el diseño. No hay un estándar internacional que se aplique en este sentido.

- f) ¿El desarrollo entregado por el equipo de desarrolladores, cumple con los estándares de calidad exigidos? ¿Explique?

Por lo general el producto final del desarrollo si cumple con un gran porcentaje de lo que se pide en el documento de requerimientos, pero siempre existen detalles que modificar. Puede ser motivado a la falta de especificaciones detalladas en dicho documento, entonces el desarrollador lo hace en función de lo que entiende literalmente y como en muchas ocasiones desconoce el proceso general donde será probada la solución no ejecuta todas las pruebas unitarias respectivas. Sería bueno hacer un mapa de procesos donde se visualice cómo engrana todos los artefactos desarrollados, argumentan los entrevistados.

- g) ¿Conoce perfectamente los procesos que usted prueba?

Para llevar a cabo las pruebas de los procesos siempre contamos con el apoyo de un usuario funcional experto en el proceso.

- h) ¿Qué metodología usa para hacer su trabajo de aseguramiento de calidad?

Con una herramienta informática que se unas para cargar todos los datos de los requerimiento que están en el documento de especificaciones funcionales, renglón por renglón y se van cambiando de estatus en a medida que se van realizando las pruebas, en las reuniones de avance se notifican los requerimientos con defectos, y al finalizar el proyecto todos los requerimientos deben tener estatus cerrado.

- i) ¿Cumple usted con los tiempos de entrega ?¿explique?

Los tiempos de entrega dependen de la cantidad de defectos que generen, y del tiempo de respuesta del equipo de desarrollo de software en solventarlos. A veces dependemos de aclaratorias del proceso por parte de usuario experto.

4.1.3 Resultados de las Entrevistas - Unidad de Requerimientos Funcionales

¿En qué momento inicia el levantamiento de Información?

Una vez que el equipo de Proyectos determina que debe darse inicio al levantamiento de información, en mutuo acuerdo con los solicitantes, se planifican las reuniones para iniciar con el proceso.

a) ¿De dónde se obtiene la información que alimenta el documento de especificaciones funcionales?

Para hacer las reuniones de trabajo deben estar presentes el personal de informática y el usuario. El principal insumo proviene de los usuarios solicitantes del proyecto quienes detallan verbalmente su solicitud; el personal de informática explica si técnicamente es factible llevar a cabo lo que se solicita, de ser así, explica técnicamente los procesos actuales si es necesario. Toda la información se procesa y se escribe en el documento.

b) 3. ¿Utiliza Ud. Algún tipo de formato para entrevistar a quienes le suministran información?

No existe un formato de entrevistas, son conversaciones donde se van tomando notas a lápiz.

c) ¿Cuáles Son las preguntas Claves de cualquier entrevista con los usuarios funcionales?

Depende del tipo de proyecto. No existen preguntas claves, se inicia la conversación y se va tomando nota, luego se envía por correo.

d) ¿Utilizan alguna Norma o Estándar para diseñar o evaluar el trabajo realizado?

Todos los entrevistados concuerdan en que actualmente NO hay estándares de calidad establecidos.

e) ¿Cuál es el tiempo estimado para hacer un documento de especificaciones?

Las normas internas exigen que no debe tardarse más de dos meses en la elaboración de un documento, expresan los entrevistados.

f) ¿Utilizan algún formato estándar para elaborar el documento de Especificaciones?

A esta pregunta se responde al unísono, todos dicen que si, una sección para describir las premisas, una sección para describir las nuevas pantallas donde se especifican tipo de datos.

g) ¿Qué dificultades presenta al momento de Realizar su trabajo?

Los entrevistados indican que, la coordinación de las reuniones con los usuarios a veces es difícil, porque deben dejar sus obligaciones diarias por lo menos un par de horas en las primeras fases del proceso. Cuando se envían los documentos para ser revisados, por lo general las respuestas son tardías.

h) ¿En cuanto a la forma como realiza su trabajo, qué mejoraría en el proceso actual?

Crear un sistema de documentación en línea o algún mecanismo que permita que el trabajo sea más rápido, argumentan los encuestados.

4.1.4 Resultados de las Entrevistas - Unidad de Seguimiento de Proyectos

a) ¿Describa brevemente como llega a iniciarse un nuevo Proyecto de Desarrollo de Software?

El Departamento cuenta con un portafolio de proyectos que son evaluados por la alta gerencia y de allí se decide la prioridad de inicio de los proyectos, indican los encuestados.

b) ¿Aparte de proyectos de Desarrollo de Software que otros tipos de proyectos se manejan en su unidad?

Todos los proyectos de tecnología son administrados por el equipo de Seguimiento de Proyectos.

- c) ¿Qué herramientas utiliza para hacer seguimiento de Proyectos de Desarrollo de Software?

Microsoft Project, se usa como herramienta corporativa.

- d) ¿Aparte de proyectos de Desarrollo de Software que otros tipos de proyectos se manejan en su unidad?

Todos los proyectos de la Gerencia de Informática.

- e) ¿Cómo son manejadas las desviaciones de tiempo en los proyectos de Desarrollo de Software?

Se celebran reuniones con todas para partes involucradas y se llegan a acuerdos de entregas reales.

- f) ¿Qué nivel de integración con la unidad de Levantamiento de Requerimientos? ¿Explique?

La unidad de Seguimiento de proyectos no participa en el levantamiento de Requerimiento, manifiestan los encuestados.

- g) ¿Qué nivel de integración con la unidad de Aseguramiento de Calidad? ¿Explique?

La unidad de Seguimiento de proyectos solicita las estadísticas de avance de las pruebas: cantidad de requerimientos cerrados, cantidad de requerimientos con defectos, porcentaje de avances de las pruebas de calidad, manifiestan los encuestados.

- h) ¿Qué nivel de integración con la unidad de Desarrollo de Software? ¿Explique?

La unidad de Seguimiento de proyectos solicita porcentaje de avance semanalmente del desarrollo de los artefactos (pantallas, procedimientos de bases de datos, etc.), si hay un defecto pendiente por entregar, hace seguimiento diario, manifiestan los encuestados.

- i) ¿Qué tipo de Normas ó Estándares utiliza para diseñar o evaluar el trabajo realizado?

Indican los encuestados, que tratan de seguir los lineamientos del PMBOOK, pero que no está establecido un estándar corporativo.

- j) ¿Utilizan algún sistema o herramienta les permita integrar las actividades donde todos los involucrados en el proyecto de Desarrollo de Software?

Indican los encuestados, que no existe un sistema integrado para este propósito.

- k) ¿Presenta algún tipo de dificultad al momento de Realizar su trabajo?

Sólo las cotidianas, las del día a día, indican los encuestados.

- l) 12.¿En cuanto a la forma como realiza su trabajo, qué mejoraría en el proceso actual?

Muchas cosas indican los encuestados, hacen énfasis en las especificaciones de requerimientos, y en la documentación de procesos del negocio.

- m) 13.¿Los proyectos de Desarrollo de Software se ejecutan en el tiempo establecido ?¿explique?

Por lo general siempre se producen retrasos de diversa índole: muchos defectos por arreglar, falta de más recursos humanos para atacar esos defectos, si algún desarrollador se enferma puede afectar el avance del proyecto, indican los encuestados.

4.3 Matriz FODA de Procesos

Fortalezas	Oportunidades
La empresa en estudio cuenta con un personal profesional con muchos años de servicios en el área de Informática, especialmente jefes y gerentes, que conocen de los procesos y del negocio.	Se ve claramente la oportunidad de documentar todo el conocimiento que tienen estos trabajadores en un repositorio de información, que sea de fácil acceso para todo el personal.
Cuenta con una Plataforma Tecnológica Robusta.	Existe el recurso tecnológico, espacio y físico donde crear un repositorio centralizado donde todos los miembros del proyecto compartan información.
Las empresas que tienen muchos años de servicio en el sector.	Certificar los procesos informáticos y ganar más prestigio, sería un objetivo que agregaría valor a sus productos, así como también establecería una normativa para todos los participantes de un proyecto.
Muchos de los recursos han participado en proyectos de desarrollo de software dentro y fuera de la empresa.	La empresa debe aprovechar al máximo las competencias de los recursos para invertirlos en el desarrollo de nuevos proyectos.
La empresa es propietaria del Código fuente del CORE del negocio.	Les permite desarrollar cualquier aplicativo nuevo o cualquier funcionalidad adicional que complemente los procesos actuales.
Debilidades	Amenazas
Muy pocos conocen de la existencia de la documentación de los procesos de desarrollo de Proyectos de Software.	Pérdida de tiempo en realizar actividades que no correspondan con la secuencia o el flujo normal del proceso.
Más de la cuarta parte de los encuestados, no conoce el proceso general que se lleva a cabo para el desarrollo de proyectos de Software.	Uso indebido del tiempo de los recursos en explicar procesos durante el tiempo del proyecto
Las tres cuartas partes de los encuestados desconoce la planificación	No se tiene una visión global del objetivo

al iniciar un Proyecto de desarrollo de proyectos de Software.	general y específico del proyecto
Pocos asistentes a las reuniones de seguimiento de Proyecto con frecuencia.	Los inasistentes no tienen la oportunidad de dar su opinión y sugerencias con respecto a los puntos tratados.
Cuando hay un nuevo ingreso en el área de informática no le dan inducción de los procesos de sistemas.	Genera errores al momento de seguir los procesos incurriendo en el retrabajo, afectando otras áreas
Una vez ya en el cargo, las personas no reciben adiestramiento de los procesos internos del área de informática	El personal nuevo aprende por ensayo y error, puede rezagar los tiempos de entrega.
En los nuevos Proyectos no existe un plan maestro	Falta de coordinación de todas las partes involucradas
Las reuniones que celebran en el proceso de levantamiento de información no quedan grabadas.	Es susceptible que si algún miembro del grupo se retira de la empresa, se pierde el conocimiento expresado en esas sesiones.
Las reuniones que se celebran durante la etapa de seguimiento del proyecto no son registradas al momento, la minuta se envía varios días después	En ese tiempo los participantes pueden perder la secuencia de lo que se dijo en la reunión.
La mayoría de los integrantes del proyecto, No asiste a las reuniones de seguimiento de Proyecto con frecuencia	No se puede tomar decisiones oportunas cuando se está desarrollando un aplicativo, afectando el tiempo del proyecto
El personal que no se encuentra físicamente en el edificio donde se celebran las reuniones debe trasladarse físicamente	Pérdida de tiempo en los viajes que se realizan para poder llegar a la reunión afectando el tiempo del proyecto
En la fase de diseño del documento de especificaciones funcionales, se envían múltiples versiones del documento antes de llegar al documento final.	Se pierde la continuidad del documento Inicial, los involucrados en la revisión del documento, pierden interés en realizar toda la lectura del documento nuevo, llenan el buzón de correo rápidamente.

4.4 Acciones a Tomar

Si existe la documentación, crear un mecanismo de divulgación, de no existir se debe generar esta documentación y divulgarla.
Crear un mecanismo para que todos los participantes de un proyecto de desarrollo de software conozcan en su totalidad el proceso
Crear un plan inicial que esboce gráficamente el proyecto y describa el plan maestro.
Grabar las Reuniones y almacenarlas en un directorio público donde en cualquier oportunidad puedan visualizarlo aquellas personas que no asistieron.
Informar a todo el personal de los planes de acción y objetivo final del proyecto, mediante diseños visuales
Documentar procesos para evitar que si algún miembro del grupo se retira de la empresa, no se pierda el conocimiento.
Implementar videoconferencias para evitar pérdida de tiempo en los viajes que se realizan para poder llegar a la reunión.

4.5 Informe descriptivo de los Procesos

4.5.1 EDA: Equipo de Desarrollo de Aplicaciones

El Equipo de Desarrollo de Aplicaciones (EDA) es una unidad encargada de dar mantenimiento y soporte técnico y funcional a las aplicaciones existentes en ambiente de producción, así como también se encarga de desarrollar los nuevos proyectos que le son asignados. Dentro de las actividades que ejecutan cotidianamente podemos enumerar:

- a) Atiende llamadas telefónicas de los usuarios de los aplicativos, y determina el motivo de la llamada, si es una falla que está reportando o si es un nuevo requerimiento.
- b) Si es una falla, por lo general viene acompañada de un correo electrónico que muestra la captura de la pantalla, realiza el análisis del problema o necesidad por parte del área solicitante.

- c) Hace el levantamiento de Información necesario para resolver las fallas e incidencia.
- d) Solicita al área de seguridad de la información los permisos necesarios para acceder a las bases de datos
- e) Si el problema es de datos o relacionado a algún objeto de base de datos, construye los scripts correspondientes destinados a resolver la falla.
- f) Si el problema es a nivel de interfaz gráfica (pantallas, reportes, etc.) modifica los objetos.
- g) En cualquiera de los casos diseña y construye la solución al problema
- h) Cumple con el procedimiento para actualización de los objetos del aplicativo en los ambientes de pruebas (test) y producción, el cual consiste en Actualización del servidor de versiones, generar la documentación del análisis, pruebas realizadas y notificación realizada al solicitante.
- i) En el caso de nuevos proyectos, recibe documento con las especificaciones funcionales y estima el tiempo que se llevará el desarrollo de la programación de las soluciones.
- j) En el proceso de pruebas unitarias, diseña las pruebas e identifica el conjunto de datos de prueba. Identificar casos de prueba con entradas y resultados definidos.
- k) En la fase de pruebas de los artefactos desarrollados para los nuevos proyectos, hace seguimiento de pruebas en ambiente de pruebas "Test".
- l) Organiza el protocolo de seguridad para migración de los programas al ambiente Producción
- m) Notifica vía electrónica la puesta en producción y seguimiento del correcto funcionamiento del aplicativo migrado.

- n) Generación de datos para análisis específicos del área de negocio.
- o) Los roles supervisores:
 - a. Participan en las reuniones de proyectos y/o requerimientos para la definición de alcances y requerimientos a manejar dentro del proyecto
 - b. Verifican el progreso de las actividades asignadas al equipo
 - c. Aclaran las dudas técnicas de los compañeros y proponer soluciones a sus inquietudes
- p) Asiste a reuniones del Comité de controles de cambios para informar ante dicho comité cual es el impacto del desarrollo que se pasará a producción.
- q) Se solicita al usuario funcional realizar pruebas en el ambiente de TEST y se detalla el resultado esperado en las pruebas.

4.5.2 ECS: Equipo de Calidad de Software

El Equipo de Calidad de Software (ECS) es una unidad encargada de probar y certificar junto al usuario funcional los artefactos de software en ambiente de pruebas (test) desarrollados por la unidad de desarrollo de aplicaciones, para cumplir con este objetivo, lleva a cabo las siguientes actividades:

- a) Solicita a la unidad de desarrollo de aplicaciones el inventario de objetos que están involucrados en las pruebas, bajo un formato preestablecido.
- b) Carga en la herramienta de control de requerimientos todas las premisas que se indican en el documento de especificaciones funcionales.
- c) Estima el tiempo total que se consumirá el periodo de las pruebas
- d) Coordina con los usuarios funcionales las fechas que tienen disponibles para probar.
- e) Asiste a las reuniones de seguimiento para reportar los defectos encontrados.

- f) Hace seguimiento de los defectos reportados
- g) Emite estadísticas de casos cerrados, en progreso y con defectos.

4.5.3 ERF: Equipo de Requerimientos Funcionales

El Equipo de Requerimientos Funcionales (ERF) es una unidad encargada de documentar los requerimientos solicitados por los usuarios, sólo documentan los nuevos proyectos, los requerimientos que no forman parte de un proyecto, no son documentados por esta área. Para llevar a cabo esta documentación realizan las siguientes actividades:

- a) Coordina las fechas de las Reuniones con Usuarios solicitantes de los nuevos proyectos.
- b) Levanta Información de los nuevos procesos solicitados por los usuarios.
- c) Informa a los asistentes vía correo, de las anotaciones que tomó en la reuniones para pedir su aprobación.
- d) Elabora el documento final de especificaciones funcionales, lo distribuye y busca las firmas necesaria para su certificación.
- e) Apoya a los desarrolladores de software a aclarar las dudas que se le presentan al momento de la lectura e interpretación del documento de especificaciones funcionales. También puede aclarar dudas al equipo de calidad en el proceso de Pruebas.
- f) Redacta un documento anexo al principal, si el equipo de Desarrollo de Aplicaciones detecta alguna inconsistencia o si se omitió algo.
- g) Asiste a las reuniones de seguimiento de proyectos cuando hay problemas de interpretación del documento.
- h) Si hay cambios de alcance del proyecto inicial, redacta un nuevo documento con las nuevas especificaciones.

4.5.4 ESP: Equipo de Seguimiento de Proyectos

El Equipo de Seguimiento de Proyectos (ESP) es una unidad encargada de administrar el portafolio de proyectos informáticos de la empresa, y, hacer seguimiento a los proyectos desde que nacen hasta que estos son culminados. Realizan las siguientes actividades:

- a) Mantiene Actualizado el portafolio de proyectos
- b) Convoca reuniones de seguimiento con las unidades de desarrollo de aplicaciones y el equipo de calidad de software, para discutir las cantidad de casos cerrados y con defectos,
- c) Actualiza el Microsoft Project y el estatus del proyecto.
- d) Se comunica con la unidad de desarrollo de aplicaciones (UDA) para hacer seguimiento a los compromisos adquiridos en las reuniones
- e) Se comunica con el equipo de calidad de software para validar la información suministrada por la (UDA).
- f) Canaliza con los proveedores externos (software Factory) el desarrollo de nuevas funcionalidades WEB.
- g) Hace seguimiento a otros tipos de proyectos diferentes al área de desarrollo de software, lo cuales no se mencionaran en este apartado.

5 CAPÍTULO V: RESULTADOS

Este capítulo está dividido en cinco partes: Mapa Conceptual del Modelo propuesto, Modelo Propuesto con BPMN 2.0, Formatos del Modelo Propuestos, Diseño de Indicadores de Gestión y Validación del Modelo según el estándar ISO 9126; cada sección será explicada detalladamente por separado para su mejor comprensión.

5.1 Mapa Conceptual del Modelo Propuesto

El diseño del mapa conceptual del Modelo propuesto, parte de un diseño General donde se visualiza cual es el objetivo general de la integración, y luego se desglosa el diseño de cada una de las cuatro unidades que los conforman (URF,USP,UDA,UCS).

5.1.1 Mapa de Proceso General

En la Figura Nro. 18 se representa gráficamente la propuesta del Mapa de Proceso General del Modelo Propuesto.

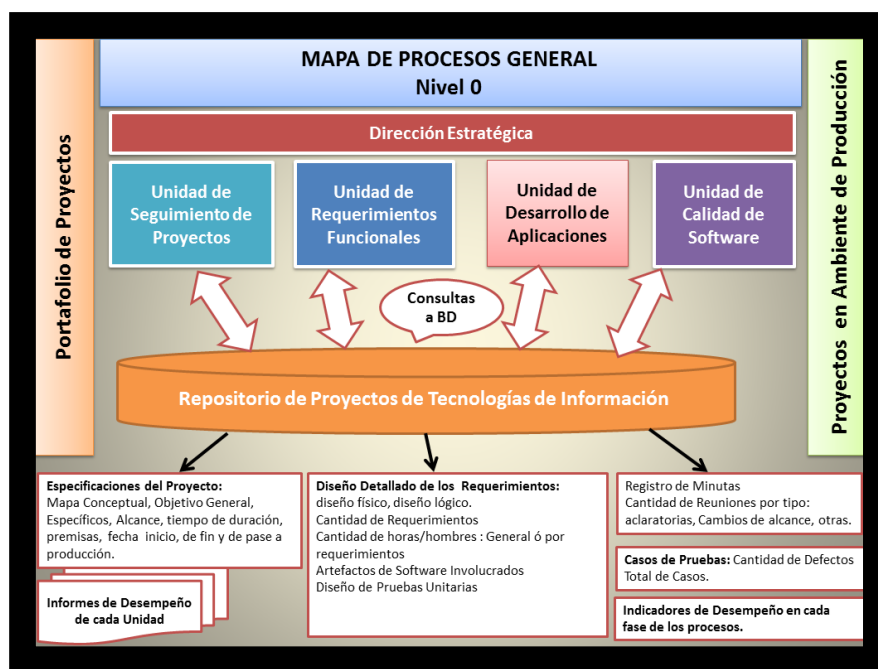


Figura Nro. 18 - Mapa de Proceso General del Modelo Propuesto

Este diseño está soportado por una base de datos Proyectos de Tecnologías de Información, al cual podrá acceder cualquiera de las personas que trabajen en alguna de las Unidades: seguimiento de proyectos, Requerimientos Funcionales, Desarrollo de Aplicaciones, Calidad de Software. El insumo de este proceso es la información del Portafolio de Proyectos y luego del proceso obtendremos a la salida, Proyectos en ambiente de Producción. El proceso está conformado por los cuatro subprocesos, que se explicarán detalladamente en los siguientes mapas de procesos propuestos para cada una de las cuatro unidades de trabajo.

5.1.2 Mapa de Proceso Unidad de Seguimiento de Proyectos

En la Figura Nro. 19 se representa gráficamente la propuesta del Mapa de Proceso de la Unidad de Seguimiento de Proyectos.

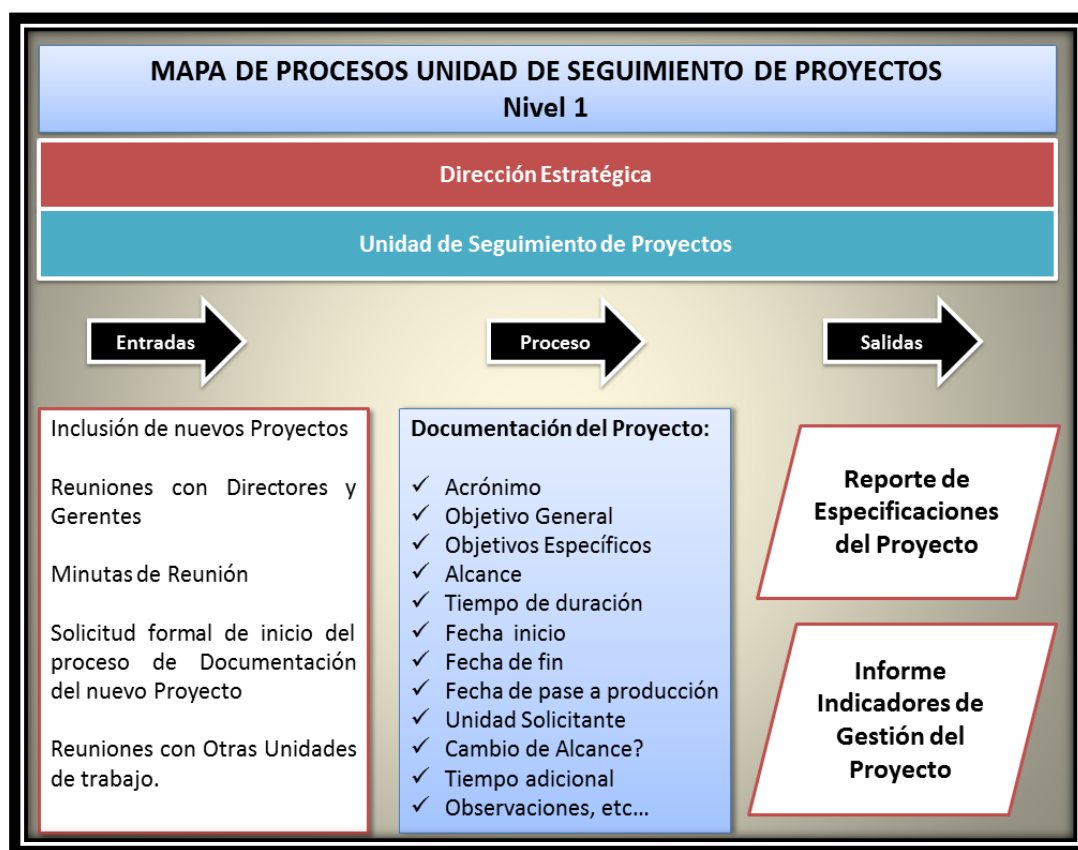


Figura Nro. 19
Mapa de Proceso Unidad de Seguimiento de Proyectos

Este diseño tiene como insumo la Inclusión de nuevos proyectos en el portafolio, reuniones con Directores y Gerentes de área donde se determina si un proyecto es aprobado o no, minutas de reuniones previas, reuniones con las otras unidades de trabajo y la solicitud formal de inicio de Proyecto.

5.1.3 Mapa de Proceso Unidad de Requerimientos Funcionales

En la Figura Nro. 20 se representa gráficamente la propuesta del Mapa de Proceso de la Unidad de Requerimientos Funcionales.

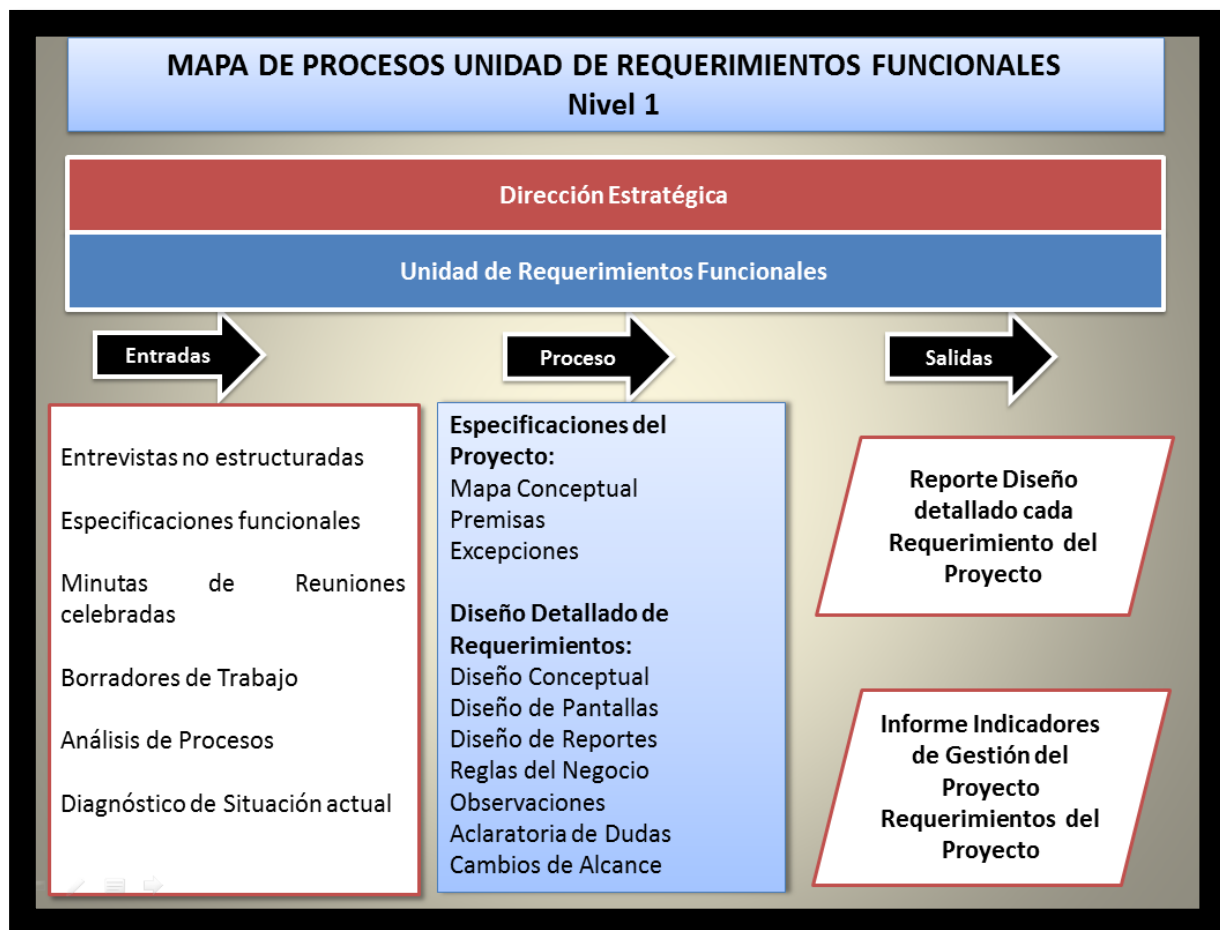


Figura Nro. 20
Mapa de Proceso Unidad de Requerimientos Funcionales

Este diseño tiene como insumo la celebración de entrevistas no estructuradas que le realizan a los usuarios funcionales, quienes aportan todas las especificaciones funcionales de los requerimientos, minutas celebradas previamente con información asociada a los requerimientos, borradores de trabajo resultantes de las reuniones, análisis de Procesos y diagnóstico de situación actual, la cual será procesada para obtener el diseño detallado de cada requerimiento y los Indicadores de Gestión de la unidad.

5.1.4 Mapa de Proceso Unidad de Desarrollo de Aplicaciones

En la Figura Nro. 21 se representa gráficamente la propuesta del Mapa de Proceso de la Unidad de Desarrollo de Aplicaciones.

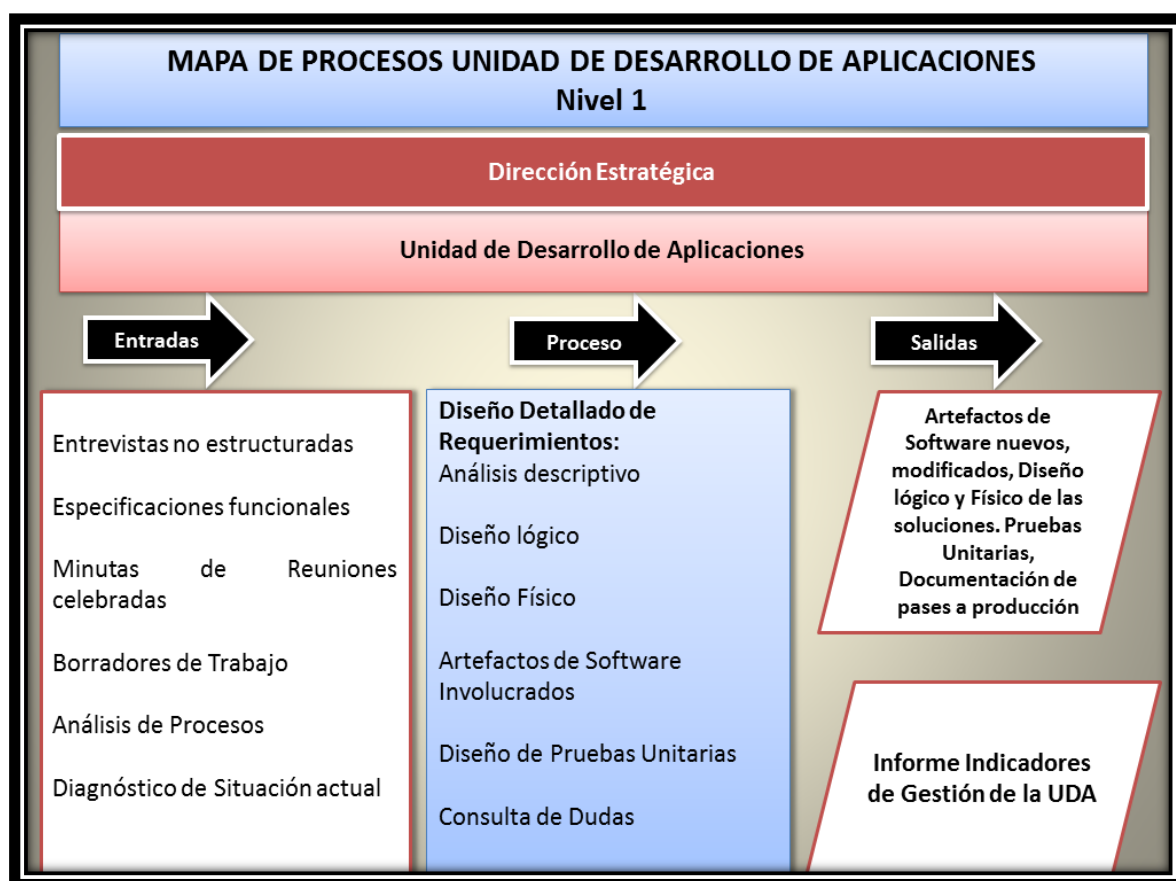


Figura Nro. 21
Mapa de Proceso Unidad de Desarrollo de Aplicaciones

Este diseño tiene como insumo la celebración de entrevistas no estructuradas que le realizan a los usuarios funcionales coordinadas por la URF, y toda la documentación que esta última proporcione: minutas de esas reuniones, Requerimiento de especificaciones funcionales, borradores de trabajo realizados en las reuniones, análisis de procesos y diagnóstico de la situación actual; y luego de procesada esta información obtendrá a la salida los artefactos de software con sus respectivas pruebas unitarias documentadas para que sirva de insumo a la siguiente unidad.

5.1.5 Mapa de Proceso Unidad de Calidad de Software

En la Figura Nro. 22 se representa gráficamente la propuesta del Mapa de Proceso de la Unidad de Calidad de Software.

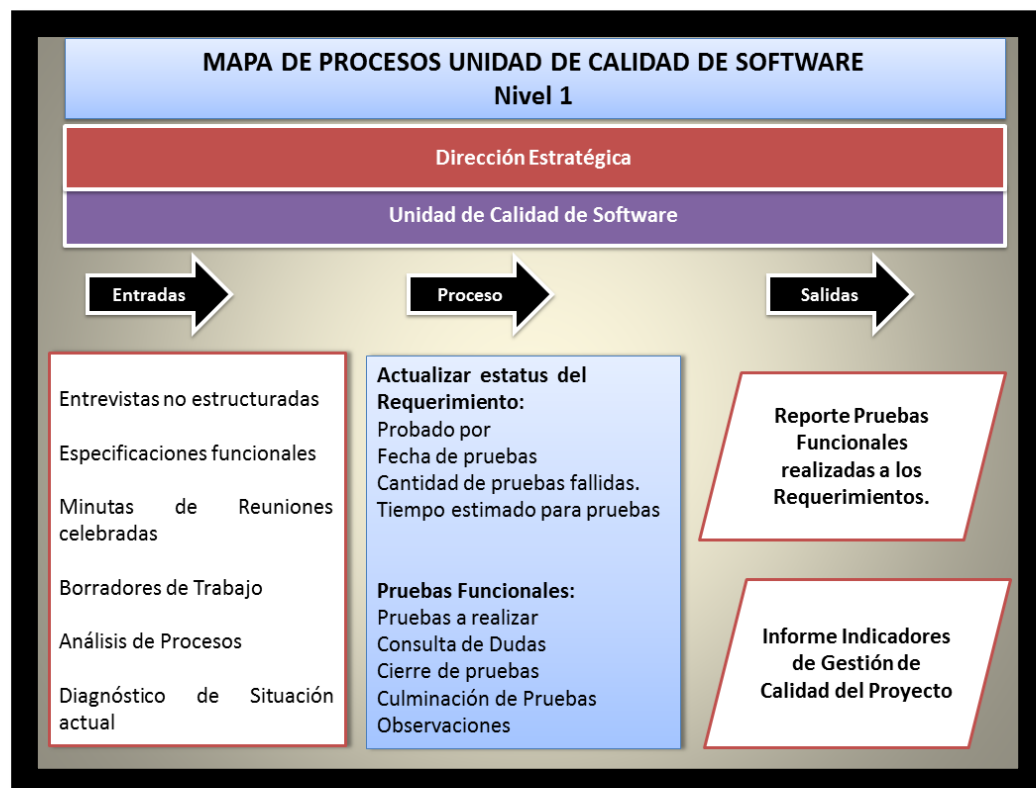


Figura Nro. 22
Mapa de Proceso Unidad de Calidad de Software

5.2 Diseño del Modelo Propuesto con BPMN 2.0

El Proceso inicia cuando los Usuarios Funcionales (UF) solicitan la incorporación de un nuevo proyecto al Portafolio existente “Solicitar Registro de Proyectos de TI” Figura Nro. 23, bajo el formato 1-SOLREGPROY_IT (sección 5.3 Formatos del Modelo Propuestos pág. 116).

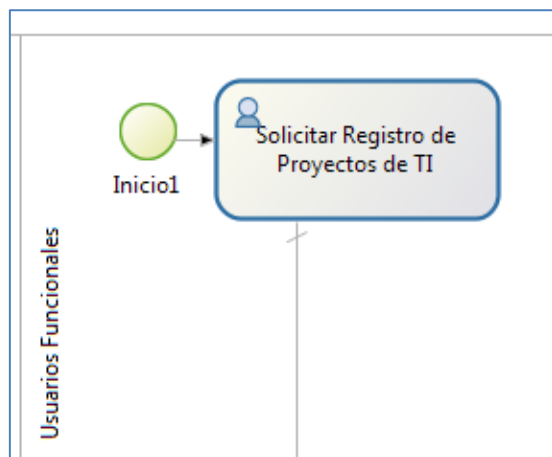


Figura Nro. 23- Solicitar Registro de Proyectos de TI

En la Figura Nro. 24- Registrar Proyectos de TI, se observa el siguiente paso correspondiente a la Unidad de Seguimiento de Proyectos (USP) donde debe registrar en el sistema los datos complementarios del Proyecto, asignándole un acrónimo y otros datos relevantes según el formato 2-REGPROY_IT (sección 5.3 Formatos del Modelo Propuestos pág. 116), teóricamente pueden solicitarse la incorporación de varios Proyectos en el portafolio, por lo cual es necesario realizar una reunión con los directores de área para definir cuál de los proyectos serán aprobados y el nivel de prioridad de cada uno de ellos. Una vez celebrada la reunión la USP se encargará de cambiar el estatus del Proyecto a “Aprobado” o “No Aprobado” (pueden aprobarse varios proyectos), este nivel de aprobación enviará un correo automático con el cambio de estatus respectivo a los solicitantes (UF); seguidamente crea la plantilla Gantt para cada uno de ellos bajo el Formato 3-PLANTILLAGANTT_IT (sección 5.3 Formatos del Modelo Propuestos pág. 118).

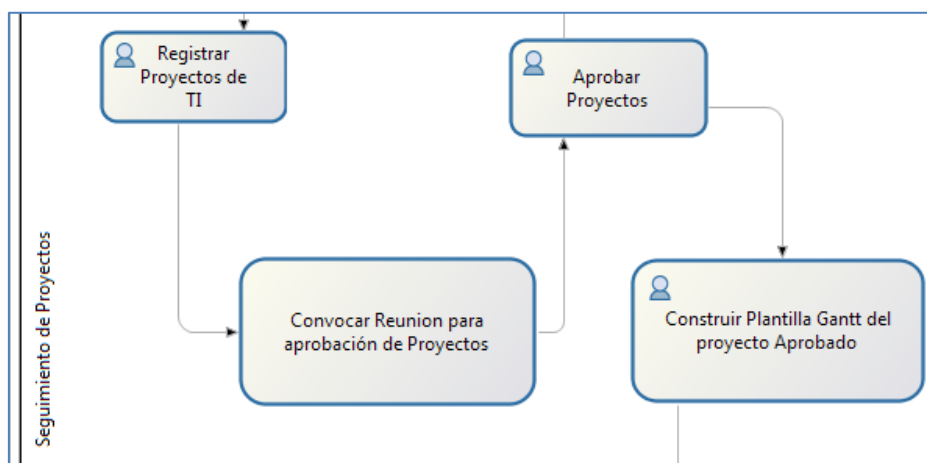


Figura Nro. 24- Registrar Proyectos de TI

En Figura Nro. 25- Notificación de Inicio de Proyecto de TI, indica que el proceso será capaz de informar automáticamente a la Unidad de Requerimientos Funcionales (URF), a través de un correo electrónico, que debe iniciarse el proceso de Levantamiento de Información para los Proyectos Aprobados indicando el nivel de prioridad “Recibir Notificación de Inicio de Proyecto”.

La URF cargará en el sistema las programación de las reuniones planificadas para llevar a cabo el levantamiento de Información, invitando a todas las partes involucradas, es un requisito que asistan a la reuniones los usuarios Funcionales y algunos integrantes de la UDA; el sistema enviará una notificación vía correo para dar por enterado de la reunión. En la Celebración de la Reunión, por cada reunión que se programe, el sistema creará una plantilla de la minuta siguiendo el formato 4-MINUTAS_PROY (sección 5.3 Formatos del Modelo Propuestos pág. 119), la cual debe ser llenada el mismo día que se celebra la reunión, de ser posible en línea. De cada reunión se va recopilando información que el personal de Requerimientos Funcionales debe identificar como Premisa, Reglas o Excepción, las cuales deben ser llenadas en el apartado del proyecto destinado para tal fin. Las reuniones pueden ser tipificadas como Levantamiento de Requerimientos (LR), Aclaratorias (AC), Otras (OT).

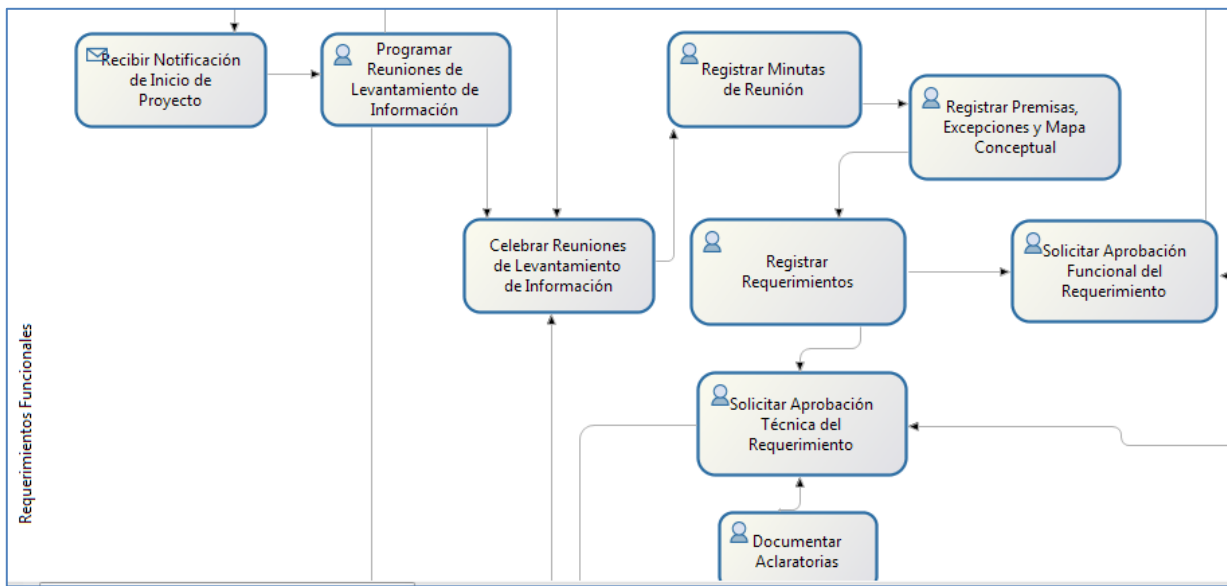


Figura Nro. 25- Notificación de Inicio de Proyecto de TI

Una vez que se tengan claras las especificaciones técnicas y funcionales del Proyecto, deben desglosarse los requerimientos de uno a uno y registrarlos en el sistema bajo el Formato 5-REQUERIMIENTOS_PROY (sección 5.3 Formatos del Modelo Propuestos pág. 122), de manera que un Proyecto estará constituido por uno o varios requerimientos, los cuales serán tipificados de acuerdo a su naturaleza. Una vez que la URF registre los requerimientos, deberá solicitar la aprobación tanto técnica como funcional del mismo, por lo que deben existir dos aprobaciones una por parte de la Unidad de Desarrollo de Aplicaciones (UDA) y otra por parte de los usuarios funcionales (UF).

En la Figura Nro. 26- Aprobación de Requerimientos ,se representa gráficamente explica lo siguiente: si la UDA tiene algún tipo de dudas con respecto al requerimiento, deberá registrarlas en el sistema en la sección correspondiente para ello según el formato 6-ACLARAR_REQ (sección 5.3 Formatos del Modelo Propuestos pág. 124), del requerimiento respectivo, esta dudas podrán visualizarla los integrantes de la URF, quienes deben documentar las aclaratorias, para proseguir con el ciclo de Solicitar la Aprobación Técnica del Requerimiento y asignar los tiempos estimados para el desarrollo del mismo. En caso de que la UDA no presente

dudas pero considera que el Requerimiento NO debe ser aprobado técnicamente, debe registrar la causa o motivo por el cual no será aprobado, ya que con esta información la URF podrá aplicar los correctivos o ajustes a los aspectos técnicos del Requerimiento, para así continuar con el ciclo de Aprobación Técnica del Documento de Especificaciones Funcionales. Para la aprobación se usa el formato APROBAR_REQ (sección 5.3 Formatos del Modelo Propuestos pág. 126).

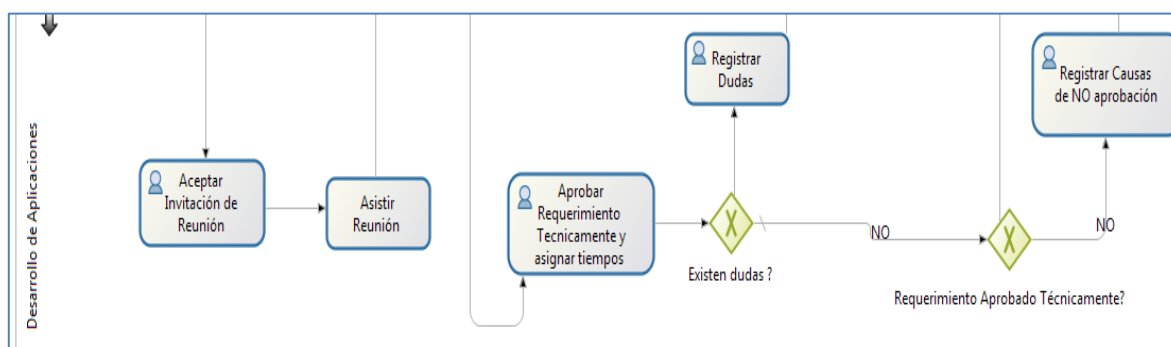


Figura Nro. 26- Aprobación de Requerimientos

De forma análoga la URF solicita la aprobación del Requerimiento desde el punto de vista funcional a los usuarios Funcionales, quienes también tienen la opción de registrar las causas de NO Aprobación para que la URF haga las correcciones respectivas (Figura Nro. 27).

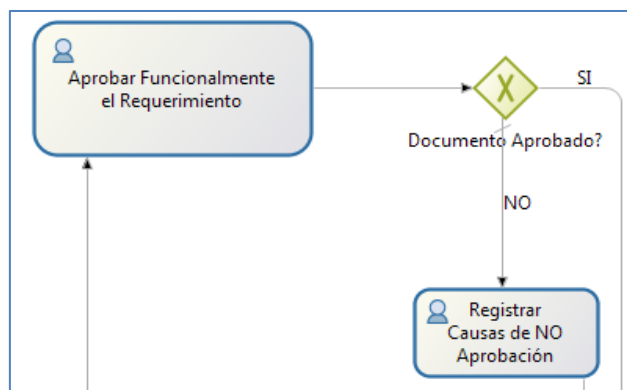


Figura Nro. 27- Aprobación Funcional del Requerimiento

Cuando el Requerimiento está aprobado técnica y funcionalmente, la USP actualiza la plantilla Gantt registrando asignando los tiempos estimados para el desarrollo, procede cambiar el estatus del Requerimiento a Aprobado (Figura Nro. 28).

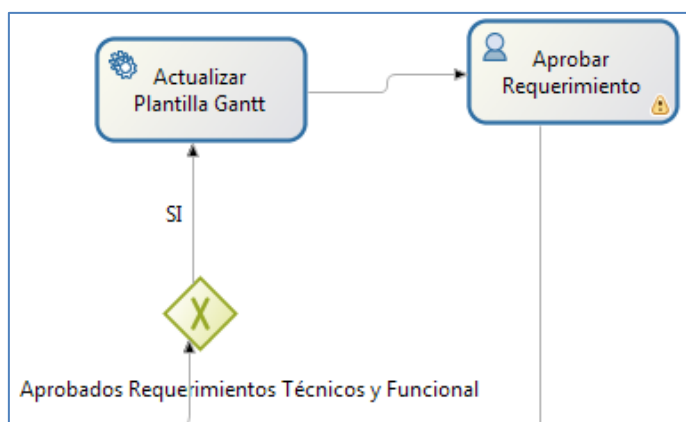
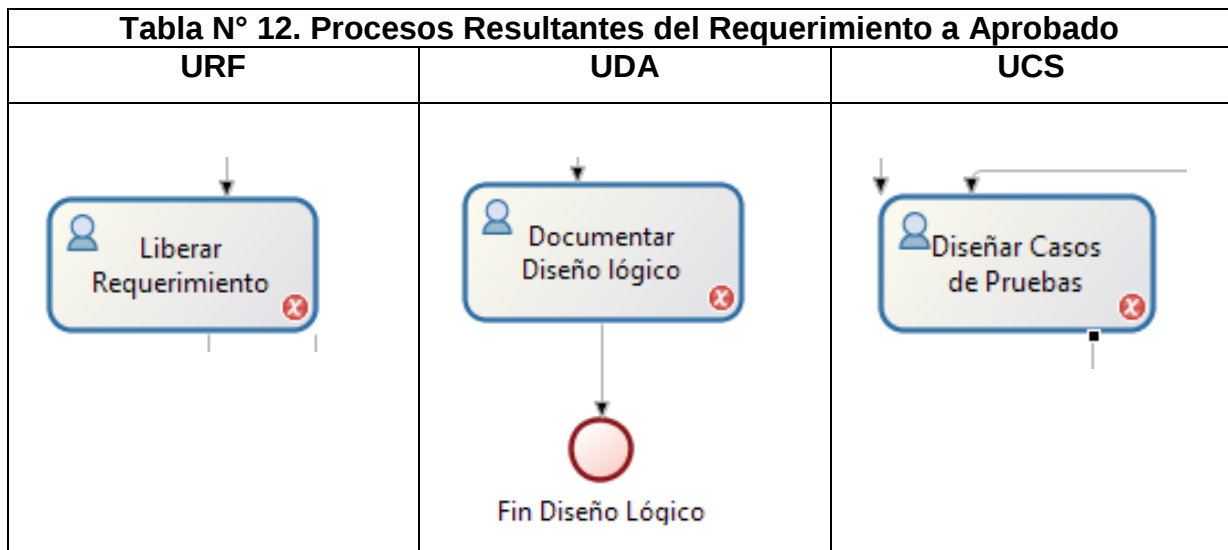


Figura Nro. 28- Actualizar Plantilla Gantt

El cambio de estatus, le permitirá a la URF liberar el Requerimiento para que puedan iniciar los trabajos de Diseño Lógico de la solución por parte de la UDA y el diseño de casos de pruebas por parte de la UCS, de manera que un requerimiento puede estar compuesto por varios casos de prueba, estos se identificarán con un correlativo por cada requerimiento. Esta última actividad es complementada por la UDA quien evalúa el diseño de las pruebas y hace sus observaciones en función de realizar sus pruebas unitarias, de esta manera las pruebas unitarias y funcionales tendrán una misma misión. Con la revisión de los casos de prueba se envía un correo automático con esta notificación a todas las Unidades, en especial a la USP para que actualice la Plantilla Gantt con los tiempos asignados para las Pruebas Unitarias y Pruebas Funcionales.



Una vez que todos los casos de pruebas han sido registrados, la USP programa una reunión con todas las partes involucradas para informar el Inicio de la Construcción del Proyecto, para ello la persona encargada de esta actividad debe actualizar la plantilla Gantt con la fecha de inicio y fin del proyecto y registrar la minuta de reunión en el sistema. Este evento sirve de insumo para dar inicio a la construcción física de los componentes de software de todos y cada uno de los requerimientos, quedando la responsabilidad del lado de la UDA, quien por cada requerimiento culminado, debe documentar las pruebas unitarias, e informar culminación del requerimiento para que la UCS pueda iniciar las pruebas parciales de cada requerimiento (Figura Nro. 29).

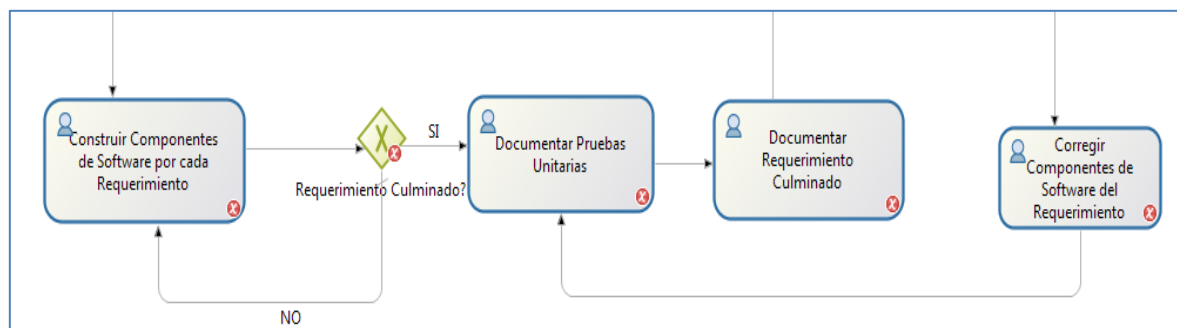


Figura Nro. 29- Construcción de Componentes de Software

La UCS efectuará las pruebas funcionales junto a los dueños de los procesos (usuarios funcionales) y si la prueba es satisfactoria el caso de prueba debe cambiarse de estatus a CERRADO, en el momento de que todos los casos de pruebas de todos los requerimiento han sido terminados, se debe Notificar Culminación de Pruebas Funcionales a las unidades restantes (Figura Nro. 30).

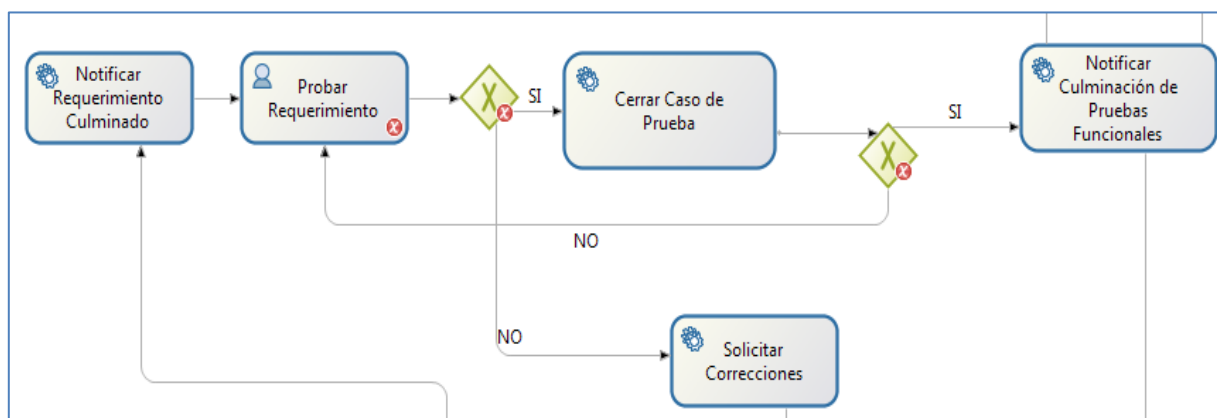


Figura Nro. 30- Culminación de Pruebas Funcionales

Cuando la USP recibe la notificación, convoca a una reunión con todas las partes para informar la culminación de Pruebas Funcionales, llena la reunión de la minuta y llena el acta de certificación que a su vez debe ser firmada por los solicitantes para poder proceder con la Documentación del pase a producción por parte de la UDA. (Figura Nro. 31).

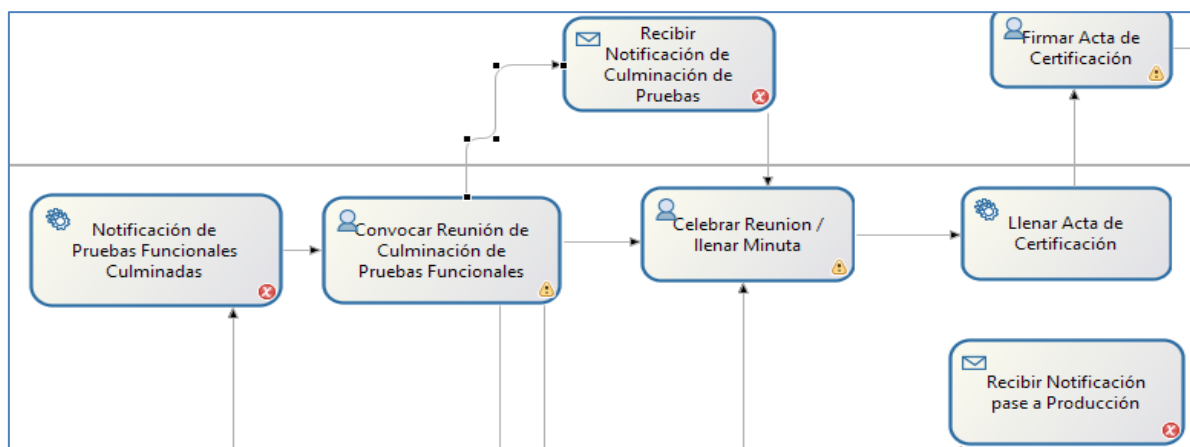


Figura Nro. 31- Firma de Acta de Certificación

Una vez que el Proyecto es migrado a producción con todos sus componentes de software asociados a cada requerimiento, se procede Notificar a todas las Unidades el pase ejecutado en producción y se espera la certificación final por parte del Usuario Funcional. El Usuario Funcional envía una notificación de Aceptación o Rechazo del cambio aplicado en producción, en el primer caso, que es el caso ideal, la UDA procede a Documentar el cierre del Proyecto, de lo contrario se activa un proceso de contingencia para revertir los cambios aplicados.

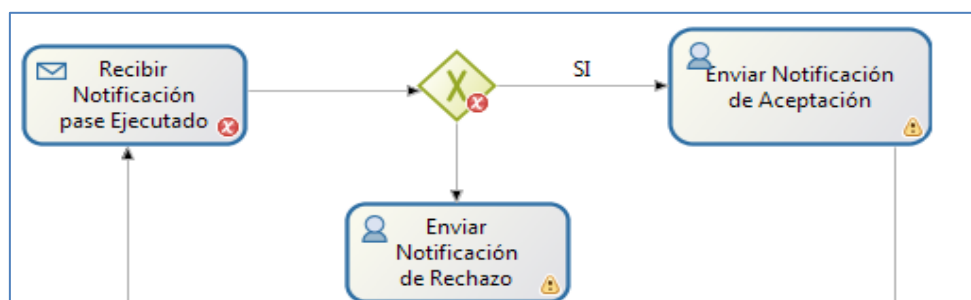


Figura Nro. 32- Aceptación de Pase a Producción

El proceso culmina con alguna de las acciones tomadas por la UDA, que dependen del resultado de la notificación enviada por el usuario funcional, si es pase a producción es rechazado se activa un plan de contingencia (el cual ya está debidamente documentado en las minutas), de lo contrario, se cierra el proyecto satisfactoriamente.

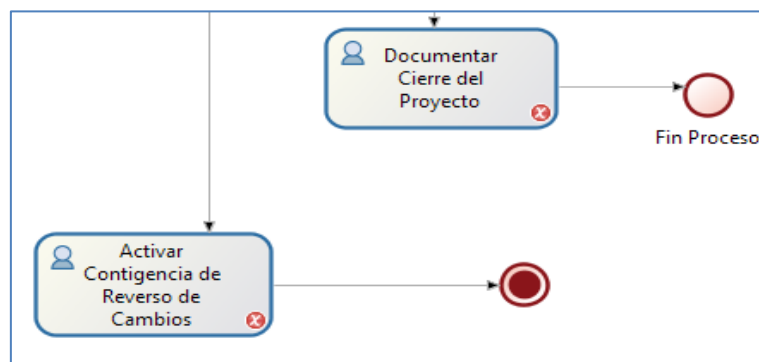


Figura Nro. 33- Cierre de Proyecto

5.3 Formatos del Modelo Propuestos

En esta sección se presentan un conjunto de once formatos que conforman parte del resultado del estudio del presente trabajo de investigación, estos formatos son los siguientes: Solicitud de Registro de Proyectos de TI, Registro de Proyectos de TI, Plantilla Gantt de Proyectos de TI, Minuta de Proyectos de IT, Requerimientos de Proyectos de TI, Aclaratorias de Dudas de Requerimientos de Proyectos de TI, Aprobación de Requerimientos de Proyectos de TI, Diseño de Pruebas de Requerimientos de Proyectos de TI, Cierre de Caso de Prueba Funcional de Requerimientos, Cierre de Caso de Prueba Funcional de Requerimientos, Cierre de Proyectos de TI.

5.3.1 Solicitud de Registro de Proyectos de TI

Este formato permite a una unidad de la organización solicitar el desarrollo de un nuevo proyecto en el área de Tecnología de Información, donde debe indicar la fecha de solicitud, el nombre de la unidad solicitante, nombres, apellidos y cargo del solicitante; así como la descripción y beneficios que aportará el desarrollo de la Solicitud.

1	Solicitud de Registro de Proyectos de TI	SOLREGPROY_IT	
Datos de la Solicitud			
Fecha de Solicitud		Unidad Solicitante	
Nombres y Apellidos del Solicitante		Cargo del Solicitante	
Descripción de la Solicitud			
Beneficios de la Solicitud			

Leyenda:

Fecha de Solicitud: Fecha del día de la solicitud

Unidad Solicitante: Nombre de la Unidad Organizativa que hace la solicitud

Nombres y Apellidos del Solicitante: Indicar nombres y apellidos del solicitante

Cargo del Solicitante: Indicar el Cargo del Solicitante (Director /Gerente /Jefe)

Descripción de la Solicitud: Describir detalladamente lo que se desea solicitar

Beneficios de la Solicitud: Indique los beneficios que se obtendrán con la realización de este Proyecto.

5.3.2 Registro de Proyectos de TI

Una vez que la solicitud (**SOLREGPROY_IT**) de ha sido evaluada, se utiliza este formato (**REGPROY_IT**), que permite registrar el estatus del proyecto luego de su evaluación: Aprobado, en Proceso, No aprobado.

2	Registro de Proyectos de TI		REGPROY_IT
Datos del Proyecto			
Descripción del Proyecto			
Nombre del Proyecto			
Acrónimo			
Estatus	APROBADO —	EN PROCESO —	NO APROBADO —
Fecha de Cambio de estatus			
Motivo de Aprobación / Desaprobación			

Leyenda

Descripción del Proyecto: debe indicar exactamente lo que el solicitante escribió en la solicitud del formato SOLREGPROY_IT

Nombre del Proyecto: Indica el nombre bajo el cual será denominado el proyecto. Ej.: Automatización del proceso de facturación.

Acrónimo: Es un identificador único para cada proyecto con máximo de doce caracteres. Ej.: AUTOPROCFAC

Estatus: se debe indicar con una equis (X) si fue aprobado o no aprobado en el comité de aprobación.

Fecha de Cambio de estatus: Indicar la fecha en la que el estatus fue cambiado.

Motivo de Aprobación / Desaprobación: llenar este campo si hay alguna observación importante que hacer, obligatorio si NO es aprobado.

5.3.3 Plantilla Gantt de Proyectos de TI

Una vez que la solicitud es registrada con estatus aprobado en el formato REGPROY_IT se procede a llenar la Plantilla Gantt para estimar el tiempo de ejecución del Proyecto.

Leyenda:

Datos de Básicos del Proyecto: Los campos de esta sección son heredados de los formatos SOLREGPROY_IT y REGPROY_IT.

Detalle de las Actividades: Esta sección se creará automáticamente cuando el proyecto sea aprobado, con los renglones preestablecidos. Debe alimentarse en cada oportunidad que sea necesaria por la Unidad de Seguimiento de Proyectos, con toda la información suministrada por el resto de las unidades (UDA/UCS/URF).

3	Plantilla Gantt de Proyectos de TI		PLANTILLAGANTT_IT	
Datos de Básicos del Proyecto				
Acrónimo		Fecha de Aprobación		
Nombre del Proyecto				
Unidad Solicitante		Fecha de Solicitud		
Detalle de las Actividades				
Nro.	Actividad	Fecha Desde	Fecha Hasta	Cantidad de días
1	<i>Fase de Levantamiento de Información</i>			
	Reunión 1			
	Reunión 2			
	Otras Reuniones			
2	<i>Fase de Construcción</i>			
	Requerimiento 1			
	Requerimiento 2			
3	<i>Fase de Pruebas Unitarias</i>			
	Pruebas Unitarias Requerimiento 1			
	Pruebas Unitarias Requerimiento 2			
4	<i>Fase de Prueba Funcionales</i>			
	Pruebas Funcionales Requerimiento 1			
	Pruebas Funcionales Requerimiento 2			
5	<i>Corrección y Ajustes</i>			
6	<i>Reunión de Cierre de Pruebas Funcionales</i>			
7	<i>Documentación pase a Producción</i>			
8	<i>Pase a Producción</i>			

5.3.4 Minuta de Proyectos de IT

Este formato facilita el registro de todas las reuniones que se celebren durante la ejecución del proyecto. Los Datos de Básicos del Proyecto se heredan de los formatos anteriormente mencionados. En la sección Datos Básicos de la Reunión se debe indicar todo lo relacionado con la reunión.

4	Minuta de Proyectos de IT			MINUTAS_PROY	
Datos de Básicos del Proyecto					
Acrónimo				Fecha de Aprobación	
Nombre del Proyecto					
Unidad Solicitante				Fecha de Solicitud	
Datos Básicos de la Reunión					
Reunión Nro.		Hora		Fecha	
Unidad Convocante					
Moderador					
Objetivo					
Tipo de Reunión					
Aprobación de Proyectos				Inicio de Proyecto	
Levantamiento de Información				Fin de Pruebas Funcionales	
Invitados Asistentes					
1	Nombres	Apellidos		Unidad de Servicios	
2					
3					
4					
Puntos Tratados					
1					
2					
3					
Acuerdos					
1					
2					
3					
Observaciones					

Leyenda:

Datos de Básicos del Proyecto: Los campos de esta sección son heredados de los formatos SOLREGPROY_IT y REGPROY_IT.

Datos Básicos de la Reunión

Reunión Nro.: Numero consecutivo de la reunión, se reinicia por cada Proyecto/ Tipo de Reunión.

Hora: Indicar la hora de la reunión.

Fecha: Indicar la fecha de la Reunión.

Unidad Convocante: Los posibles convocantes son los participantes en el proyecto: UDA, UCS, URF, USP y la Unidad Administrativa de los Usuario Solicitantes.

Moderador: Indicar Nombre y Apellido de la persona que convoca la reunión.

Objetivo: describir brevemente el objetivo de la reunión.

Tipo de Reunión: Debe marcar con una equis (X) el tipo de reunión, dependiendo de su naturaleza.

Invitados Asistentes: en esta sección se colocará el nombre, apellidos y unidad donde trabaja la persona que asistió a la reunión.

Puntos Tratados: Se debe enumerar todos los puntos tratados.

Acuerdos: Se debe llenar si hubo conclusiones, acuerdos o próximos pasos a tomar. No se debe llenar si la reunión tuvo carácter informativo.

Observaciones: es opcional agregar algún comentario u observación adicional en esta sección.

5.3.5 Requerimientos de Proyectos de IT

Este formato permite llenar detalladamente todos y cada uno de los requerimientos, que serán desarrollados dentro del proyecto, es uno de los formatos más importantes ya que permite el enlace con todas las unidades de trabajo.

5	Requerimientos de Proyectos de TI		REQUERIMIENTOS_PROY
Datos Básicos del Proyecto			
Acrónimo		Unidad Solicitante	
Nombre del Proyecto			
Datos Básicos del Requerimiento			
Requerimiento Nro.		Fecha/Hora	
Nombre			
Descripción General			
Descripción Detallada			
1			
2			
3			
Premisas			
1			
2			
3			
Excepciones			
1			
2			
3			
Mapa Conceptual			
Diseño de Pantallas/Reportes			
Descripción de campos y secciones de Pantallas/Reportes			

Leyenda:

Datos de Básicos del Proyecto: Los campos de esta sección son heredados de los formatos SOLREGPROY_IT y REGPROY_IT.

Datos Básicos del Requerimiento

Requerimiento Nro.: número consecutivo del requerimiento, se reinicia por cada proyecto

Fecha/Hora: se cargará automáticamente, indicando la fecha / hora del registro del requerimiento.

Nombre: se debe asignar un nombre al Requerimiento debe ser corto y descriptivo. Ej.: Diseño de Reporte de Facturación diaria.

Descripción General: Se utiliza para ampliar el nombre del requerimiento.

Descripción Detallada: indicar la secuencia en especie de lista de chequeo con lo que debe cumplir el requerimiento para satisfacer el requisito.

Premisas: Detallar cuales son las premisas que deben tomarse en cuenta para realizar el requerimiento. Ej.: los datos a consultar desde una tabla X deben estar llenos o con estatus activos, etc.

Excepciones: se debe indicar que cosas no cumple el requerimiento.

Mapa Conceptual: para cada requerimiento se deber hacer un esquema de entrada/proceso/salida, que permita visualizar de manera gráfica lo que se pide.

Diseño de Pantallas/Reportes: Se debe indicar una imagen con la distribución de los campos bien sea de una pantalla de entrada o de un reporte de salida.

Descripción de campos y secciones de Pantallas/Reportes: Se debe describir todos y cada una de las secciones y campos que conformen la sección “Diseño de Pantallas/Reportes”.

5.3.6 Aclaratorias de Dudas de Requerimientos de Proyectos de IT

Este formato permite que dos unidades de trabajo diferentes trabajen sobre él, una describe las dudas existentes con respecto a un requerimiento, y la otra aclara dichas dudas.

6	Aclaratorias de Dudas de Requerimientos de Proyectos de TI		ACLARAR_REQ	
Datos Básicos del Proyecto				
Acrónimo		Unidad Solicitante		
Nombre del Proyecto				
Datos Básicos del Requerimiento				
Requerimiento Nro.		Estatus	CON DUDAS _____ ACLARATORIAS REGISTRADAS _____	
Nombre				
Tiempo de Desarrollo estimado		Analista Asignado		
Desaprobado por			Fecha Desaprobación	
Datos de Aclaratorias				
Nro.		Fecha / hora		
Solicitante			Unidad	
Dudas existentes				
Aclaratoria de Dudas				

Leyenda:

Este formato será llenado inicialmente por la UDA o por el usuario funcional, en caso de requerir aclaratorias. Luego la URF culmina el llenado completando el campo aclaratoria de dudas.

Datos de Básicos del Proyecto: Los campos de esta sección son heredados de los formatos SOLREGPROY_IT y REGPROY_IT.

Datos Básicos del Requerimiento: Los campos Requerimiento Nro. y Nombre son heredados del formato REQUERIMIENTOS_PROY.

Estatus: Marcar con una equis (X) el estatus correspondiente.

Tiempo de Desarrollo estimado: la UDA debe indicar el tiempo en horas que dedicará al desarrollo del requerimiento sin incluir tiempo de pruebas unitarias.

Analista Asignado: Se debe colocar el nombre del desarrollador de software que construirá el requerimiento.

Desaprobado por: Este campo debe llenarse con el nombre/apellido de la persona que indica que el requerimiento no será desarrollado.

Fecha Desaprobación: fecha en la que se toma la decisión de no desarrollar el requerimiento.

Datos de Aclaratorias

Nro.: consecutivo de aclaratoria por cada requerimiento.

Fecha / hora: se registrará automáticamente al momento de guardar.

Solicitante: nombres y apellidos de la persona que solicita la aclaratoria.

Unidad: Unidad que solicita la aclaratoria (Usuarios Funcionales o UDA).

Dudas existentes: Si la UDA tiene dudas con respecto a la redacción o términos utilizado, o cualquier otra duda, debe llenar esta campo para que la URF pueda proceder con las aclaratorias correspondientes.

Aclaratoria de Dudas: este campo solo debe ser llenado por la URF si el campo Dudas existentes tiene información.

5.3.7 Aprobación de Requerimientos de Proyectos de IT

Una vez que todas las dudas de un requerimiento hayan sido aclaradas y todo el equipo de trabajo esté de acuerdo con el desarrollo que debe hacerse, se procede a llenar este formato para indicar que el mismo fue aprobado; o desaprobado, si no aplica su desarrollo.

7	Aprobación de Requerimientos de Proyectos de TI			APROBAR_REQ
Datos Básicos del Proyecto				
Acrónimo		Unidad Solicitante		
Nombre del Proyecto				
Datos Básicos del Requerimiento				
Requerimiento Nro.		Estatus	APROBADO — NO APROBADO —	
Nombre				
Tiempo de Desarrollo estimado		Analista Asignado		
Aprobado por			Fecha Aprobación	
Datos de Aclaratorias				
Nro.		Fecha / hora		
Solicitante			Unidad	
Dudas existentes		Aclaratorias de dudas		
Motivo de Desaprobación				
Desaprobado por			Fecha Desaprobación	

Leyenda:

Este formato será llenado inicialmente por la URF en caso de requerir aclaratorias

Datos de Básicos del Proyecto: Los campos de esta sección son heredados de los formatos SOLREGPROY_IT y REGPROY_IT.

Datos Básicos del Requerimiento: Los campos Requerimiento Nro., Nombre, Tiempo de Desarrollo estimado, Analista Asignado, Dudas existentes y Aclaratoria de Dudas son heredados del formato REQUERIMIENTOS_PROY.

Estatus: se debe marcar con una equis (X) un de las opciones disponibles.

Desaprobado por: Este campo debe llenarse con el nombre/apellido de la persona que indica que el requerimiento no será desarrollado.

Fecha Desaprobación: fecha en la que se toma la decisión de no desarrollar el requerimiento.

5.3.8 Diseño de Pruebas de Requerimientos de Proyectos de IT

Este formato es llenado inicialmente por el desarrollador de software quien establece las premisas de las pruebas unitarias realizadas durante el desarrollo, luego, bajo el mismo formato los analistas de pruebas complementan el diseño de las mismas.

8	Diseño de Pruebas de Requerimientos de Proyectos de TI		DISEÑO_PRUEBA
Datos Básicos del Proyecto			
Acrónimo		Unidad Solicitante	
Nombre del Proyecto			
Datos Básicos del Requerimiento			
Requerimiento Nro.		Estatus	APROBADO
Nombre			
DISEÑO DE PRUEBAS			
Descripción General			
Tiempo estimado Pruebas Unitarias		Analista Desarrollo	
Tiempo estimado Pruebas Funcionales		Analista Funcional	
Pruebas			
1	Prueba 1		
1.1	Detalle Prueba 1.1	Consideraciones Prueba 1.1	
1.2	Detalle Prueba 1.2	Consideraciones Prueba 1.2	
1.3	Detalle Prueba 1.3	Consideraciones Prueba 1.3	
2	Prueba 2		
2.1	Detalle Prueba 2.1	Consideraciones Prueba 2.1	
2.2	Detalle Prueba 2.2	Consideraciones Prueba 2.2	
2.3	Detalle Prueba 2.3	Consideraciones Prueba 2.3	

Leyenda:

Este formato será llenado por dos unidades la UDA y la URF para estimar los tiempos requeridos para las pruebas unitarias y funcionales.

Datos de Básicos del Proyecto: Los campos de esta sección son heredados de los formatos SOLREGPROY_IT y REGPROY_IT.

Datos Básicos del Requerimiento: Los campos Requerimiento Nro., Nombre, Tiempo de Desarrollo estimado, Analista Asignado, Dudas existentes y Aclaratoria de Dudas son heredados del formato REQUERIMIENTOS_PROY.

DISEÑO DE PRUEBAS

Descripción General: describir de forma clara y concisa en que consiste la prueba a realizar

Tiempo estimado Pruebas Unitarias: Este campo debe ser llenado por la UDA indicando el tiempo en horas.

Analista Desarrollo: Analista asignado a realizar las pruebas unitarias

Tiempo estimado Pruebas Funcionales: Este campo debe ser llenado por la URF indicando el tiempo en horas.

Analista Funcional: Analista asignado a realizar las pruebas funcionales.

Prueba-n: indica la cantidad de pruebas que se realizaran al requerimiento.

Detalle Prueba n-m: describir los pasos para realizar la prueba.

Consideraciones Prueba n-m: indicar que consideraciones deben tomarse en cuenta para realizar las pruebas.

5.3.9 Cierre de Requerimientos de Proyectos de IT

Este formato será llenado por dos unidades la UDA y la URF para estimar los tiempos requeridos para las pruebas unitarias y funcionales.

9	Cierre de Requerimientos de Proyectos de TI		CIERRE_REQ
Datos Básicos del Proyecto			
Acrónimo		Unidad Solicitante	
Nombre del Proyecto			
Datos Básicos del Requerimiento			
Requerimiento Nro.		Estatus	CERRADO
Nombre			
Tiempo Pruebas Unitarias			
Tiempo estimado		Analista	
Tiempo Real invertido		Analista	
Cerrado por		Fecha de Cierre	

Leyenda:

Datos de Básicos del Proyecto: Los campos de esta sección son heredados de los formatos SOLREGPROY_IT y REGPROY_IT.

Datos Básicos del Requerimiento: Los campos Requerimiento Nro., Nombre, Tiempo de Desarrollo estimado, Analista Asignado, Dudas existentes y Aclaratoria de Dudas son heredados del formato REQUERIMIENTOS_PROY.

Tiempo Pruebas Unitarias

Tiempo estimado: El valor de este campo se hereda del formato 9-DISEÑO_PRUEBA

Tiempo Real invertido: Se asignará el valor del tiempo total invertido en el desarrollo del requerimiento.

Cerrado por: Analista que cambia el estatus acerrado.

Fecha de Cierre: fecha y hora en la que se cierra el requerimiento.

5.3.10 Cierre de Caso de Prueba Funcional de Requerimientos de Proyectos de IT

Este formato es llenado por el equipo de Calidad de Software, quienes luego de haber completado las pruebas de un requerimiento proceden a registrar el cierre del mismo.

10	Cierre de Caso de Prueba Funcional de Requerimientos de Proyectos de TI	CIERRE_PRUEBAF
----	---	----------------

Datos Básicos del Proyecto			
Acrónimo		Unidad Solicitante	
Nombre del Proyecto			
Datos Básicos del Requerimiento			
Requerimiento Nro.		Estatus	CERRADO
Nombre			
DISEÑO DE PRUEBAS			
Descripción General			
Tiempo Pruebas Funcionales			
Tiempo estimado		Analista	
Tiempo Real invertido		Analista	
Cerrado por		Fecha de Cierre	

Leyenda:

Datos de Básicos del Proyecto: Los campos de esta sección son heredados de los formatos SOLREGPROY_IT y REGPROY_IT.

Datos Básicos del Requerimiento: Los campos Requerimiento Nro., Nombre, Tiempo de Desarrollo estimado, Analista Asignado, Dudas existentes y Aclaratoria de Dudas son heredados del formato REQUERIMIENTOS_PROY.

DISEÑO DE PRUEBAS / Descripción General: el valor de este campo se hereda desde el formato DISEÑO_PRUEBA al igual que el campo Tiempo estimado.

Tiempo Pruebas Funcionales

Tiempo Real invertido: Tiempo total en realizar las pruebas funcionales.

Analista: Analista que realizó las pruebas funcionales.

Cerrado por: Analista que cierra la prueba funcional.

Fecha de Cierre: Fecha y hora de cierre de la prueba funcional.

5.3.11 Cierre de Proyectos de IT

Este formato es llenado es llenado por la Unidad de Seguimiento de Proyectos una vez que valide con todos los participantes en el proyecto, que todos los requerimientos están probados, cerrados y certificados.

11	Cierre de Proyecto de TI	CIERRE_PROY
----	--------------------------	-------------

Datos Básicos del Proyecto			
Acrónimo		Unidad Solicitante	
Nombre del Proyecto			
Tiempos del Proyecto			
Tiempo estimado		Analista	
Tiempo Real invertido		Analista	
Cerrado por		Fecha de Cierre	
Observaciones			

Leyenda:

Datos de Básicos del Proyecto: Los campos de esta sección son heredados de los formatos SOLREGPROY_IT y REGPROY_IT.

Tiempos del Proyecto

Tiempo estimado: El valor de esta campo se obtiene de la programación inicial arrojado por el Gantt.

Tiempo Real invertido: este valor lo llena el analista de proyectos con el valor final luego de los ajustes realizados al Gantt.

Cerrado por: Analista de proyectos que cierra el Proyecto.

Fecha de Cierre: Fecha y hora de cierra del Proyecto.

Observaciones: Observaciones finales del proyecto.

5.4Diseño de Indicadores de Gestión

5.4.1 Indicadores de la Unidad de Seguimiento de Proyectos

- ✓ Indicador Porcentaje de Horas Hombres (H/H) por Unidad / Gráfica Total de Recursos

$$\frac{\text{Cantidad de h/h de la UDA en el proyecto} \times 100}{\text{Cantidad de h/h Total en el proyecto}}$$

$$\frac{\text{Cantidad de h/h de la URF en el proyecto} \times 100}{\text{Cantidad de h/h Total en el proyecto}}$$

$$\frac{\text{Cantidad de h/h de la USP en el proyecto} \times 100}{\text{Cantidad de h/h Total en el proyecto}}$$

$$\frac{\text{Cantidad de h/h de la UCS en el proyecto} \times 100}{\text{Cantidad de h/h Total en el proyecto}}$$

- ✓ Porcentaje de Desviación Final: si este patrón de medición es menor que uno (1) se lee como un proyecto que fue ejecutado eficientemente, cuando el valor resulta mayor a uno se puede leer que hubo mayor consumo de h/h en el proyecto a lo inicialmente calculado.

$$\frac{\text{Tiempo final invertido para la ejecución del proyecto} \times 100}{\text{Tiempo Inicial calculado en la ejecución del proyecto}}$$

- ✓ Porcentaje de Desviación en cambios de alcances: lo ideal es que este indicador resulte igual a cero, lo cual indica que no hubo cambios de alcance en el proyecto.

$$\frac{\text{Tiempo total invertido en cambios de alcances/proyecto} \times 100}{\text{Tiempo final invertido en la ejecución del proyecto}}$$

5.4.2 Indicadores de la Unidad de Requerimientos Funcionales

- ✓ Indicador de Requerimientos Aprobados: lo ideal es que este indicador sea igual a uno, es decir que todos los requerimientos sean aprobados

$$\frac{\text{Cantidad de Requerimientos Aprobados}}{\text{Cantidad Total de Requerimientos por Proyecto}}$$

- ✓ Indicador de Requerimientos NO Aprobados: lo ideal es que este indicador sea igual a cero, es decir que no existan requerimientos NO aprobados.

$$\frac{\text{Cantidad de Requerimientos NO Aprobados}}{\text{Cantidad Total de Requerimientos por Proyecto}}$$

- ✓ Porcentaje de Dudas en el proyecto: lo ideal es que este indicador sea igual a cero (0), es decir, que no exista dudas en los requerimientos solicitados.

$$\frac{\text{Cantidad Total de Dudas presentadas en el proyecto} \times 100}{\text{Cantidad Total de Requerimientos por Proyecto}}$$

- ✓ Indicador de Reuniones: este indicador debe ser menor o igual a uno, es decir lo ideal es que se celebre una reunión para levantar uno o más requerimientos. Si se celebran varias reuniones para un mismo requerimiento, sería un punto de atención.

Cantidad Total de Reuniones Celebradas

Cantidad Total de Requerimientos por Proyecto

5.4.3 Indicadores de la Unidad de Calidad de Software

- ✓ Porcentaje de Pruebas Defectuosas: lo ideal es que este porcentaje sea igual a cero.

$$\frac{\text{Cantidad Total de Pruebas Defectuosas} \times 100}{\text{Cantidad Total de Pruebas Funcionales Efectuadas}}$$

- ✓ Porcentaje de Pruebas Satisfactorias: lo ideal es que este porcentaje sea igual a cien por ciento.

$$\frac{\text{Cantidad Total de Pruebas Satisfactorias} \times 100}{\text{Cantidad Total de Pruebas Funcionales Efectuadas}}$$

- ✓ Porcentaje de Pruebas NO Programadas: lo ideal es que este porcentaje sea igual a cero.

$$\frac{\text{Cantidad Total de Pruebas No programadas} \times 100}{\text{Cantidad Total de Pruebas Funcionales Efectuadas}}$$

- ✓ Indicador de Efectividad de Pruebas Funcionales Total: lo ideal es que este indicador sea igual a uno. Si resultara mayor a uno sería indicativo de que tal vez se incluyeron más pruebas de las previstas inicialmente.

$$\frac{\text{Cantidad Total de Tiempo en Pruebas funcionales inicial}}{\text{Cantidad Total de Tiempo en Pruebas funcionales final}}$$

- ✓ Indicador de Solicitudes de Corrección de Defectos: lo ideal es que este indicador sea igual a cero, es decir que las pruebas sean satisfactorias desde el primer momento.

$$\frac{\text{Cantidad Total de Solicitudes de Corrección de Requerimientos} \times 100}{\text{Cantidad Total de Requerimientos}}$$

5.4.4 Indicadores de la Unidad de Desarrollo de Aplicaciones

- ✓ Indicador de Objetos Nuevos: indica la cantidad de objetos(pantallas de entrada, reportes de salida, objetos de base de datos) fueron desarrollados desde cero. Si es igual a uno, indicaría que es un proyecto nuevo sin incidencias sobre objetos ya existentes.

$$\frac{\text{Cantidad Total de Nuevos Artefactos de Software en el Proyecto}}{\text{Cantidad Total de Artefactos de Software en el Proyecto}}$$

- ✓ Indicador de Objetos Modificados: indica la cantidad de objetos(pantallas de entrada, reportes de salida, objetos de base de datos) ya existentes en ambiente de producción fueron adecuados o modificados. Si es igual a uno, indicaría que es un proyecto de mantenimiento, y pudiesen existir afectaciones en el pase a producción .

$$\frac{\text{Cantidad Total de Artefactos de Software Modificados /Proyecto}}{\text{Cantidad Total de Artefactos de Software en el Proyecto}}$$

- ✓ Porcentaje de Pantallas nuevas

$$\frac{\text{Cantidad de Pantallas de Entrada de Datos Nuevas x 100}}{\text{Cantidad Total de Artefactos de Software en el Proyecto}}$$

- ✓ Índice de Pantallas modificadas

$$\frac{\text{Cantidad de Pantallas de Entrada de Datos Modificadas x 100}}{\text{Cantidad Total de Artefactos de Software en el Proyecto}}$$

- ✓ Índice de Reportes nuevos

$$\frac{\text{Cantidad de Reportes de Salida de datos Nuevos x 100}}{\text{Cantidad Total de Artefactos de Software en el Proyecto}}$$

- ✓ Índice de Reportes modificados

$$\frac{\text{Cantidad de Reportes de Salida de datos Modificados x 100}}{\text{Cantidad Total de Artefactos de Software en el Proyecto}}$$

- ✓ Índice de Objetos de Base de Datos nuevos

$$\frac{\text{Cantidad de Objetos de BD nuevos x 100}}{\text{Cantidad Total de Artefactos de Software en el Proyecto}}$$

- ✓ Índice de Objetos de Base de Datos modificados

$$\frac{\text{Cantidad de Objetos de BD modificados x 100}}{\text{Cantidad Total de Artefactos de Software en el Proyecto}}$$

- ✓ Porcentaje Efectividad (tiempo dedicado) de los requerimiento

$$\frac{\text{Cantidad de Objetos culminados en el tiempo preestablecido} \times 100}{\text{Cantidad Total de Artefactos de Software en el Proyecto}}$$

$$\frac{\text{Cantidad de Objetos culminados fuera del tiempo preestablecido} \times 100}{\text{Cantidad Total de Artefactos de Software en el Proyecto}}$$

5.5 Validación del Modelo según el estándar ISO 9126

La fase de evaluación del Modelo Propuesto, utiliza una lista de ponderación para Validación basada en los estándares de calidad sugeridos la norma ISO/IEC 9126, que permite especificar y evaluar la calidad del software desde diferentes criterios asociados con los requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoria de software. El estándar describe seis características generales: Funcionalidad, Confiabilidad, Usabilidad, Eficiencia, Mantenibilidad y Portabilidad; se diseñó un formato para tabular cada una de estas características y debe asignársele una ponderación para finalmente ubicarlo en una escala de valores. El formato de validación consta de una Portada, los parámetros de evaluación, Escala de Evaluación y Tabulación del Formato de Evaluación. Para hacer la evaluación, se sugiere un comité evaluador con tres perfiles de usuario, desarrolladores de software que no hayan participado en el desarrollo para validar mantenibilidad y portabilidad, usuarios funcionales para evaluar performance, eficiencia, facilidad de uso, entre otros aspectos. y gerentes generales para evaluar el panorama general (tiempos/costos).

5.5.1.1 Portada del Formato de Evaluación

Datos de Básicos del Proyecto			
Acrónimo		Fecha de la evaluación	
Nombre del Proyecto			
Objetivos general			
Objetivos Específicos			
Participantes en la Evaluación			
Nro.	Nombre y Apellidos	Cargo	Firma

5.5.2 Parámetros de Evaluación

Funcionalidad (0-20)

Marcar con Equis (X) la ponderación otorgada, cada renglón tiene un valor de 1,1 este valor se multiplica por el valor de la columna marcada y luego se suman todos los resultados, de esta manera si todos los criterios son marcados en la columna tres el resultado máximo será de veinte puntos.

Usabilidad y Confiabilidad (0-20)

Marcar con Equis (X) la ponderación otorgada el valor marcado se multiplica por el valor de la columna y luego se suman todos los resultados, de esta manera si todos los criterios son marcados en la columna tres el resultado máximo será de veinte puntos.

Eficiencia (0-15)

Marcar con Equis (X) la ponderación otorgada, cada renglón tiene un valor de 1,1 este se multiplica por el valor de la columna y luego se suman todos los resultados, d esta manera si todos los criterios son marcados en la columna tres el resultado máximo será de quince puntos.

Capacidad de Mantenimiento y Portabilidad (0-15)

Marcar con Equis (X) la ponderación otorgada el valor marcado se multiplica por el valor de la columna y luego se suman todos los resultados, d esta manera si todos los criterios son marcados en la columna tres el resultado máximo será de quince puntos.

✓ Escala de Evaluación.

91 – 100 puntos	➔	Excelente Software	- Aceptable
81 – 90 puntos	➔	Muy Buen Software	- Aceptable
61 – 80 puntos	➔	Buen Software	- Aceptable
50 – 60 puntos	➔	Software Deficiente	- NO Aceptable
0 – 50 puntos	➔	No es un producto de Software	- NO Aceptable

5.5.3 Tabulación del Formato de Evaluación

Características de Funcionalidad				
Denominación / Ponderación	0	1	2	3
Funcionalidad				
Adecuación				
Exactitud				
Interoperabilidad				
Seguridad				
Conformidad de la funcionalidad				

Leyenda

Adecuación: Indique si la solución de software provee un adecuado conjunto de funciones cumplan las tareas y objetivos especificados por el usuario.

Exactitud La capacidad del software para hacer procesos y entregar los resultados solicitados con precisión o de forma esperada.

Interoperabilidad La capacidad del software de interactuar con uno o más sistemas específicos.

Seguridad La capacidad del software para proteger la información y los datos de manera que los usuarios o los sistemas no autorizados no puedan acceder a ellos para realizar operaciones, y la capacidad de aceptar el acceso a los datos de los usuarios o sistemas autorizados

Conformidad de la funcionalidad La capacidad del software de cumplir los estándares referentes a la funcionalidad.

Características de Confiabilidad						
Denominación / Ponderación	0	1	2	3	4	5
Madurez						
Tolerancia a errores						
Recuperabilidad						
Conformidad de la fiabilidad						

Leyenda

Madurez La capacidad que tiene el software para evitar fallas cuando encuentra errores. Ejemplo, la forma como el software advierte al usuario cuando realiza operaciones en la unidad de diskett vacía, o cuando no encuentra espacio suficiente el disco duro donde está almacenando los datos.

Tolerancia a errores La capacidad que tiene el software para mantener un nivel de funcionamiento en caso de errores.

Recuperabilidad La capacidad que tiene el software para restablecer su funcionamiento adecuado y recuperar los datos afectados en el caso de una falla.

Conformidad de la fiabilidad La capacidad del software de cumplir a los estándares o normas relacionadas a la fiabilidad.

Características de Usabilidad					
Denominación / Ponderación	0	1	2	3	4
Entendimiento					
Aprendizaje					
Operabilidad					
Atracción					
Conformidad de uso					

Leyenda

Entendimiento La capacidad que tiene el software para permitir al usuario entender si es adecuado, y de una manera fácil como ser utilizado para las tareas y las condiciones particulares de la aplicación.

En este criterio se debe tener en cuenta la documentación y de las ayudas que el software entrega.

Aprendizaje La forma como el software permite al usuario aprender su uso. También es importante considerar la documentación.

Operabilidad La manera como el software permite al usuario operarlo y controlarlo.

Atracción La presentación del software debe ser atractivo al usuario. Esto se refiere a las cualidades del software para hacer más agradable al usuario, ejemplo, el diseño gráfico.

Conformidad de uso La capacidad del software de cumplir los estándares o normas relacionadas a su usabilidad

Características de Eficiencia				
Denominación / Ponderación	0	1	2	3
Comportamiento de tiempos				
Utilización de recursos				
Conformidad de eficiencia				

Leyenda

Comportamiento de tiempos Los tiempos adecuados de respuesta y procesamiento, el rendimiento cuando realiza su función en condiciones específicas. Ejemplo, ejecutar el procedimiento mas complejo del software y esperar su tiempo de respuesta, realizar la misma función pero con mas cantidad de registros.

Utilización de recursos La capacidad del software para utilizar cantidades y tipos adecuados de recursos cuando este funciona bajo requerimientos o condiciones establecidas. Ejemplo, los recursos humanos, el hardware, dispositivos externos.

Conformidad de eficiencia La capacidad que tiene el software para cumplir con los estándares o convenciones relacionados a la eficiencia.

Características de Capacidad de Mantenimiento				
Denominación / Ponderación	0	1	2	3
Capacidad de ser analizado				
Cambiabilidad				
Estabilidad				
Facilidad de prueba				
Conformidad de facilidad de mantenimiento				

Leyenda

Capacidad de ser analizado La forma como el software permite diagnósticos de deficiencias o causas de fallas, o la identificación de partes modificadas.

Cambiabilidad La capacidad del software para que la implementación de una modificación se pueda realizar, incluye también codificación, diseño y documentación de cambios.

Estabilidad La forma como el software evita efectos inesperados para modificaciones del mismo.

Facilidad de prueba La forma como el software permite realizar pruebas a las modificaciones sin poner el riesgo los datos.

Conformidad de facilidad de mantenimiento La capacidad que tiene el software para cumplir con los estándares de facilidad de mantenimiento.

Características de Portabilidad					
Denominación / Ponderación	N/A	0	1	2	3
Adaptabilidad					
Facilidad de instalación					
Coexistencia					
Reemplazabilidad					
Conformidad de portabilidad					

Leyenda

Adaptabilidad Es como el software se adapta a diferentes entornos especificados (hardware o sistemas operativos) sin que implique reacciones negativas ante el cambio. Incluye la escalabilidad de capacidad interna (Ejemplo: Campos en pantalla, tablas, volúmenes de transacciones, formatos de reporte, etc.).

Facilidad de instalación La facilidad del software para ser instalado en un entorno específico o por el usuario final.

Coexistencia La capacidad que tiene el software para coexistir con otro o varios software, la forma de compartir recursos comunes con otro software o dispositivo.

Reemplazabilidad La capacidad que tiene el software para ser reemplazado por otro software del mismo tipo, y para el mismo objetivo. Ejemplo, la reemplazabilidad de una nueva versión es importante para el usuario, la propiedad de poder migrar los datos a otro software de diferente proveedor

Conformidad de portabilidad La capacidad que tiene el software para cumplir con los estándares relacionados a la portabilidad

5.5.4 Resultados de la Evaluación

Datos de Básicos del Proyecto			
Acrónimo			Fecha de la evaluación
Nombre del Proyecto			
Participantes en la Evaluación			
Nro.	Nombre y Apellidos	Escala	Puntos Obtenidos
1	Funcionalidad	25	
2	Confiabilidad	20	
3	Usabilidad	20	
4	Eficiencia	10	
5	Capacidad de Mantenimiento	15	
6	Portabilidad	15	
Total Escala de Evaluación		100	
Total del puntaje de la Evaluación		(en número)	(en letras)
Recomendaciones			

6 CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

En el Diagnóstico de la situación actual de los procesos estudiados, se determinó que existen debilidades en el proceso de levantamiento de información, ya que dentro de las prácticas actuales, los requerimientos no son identificados de manera particular, y que, a pesar de que se celebran reuniones el resultado de estas no llegan a todos los que participan en el proyecto. En el proceso de Desarrollo de aplicaciones, se determinó que los desarrolladores de software pueden participar paralelamente en el mantenimiento cotidiano de aplicaciones y en el desarrollo de nuevos proyectos, así como también desconocen detalladamente los procesos a desarrollar. En el proceso de calidad de software, se determinó que la calidad es aplicada al final de desarrollo, no al principio ni en las fases intermedias. En el proceso de seguimiento de proyectos, se determinó que a pesar de que interactúa con todos los demás procesos, no existe un mecanismo de integración con estos. Finalmente en el diagnóstico, se concluye que todos los trabajadores encuestados tienen experiencia trabajando en proyectos de desarrollo de Software y conocen el trabajo que deben hacer dentro de un proyecto, sin embargo, pocos conocen los procesos de manera detallada, la planificación inicial de los proyectos y no están asistiendo con frecuencia a las reuniones de seguimiento.

Se construyó un Modelo que integra a los cuatro procesos que participan en el Desarrollo de Software para las Pequeñas y Medianas Empresas de Venezuela: Unidad de Seguimiento de Proyectos, Unidad de Requerimientos Funcionales, Unidad de Calidad de Software, Unidad de Desarrollo de Aplicaciones y a los Usuarios Funcionales; el cual está compuesto por un diagrama de objetos diseñados con notación BPMN 2.0 y un conjunto de once formatos. El modelo dicta las pautas para que la responsabilidad del registro individual de requerimientos sea compartida con diversas unidades, el diseño de pruebas unitarias y funcionales con responsabilidad compartida entre la UDA y la UCS, el

registro de minutas por tipo de reunión. El modelo auspicia la participación de la UCS desde fases tempranas del proyecto, promueve la integración total de la Unidad de Seguimiento de Proyectos al manejo de los requerimientos, exige el uso de los indicadores de gestión y, finalmente proporciona un formato de evaluación basado en ISO 9126, todo esto con el fin último de implementar el uso de mejores prácticas.

Finalmente, se concluye que con la implementación del presente MODELO DE INTEGRACION DE LOS PROCESOS INVOLUCRADOS EN EL DESARROLLO DE SOFTWARE EN LAS PEQUEÑAS Y MEDIANAS EMPRESAS EN VENEZUELA, se obtiene una mayor organización de los proyectos de desarrollo de software, ya que con el uso de los formatos permite documentar, controlar e integrar toda la información que se genera durante todo el proceso, aportando un documento final que permite un fácil análisis y estudio de los resultados arrojados por el proyecto, lo cual constituye un valor agregado para futuros desarrollos de aplicaciones, y prevé que en la ausencia de cualquier recurso humano, la información esté accesible para todos los participantes en el proyecto, y fácil integración de nuevos recursos.

6.2 RECOMENDACIONES

Se recomienda automatizar el modelo propuesto, implementando una base de datos corporativa, que permita registrar a todos los perfiles de usuarios ,crear la unidades, registrar el Gantt, las minutas, es decir, todos los formatos y objetos que son utilizados por el modelo, que facilite la generación automática de los indicadores de gestión, así como también compartir información entre todos los autorizados, y que todo el acervo documental que se va generando con la ejecución de cada proyecto, sirva como historia para tomar decisiones en la ejecución de proyectos futuros.

El modelo propuesto está diseñado para las pequeñas y medianas empresas del en Venezuela, sin embargo, no es limitante su uso exclusivamente a este sector,

puede ser utilizado en cualquier empresa de Desarrollo de Software que tenga una estructura similar a la planteada.

Se recomienda dividir las funciones de los desarrolladores de software en recursos diferentes, para evitar que participen paralelamente en actividades mantenimiento cotidiano de aplicaciones y en el desarrollo de nuevos proyectos.

Recomendaciones educativas: abrir una línea de investigación para la puesta en práctica, seguimiento y evaluación del modelo propuesto.

REFERENCIAS BIBLIOGRAFICAS

- Aguinaga, G., & Araujo, M. (2013). *Evaluación de la Norma ISO 12207 en el proceso de Desarrollo de Software de la empresa Logicel LTDA*. Sangolquí: Repositorio Dspace .
- Aguirre, L., & La Serna Palomino, N. (2015). *Desarrollo de un marco metodológico del proceso de verificación y validación de software para pequeñas y medianas empresas*. Lima: Universidad Nacional Mayor de San Marcos-Revista de la Facultad de Ingeniería Industrial.
- Bedini González, A. (2006). *Gestión de Proyectos de Software*. Buenos Aires: sin editorial.
- BORBÓN ARDILA, N. I. (2013). *NORMA DE EVALUACIÓN ISO/IEC 9126*. Colombia: Actividadreconocimiento.
- BPM, C. (2011). *El libro del BPM*. España: Club BPM.
- Calderón, E. (2013). *Madurez y planificación estratégica de proyectos BPM en el sistema financiero peruano*. Lima: UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS.
- Carvajal, Á. (2002). *Teorias y Modelos: Formas de Representación de la Realidad*. Costa Rica: Revista Comunicación.
- Conlago, M. (2014). *Plan de Moejora para el proceso de Desarrollo de Software de Intellectus System S.A*. Sangolquí: ESPE.
- Farez, M. (2015). *MODELO DE CALIDAD PARA LA GESTIÓN DE PROCESOS DE DESARROLLO DE SOFTWARE APLICADO AL SISTEMA INFORMÁTICO EN USO DE LA UNIVERSIDAD TÉCNICA DE MACHALA*. Latacunga: ESPE.
- García, J. (2015). *Una propuesta para el uso del paradigma guiado por modelos (MDE) para la definición y ejecución de procesos de negocios*. Sevilla: Universidad de Sevilla.
- García, L. (2015). *Un marco de referencia para facilitar la interoperabilidad y mantenibilidad de los modelos de procesos de software*. Sevilla: Universidad de Sevilla.
- Hernández, R., Fernández, C., & Baptista, M. (2010). *METODOLOGÍA de la Investigación*. México: Mc Graw Hill INTERAMERICANA EDITORES, S.A. DE C.V.

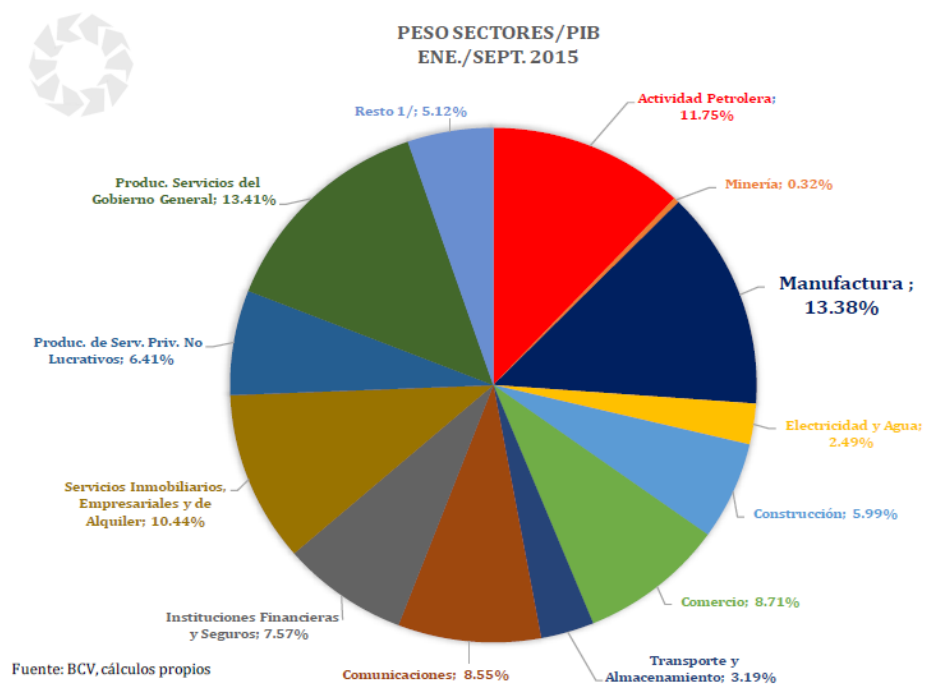
- Hoyle, D. &. (2002). *Del aseguramiento a la gestión de la calidad: el enfoque basado en procesos*. Madrid: AENOR.
- Hurtado de Barrera, J. (2000). *Metodología de la Investigación Holística*. Caracas: Fundación Sypal.
- Institute, S. E. (2010). *CMMI para Desarrollo Version 1.3*. España: Editorial Universitaria Ramón Areces.
- Lapiedra Alcam, R. (2011). *Introducción a la gestión de sistemas de información en la empresa*. Castelló de la Plana: UNIVERSITAT JAUME.
- Laudon, K., & Laudon, J. (2012). *SISTEMAS DE INFORMACIÓN GERENCIAL*. México: PEARSON EDUCACIÓN.
- Lemes Batista, A., & Machado Hernández, T. (2007). *SEGUNDO ENCUENTRO INTERNACIONAL SOBRE Las Medianas, Pequeñas y Micro-Empresas del Siglo XXI*. CUBA: EUMEDNET .
- Llanes-Font, M. (2013). *De la gestión por procesos a la gestión integrada por procesos*. Cuba: Universidad de Holguín.
- Macías García, M. (2007). *GUIA PARA LA IDENTIFICACIÓN Y ANÁLISIS GUIA PARA LA IDENTIFICACIÓN Y ANÁLISIS IDENTIFICACIÓN Y ANÁLISIS DE PROCESOS DE PROCESOS DE PROCESOS. V01*. Cadiz: Universidad de Cadiz.
- Marcos, J., Arroyo, A., Garzás, J., & Piattini, M. (2008). *La norma ISO/IEC 25000 y el proyecto KEMIS para su automatización con software libre*. Madrid: REICIS.
- Marín Sánchez, J. (2016). *Control de proyectos de software: actualidad y retos para la industria cubana*. La Habana: Ingeniare.
- Marín-Sánchez, J. (2014). *Proceso para la planificación y control de proyectos de software utilizando Xedro-GESPRO*. La Habana: Revista Cubana de Ciencias Informáticas.
- Martínez, A. (2012). *Propuesta Integral de un Modelo de Gestión por Procesos de negocio (PIM-GPN)*. México: UPUESA.
- Monje, C. (2011). *Metodología de la Investigación Cuantitativa y Cualitativa, Guía Didáctica*. Neiva: Universidad Surcolombiana del Docente.
- Orna, C. (2013). *Modelo de calidad para la etapa de planificación de Proyectos Medianos de Software*. Latacunga: ESPE.

- Palella, S., & Martins, F. (2012). *Metodología de la Investigación Cuantitativa*. Caracas: FEDUPEL.
- PMI, I. P. (2013). *Guía de los fundamentos para la dirección de proyectos (guía del PMBOK®)*. Pensilvania: GLOBALSTANDAR.
- Ponce, H. (2007). *LA MATRIZ FODA: ALTERNATIVA DE DIAGNÓSTICO Y DETERMINACIÓN DE ESTRATEGIAS DE INTERVENCIÓN EN DIVERSAS ORGANIZACIONES*. México: ENSEÑANZA E INVESTIGACIÓN EN PSICOLOGÍA .
- Pressman, R. S. (2010). *Ingeniería del software: un enfoque práctico*. México: MCGRAW-HILL INTERAMERICANA EDITORES, S.A.
- Sabino, C. (1992). *EL PROCESO DE INVESTIGACION*. Buenos Aires: Panapo.
- Santamaría, P. (2012). *ESTUDIO PARA LA IMPLEMENTACIÓN DE ADMINISTRACIÓN DE PROCESOS DE NEGOCIO (BPM) EN LA FUERZA AÉREA COLOMBIANA*. Bogotá, Colombia.
- Sanz, A. (2012). *Marco Metodológico para la mejora de las actividades de Verificación y Validación de Productos Software*. Leganés: UNIVERSIDAD CARLOS III DE MADRID.
- Sommerville, I. (2005). *Ingeniería del Software*. Madrid: PEARSON EDUCACION S.A.
- Suárez Lorenzo, F., & Garzás Parra, J. (2014). *I Jornada sobre Calidad del Producto Software e ISO 25000*. Santiago de Compostela: XUNTA DE GALICIA.

ANEXOS

ANEXO 1

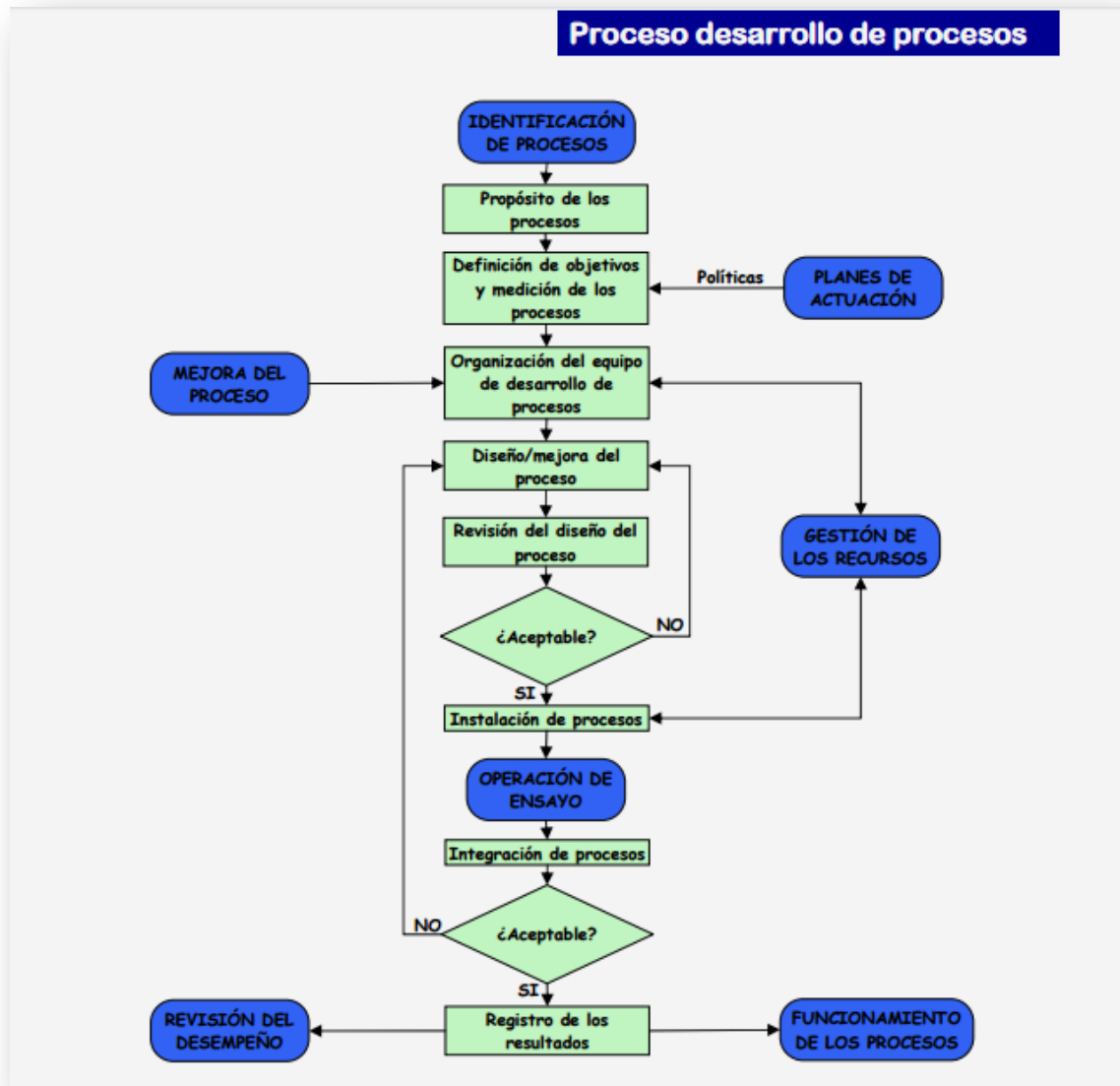
EVOLUCIÓN DEL PIB DE LOS PRINCIPALES SECTORES DESDE 1997-ENE. /SEPT. 2015



Fuente: RADIOGRAFÍA DE LA INDUSTRIA - CONINDUSTRIA

ANEXO 2

Diagramas de Flujo Relacionado con la Revisión de los Procesos



Fuente: “Del aseguramiento a la gestión de la calidad: el enfoque basado en procesos”; David Hoyle y John Thompson. Editado por AENOR.

ANEXO 3

Ejemplo de Mapa de Proceso de Sistemas y Telecomunicaciones de una Organización



Fuente: “Del aseguramiento a la gestión de la calidad: el enfoque basado en procesos”; David Hoyle y John Thompson. Editado por AENOR.

ANEXO 4

Capítulo 7

Diez prácticas recomendadas de BPM

En este capítulo

- _ Conoceremos el papel que desempeñan los profesionales en BPM
- _ Tomaremos las decisiones adecuadas sobre por dónde empezar
- _ Lograremos el equilibrio entre metodología y tecnología

Tenga en cuenta estas diez recomendaciones y estará en el buen camino para el éxito de BPM:

_ **Piense en el proceso; sea el proceso.** Establezca una organización interdisciplinaria que impulse y respalde su orientación de los procesos. Los equipos de procesos y los propietarios de los procesos deben elaborar planes juntos, reunirse periódicamente y trabajar en colaboración. Asegúrese de que el principal stakeholder participe directamente.

_ **¡Sea listo!** BPM es una nueva disciplina cuyos conocimientos pueden ser difíciles de adquirir. Pero las personas que le rodean tienen la mayor parte de lo que necesita. Es posible que no posean títulos, pero tendrán la comprensión adecuada de su negocio, las personas, los procesos y las habilidades básicas adecuadas para tener éxito. ¡Fórmelos! Haga un llamamiento a los proveedores de servicios para que le ayuden, especialmente a corto plazo a medida que realice los preparativos. Y, lo crea o no, ¡todo no lo va a aprender completamente con este libro! Salga y vea qué hacen las personas que le rodean. Asista a conferencias, visite sitios web, lea blogs. Busque compañías que hayan hecho esto antes, y logrado crear valor empresarial real.

_ **Adopte un ejecutivo.** Busque a los individuos de más experiencia cuya responsabilidad sea resolver los grandes problemas y reclútelos como patrocinadores, partidarios o BPM puede hacer por ellos. Hágales una demostración de lo que es posible y se subirán a bordo.

_ **Excelentes expectativas.** Los programas BPM involucran a muchas partes interesadas, cada una con su propia perspectiva y definición del éxito. Sea claro en lo referente a las propuestas de valor de sus programas y proyectos. Repítalas a menudo y preséntelas directamente, evitando que aumenten las expectativas.

_ **Seleccione una metodología.** ¿Cuenta ya en su compañía con un programa de mejora de los procesos como Lean o Six Sigma? Si es así, estos programas constituyen una base efectiva para BPM. Si no, seleccione algo. Seleccione la metodología de mejora y gestión de los procesos que le sea más adecuada, y conviértala en la piedra angular de su arquitectura de procesos.

_ **La tecnología adecuada.** Haga los deberes y elija la tecnología que mejor satisfaga sus necesidades y requisitos. La tecnología varía ampliamente de un proveedor a otro. Una vez que ha encontrado una tecnología que encaje con su arquitectura, asegúrese de ejecutarla recorriendo sus distintos pasos: pida demostraciones personalizadas y asegúrese de que el proveedor le muestre lo que puede hacer.

_ **Escuche la voz del cliente.** La finalidad de BPM es crear valor para el cliente. Utilice BPM para ver su negocio como lo ven sus clientes. A sus clientes no les importa cómo se hacen las cosas, lo único que les preocupa es experimentar un servicio excepcional y recibir lo que han pedido, al mejor precio y en el momento en que lo desean. Recuerde que todo lo que hace debe crear en última instancia más valor para sus clientes.

_ **Seleccione un proyecto.** ¡Tantos procesos y tan poco tiempo! ¿Cómo elegir? Seleccione el proyecto que proporcione la mayor rentabilidad para su negocio y que pueda completarse en tres meses o menos. Y recuerde, no tiene que mejorar procesos enteros de una vez.

– **Seleccione un proyecto.** ¡Tantos procesos y tan poco tiempo! ¿Cómo elegir? Seleccione el proyecto que proporcione la mayor rentabilidad para su negocio y que pueda completarse en tres meses o menos. Y recuerde, no tiene que mejorar procesos enteros de una vez.

– **Primero mida.** No comience a diseñar nuevos procesos hasta que comprenda lo que ocurre con los actuales. ¿Verdad que no permitiría que un médico le administrara un tratamiento sin un minucioso diagnóstico? De modo que no empiece a implementar cambios en los procesos hasta que haya diagnosticado el estado actual de estos. Mediante BAM, establezca las métricas básicas. Sólo entonces estará actuando desde una posición de conocimiento.

– **Planee qué cambiar.** BPM es un sistema para el cambio. Está concebido para ayudarle a identificar dónde se necesita el cambio y para que realice cambios rápidamente y pase al siguiente nivel en cuanto a rendimiento operacional se refiere. Con un conjunto de herramientas tan potente, necesita que BPM se rodee de elementos de apoyo para el cambio. Gestione el cambio, implemente políticas para realizar cambios, articule aprobaciones de cambio, reconozca eventos de cambio, mida el cambio, recompense el cambio.

Fuente: BPM, 2011

- Aguinaga, G., & Araujo, M. (2013). *Evaluación de la Norma ISO 12207 en el proceso de Desarrollo de Software de la empresa Logicel LTDA*. Sangolquí: Repositorio Dspace .
- Aguirre, L., & La Serna Palomino, N. (2015). *Desarrollo de un marco metodológico del proceso de verificación y validación de software para pequeñas y medianas empresas*. Lima: Universidad Nacional Mayor de San Marcos-Revista de la Facultad de Ingeniería Industrial.
- Bedini González, A. (2006). *Gestión de Proyectos de Software*. Buenos Aires: sin editorial.
- BORBÓN ARDILA, N. I. (2013). *NORMA DE EVALUACIÓN ISO/IEC 9126*. Colombia: Actividadreconocimiento.
- BPM, C. (2011). *El libro del BPM*. España: Club BPM.
- Calderón, E. (2013). *Madurez y planificación estratégica de proyectos BPM en el sistema financiero peruano*. Lima: UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS.
- Carvajal, Á. (2002). *Teorías y Modelos: Formas de Representación de la Realidad*. Costa Rica: Revista Comunicación.
- Conlago, M. (2014). *Plan de Moejora para el proceso de Desarrollo de Software de Intellectus System S.A*. Sangolquí: ESPE.
- Farez, M. (2015). *MODELO DE CALIDAD PARA LA GESTIÓN DE PROCESOS DE DESARROLLO DE SOFTWARE APLICADO AL SISTEMA INFORMÁTICO EN USO DE LA UNIVERSIDAD TÉCNICA DE MACHALA*. Latacunga: ESPE.

- Franco, C. A. (2004). *CONDUCCIÓN DE REUNIONES PARA LA OPERACIÓN EFICAZ DE LOS EQUIPOS DE TRABAJO*. Cali, Colombia: Estudios Gerenciales.
- García, J. (2015). *Una propuesta para el uso del paradigma guiado por modelos (MDE) para la definición y ejecución de procesos de negocios*. Sevilla: Universidad de Sevilla.
- García, L. (2015). *Un marco de referencia para facilitar la interoperabilidad y mantenibilidad de los modelos de procesos de software*. Sevilla: Universidad de Sevilla.
- Hernández, R., Fernández, C., & Baptista, M. (2010). *METODOLOGÍA de la Investigación*. México: McGraw Hill INTERAMERICANA EDITORES, S.A. DE C.V.
- Hoyle, D. &. (2002). *Del aseguramiento a la gestión de la calidad: el enfoque basado en procesos*. Madrid: AENOR.
- Hurtado de Barrera, J. (2000). *Metodología de la Investigación Holística*. Caracas: Fundación Sygal.
- Institute, S. E. (2010). *CMMI para Desarrollo Version 1.3*. España: Editorial Universitaria Ramón Areces.
- Lapedra Alcam, R. (2011). *Introducción a la gestión de sistemas de información en la empresa*. Castelló de la Plana: UNIVERSITAT JAUME.
- Laudon, K., & Laudon, J. (2012). *SISTEMAS DE INFORMACIÓN GERENCIAL*. México: PEARSON EDUCACIÓN.
- Lemes Batista, A., & Machado Hernández, T. (2007). *SEGUNDO ENCUENTRO INTERNACIONAL SOBRE Las Medianas, Pequeñas y Micro-Empresas del Siglo XXI*. CUBA: EUMEDNET.
- Llanes-Font, M. (2013). *De la gestión por procesos a la gestión integrada por procesos*. Cuba: Universidad de Holguín.
- Macías García, M. (2007). *GUIA PARA LA IDENTIFICACIÓN Y ANÁLISIS GUIA PARA LA IDENTIFICACIÓN Y ANÁLISIS IDENTIFICACIÓN Y ANÁLISIS DE PROCESOS DE PROCESOS DE PROCESOS. V01*. Cadiz: Universidad de Cadiz.
- Marcos, J., Arroyo, A., Garzás, J., & Piattini, M. (2008). *La norma ISO/IEC 25000 y el proyecto KEMIS para su automatización con software libre*. Madrid: REICIS.
- Marín Sánchez, J. (2016). *Control de proyectos de software: actualidad y retos para la industria cubana*. La Habana: Ingeniare.
- Marín-Sánchez, J. (2014). *Proceso para la planificación y control de proyectos de software utilizando Xedro-GESPRO*. La Habana: Revista Cubana de Ciencias Informáticas.

- Martínez, A. (2012). *Propuesta Integral de un Modelo de Gestión por Procesos de negocio (PIM-GPN)*. México: UPUESA.
- Monje, C. (2011). *Metodología de la Investigación Cuantitativa y Cualitativa, Guía Didáctica*. Neiva: Universidad Surcolombiana del Docente.
- Orna, C. (2013). *Modelo de calidad para la etapa de planificación de Proyectos Medianos de Software*. Latacunga: ESPE.
- Palella, S., & Martins, F. (2012). *Metodología de la Investigación Cuantitativa*. Caracas: FEDUPEL.
- PMI, I. P. (2013). *Guía de los fundamentos para la dirección de proyectos (guía del PMBOK®)*. Pensilvania: GLOBALSTANDAR.
- Ponce, H. (2007). *LA MATRIZ FODA: ALTERNATIVA DE DIAGNÓSTICO Y DETERMINACIÓN DE ESTRATEGIAS DE INTERVENCIÓN EN DIVERSAS ORGANIZACIONES*. México: ENSEÑANZA E INVESTIGACIÓN EN PSICOLOGÍA .
- Pressman, R. S. (2010). *Ingeniería del software: un enfoque práctico*. México: McGRAW-HILL INTERAMERICANA EDITORES, S.A.
- Sabino, C. (1992). *EL PROCESO DE INVESTIGACION*. Buenos Aires: Panapo.
- Santamaría, P. (2012). *ESTUDIO PARA LA IMPLEMENTACIÓN DE ADMINISTRACIÓN DE PROCESOS DE NEGOCIO (BPM) EN LA FUERZA AÉREA COLOMBIANA*. Bogotá, Colombia.
- Sanz, A. (2012). *Marco Metodológico para la mejora de las actividades de Verificación y Validación de Productos Software*. Leganés: UNIVERSIDAD CARLOS III DE MADRID.
- Sommerville, I. (2005). *Ingeniería del Software*. Madrid: PEARSON EDUCACION S.A.
- Suárez Lorenzo, F., & Garzás Parra, J. (2014). *I Jornada sobre Calidad del Producto Software e ISO 25000*. Santiago de Compostela: XUNTA DE GALICIA.