UNIVERSIDAD CATÓLICA ANDRÉS BELLO
FACULTAD DE INGENIERIA
DIRECCIÓN DE POSTGRADO
POSTGRADO EN SISTEMAS DE INFORMACIÓN

TRABAJO ESPECIAL DE GRADO

DISEÑO DE UNA METODOLOGIA DE PRUEBAS EN EL CICLO DE DESARROLLO DEL SOFTWARE

Caso de Estudio: Mercantil Seguros

Trabajo Especial de Grado presentado como requisito parcial para al Título de Especialista en Sistemas de Información

Presentado por
Lic. Santana Segovia, Cecilia Josefina
para optar al título de
Especialista en Sistemas De Información

Asesor
MsC. María Esther Remedio

Caracas, Julio 2018

UNIVERSIDAD CATÓLICA ANDRÉS BELLO
FACULTAD DE INGENIERIA
DIRECCIÓN DE POSTGRADO
POSTGRADO EN SISTEMAS DE INFORMACIÓN

TRABAJO ESPECIAL DE GRADO

DISEÑO DE UNA METODOLOGIA DE PRUEBAS EN EL CICLO DE DESARROLLO DEL SOFTWARE

Caso de Estudio: Mercantil Seguros

Presentado por
Lic. Santana Segovia, Cecilia Josefina
para optar al título de
Especialista en Sistemas De Información

Asesor
MsC. María Esther Remedio

Caracas, Julio 2018

UNIVERSIDAD CATÓLICA ANDRÉS BELLO
Urb. Montalbán - La Vega - Apartado 20332
Telf.: (0212) 407-44-44 Fax 407-43-49
Estudios de Postgrado

ACTA DE EVALUACIÓN DE TRABAJO ESPECIAL DE GRADO ESPECIALIZACIÓN EN SISTEMAS DE INFORMACIÓN

Nosotros, Profesores MARÍA REMEDIOS DE LEGONIA (Asesor) y FANNY HERNÁNDEZ DE LÓPEZ, 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 Especial de Grado " DISEÑO DE UNA METODOLOGÍA DE PRUEBAS EN EL CICLO DE DESARROLLO DEL SOFTWARE. CASO DE ESTUDIO: MERCANTIL SEGUROS ", presentado por la ciudadana Santana Segovia, Cecilia Josefina, C.I. N°. 6328539., para optar al grado de Magister en Sistemas de Información.

Declaramos que:

Hemos leído el ejemplar del Trabajo Especial de Grado que nos fue entregado con anterioridad por la Dirección del Programa.

Reunidos a los veinticinco días del mes de julio del año dos mil dieciocho, en la sede de los Estudios de Postgrado de la Universidad Católica Andrés Bello, previa lectura y estudio del mencionado trabajo, hemos decidido convocar al estudiante con el fin de responder las preguntas que le formule el jurado. Hechas por nuestra parte las preguntas y aclaratorias correspondientes, se consideró formalizar el siguiente dictamen:

APROBADO

Hemos acordado calificar el Trabajo Especial de Grado con DIECINUEVE (19) 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 veinticinco días del mes de julio del año dos mil dieciocho.

María Remedios De Legonia
C.I.: 5.530.488

Fanny Hernández de López
C.I.: 7.958.174

UNIVERSIDAD CATÓLICA ANDRÉS BELLO
DIRECCIÓN DE POSTGRADO DE LA FACULTAD DE INGENIERÍA
MONTALBÁN

Caracas, 24 de Abril de 2018

Director del Programa Sistemas de Información
Estudios de Postgrado
Universidad Católica Andrés Bello (UCAB)
Presente.-

Referencia: Aprobación del Asesor

Tengo a bien dirigirme a Usted a fin de informarle que he leído y revisado el Trabajo Especial de Grado titulado: "DISEÑO DE UNA METODOLOGIA DE PRUEBAS EN EL CICLO DE DESARROLLO DEL SOFTWARE, Caso de Estudio: Mercantil Seguros", presentado por Cecilia Josefina Santana Segovia, portadora de la cédula de identidad N° 6.328.539, como parte de los requisitos para optar al Título de Especialista en Sistemas de Información.

A partir de dicha revisión, considero que el mencionado Trabajo Especial de Grado reúne los requisitos y méritos suficientes para ser sometido a evaluación por el distinguido jurado que se decida designar a tal fin.

En la ciudad de Caracas, a los 24 días del mes de abril de 2018

MsC. María Esther Remedios
C.I. N°. 5.530.488

Caracas, 10 de Abril de 2017

Sres.
UNIVERSIDAD CATÓLICA ANDRÉS BELLO
Postgrado en Sistemas de Información
Núcleo Caracas.

Nos dirigimos a ustedes para informarles que hemos autorizado a la Licenciada CECILIA JOSEFINA SANTANA SEGOVIA, C.I 6.328.539, quien labora en la organización Mercantil Seguros, Gerencia de Tecnología y Sistemas, a hacer uso de la información proveniente de esta organización, para documentar y soportar los elementos de los distintos análisis estrictamente académicos que conllevaran a la realización del trabajo especial de grado "DISEÑO DE UNA METODOLOGIA DE PRUEBAS EN EL CICLO DE DESARROLLO DEL SOFTWARE, Caso de Estudio: Mercantil Seguros", como requisito para optar al título de Especialista en Sistemas de Información, exigidos por la Dirección General de los Estudios de Postgrado de la Universidad Católica Andrés Bello.

Sin más que hacer referencia, atentamente,

P/

Gerente de Sistemas y Tecnología
Mercantil Seguros

Mercantil Seguros
Gerencia de Tecnología y Sistemas

UNIVERSIDAD CATÓLICA ANDRÉS BELLO
FACULTAD DE INGENIERIA
DIRECCIÓN DE POSTGRADO
POSTGRADO EN SISTEMAS DE INFORMACIÓN

DISEÑO DE UNA METODOLOGIA DE PRUEBAS EN EL CICLO DE DESARROLLO DEL SOFTWARE
Caso de Estudio: Mercantil Seguros

Autor: Cecilia Santana
Asesor: María Esther Remedios
Fecha: Julio 2018

RESUMEN

El ciclo de desarrollo del software describe la vida de un producto desde su etapa de concepción hasta su entrega, utilización y mantenimiento. En forma análoga, el proceso de pruebas define la forma en la cual dicho producto debe ser probado. Esta es una actividad clave dentro de un proyecto de software ya que determina el grado de cumplimiento del producto desarrollado con los requisitos establecidos para el sistema. En Mercantil Seguros se han visto en la necesidad de mejorar sus sistemas con la intención de prestar un mejor y oportuno servicio. Por esta necesidad es que se planteó desarrollar una investigación siguiendo la modalidad de proyecto factible con el objetivo de Diseñar una metodología de pruebas que pueda ser aplicada en el ciclo de desarrollo del software, basándose en normas y mejores prácticas, abarcando cuatro (4) objetivos específicos: elaboración de un modelo conceptual que soporte el trabajo de investigación, describir la situación actual del área de pruebas, identificación de técnicas y herramientas de prueba y elaboración de una guía metodológica de pruebas de software. Para recopilar la información se utilizaron técnicas documentales, la entrevista y el benchmarking. Esta guía metodológica permitirá mantener un servicio de apoyo en el desarrollo óptimo, lo cual dará garantía de los procesos en el menor tiempo, garantizando la mayor calidad de los productos que se desarrollan.

Palabras Clave: Ciclo de desarrollo de software, pruebas de software, Gerencia de Tecnología y Sistemas, metodología de pruebas.

Línea de investigación: Ingeniería del Software

INDICE

RESUMEN V
INDICE VI
LISTA DE FIGURAS VIII
LISTA DE TABLAS IX
INTRODUCCION 10
CAPITULO I 13
EL PROBLEMA DE INVESTIGACION 13
1.1. PLANTEAMIENTO DEL PROBLEMA 13
1.2. FORMULACIÓN DEL PROBLEMA 18
1.3. OBJETIVOS DE LA INVESTIGACIÓN 18
1.3.1. Objetivo general 18
1.3.2. Objetivos específicos 19
1.4. JUSTIFICACIÓN E IMPORTANCIA DE LA INVESTIGACIÓN 19
1.5. ALCANCE Y LIMITACIONES 20
CAPITULO II 22
MARCO TEORICO 22
2.1. ANTECEDENTES DE LA INVESTIGACIÓN 22
2.2. BASES TEÓRICAS 23
2.2.1. Calidad del Software 23
2.2.2. Pruebas de Software 57
2.2.3. Modelos para la Mejora de las Pruebas 78
2.2.4. Las Pruebas de Software basado en el Proceso de Desarrollo 93
2.3. BASES LEGALES Y ÉTICAS 99
CAPITULO III 101
MARCO METODOLOGICO 101
3.1. NIVEL Y DISEÑO DE LA INVESTIGACIÓN 101
3.1.1. Nivel de Investigación 101

3.1.2. Diseño de la Investigación 102
3.2. POBLACIÓN Y MUESTRA 104
3.3. TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS 104
3.4. LAS VARIABLES, SUS DIMENSIONES E INDICADORES 106
3.4.1. Operacionalización de variables 107
3.5. TÉCNICAS DE PROCESAMIENTO Y ANÁLISIS DE DATOS 109
3.6. FASES DE LA INVESTIGACIÓN 111
CAPITULO IV 115
MARCO ORGANIZACIONAL 115
4.1. HISTORIA DE MERCANTIL SEGUROS 115
4.2. MISIÓN 116
4.3. VISIÓN 116
4.4. ESTRUCTURA ORGANIZATIVA 116
4.4.1. Junta Directiva Mercantil Seguros 116
4.4.2. Organigrama 117
CAPITULO V 120
ASPECTOS ADMINISTRATIVOS DEL PROYECTO 120
5.1. RECURSOS NECESARIOS 120
5.2. PRESUPUESTO ESTIMADO 120
5.3. CRONOGRAMA DE ACTIVIDADES 121
CAPITULO VI 122
DESARROLLO DE LA PROPUESTA 122
CAPÍTULO VII 149
CONCLUSIONES Y RECOMENDACIONES 149
BIBLIOGRAFIA 152
ANEXOS 158

LISTA DE FIGURAS

Figura 1. Áreas de los Procesos de ITIL (Scalone, 2006) 46

Figura 2. Calidad en el ciclo de vida según ISO/IEC 9126-1 52

Figura 3 Estructura de cada nivel de madurez (Santana Cecilia, 2018) 79

Figura 4. Estructura y Objetivos de Madurez para TMM (Santana Cecilia, 2018) 79

Figura 5. Áreas y Objetivos por Nivel de Madurez (Sanz Esteban, 2011) 81

Figura 6. Niveles de TMMI (Sanz Esteban, 2011) 82

Figura 7. Estructuras de las áreas definidas para TMMI (Santana Cecilia, 2018) 84

Figura 8. Matriz de Madurez de TPI (Sanz Esteban, 2011) 86

Figura 9. Elementos de Test Management Approach (Sanz, Saldaña, Javier,&Gaitero,2008) 87

Figura 10. Ciclo de Vida de TMap (Sogeti, 2013) 89

Figura 11. Objetivos y Prácticas Específicas de TestPAI (Sanz, Saldaña, Javier, &Gaitero, 2008). 93

Figura 12. Fases y Disciplinas según el Proceso Unificado. (Jacobson, Booch, & Rumbaugh, 2000) 95

Figura 13. Resumen Componentes y Fases de Proceso Unificado (UP) (Torossi, 2012) 96

Figura 14. Disciplina de Prueba.(Jaconson, Booch, & Rumbaugh, 2000) 97

Figura 15. Estructura Desagregada de Trabajo (Santana Cecilia, 2018) 112

Figura 16. Organigrama Mercantil Seguros (Santana Cecilia, 2018) 118

Figura 17. Organigrama Gerencia de Sistemas y Tecnología de Mercantil Seguros (Santana Cecilia, 2018) 119

Figura 18 Modelo Conceptual asociado a los casos de prueba (Méndez, Pérez, & Mendoza, 2007) 123

Figura 19 Modelo Conceptual asociado a los tipos, niveles y técnicas de Pruebas 124

Figura 20 Modelo Conceptual asociado a los Modelos de Desarrollo de Software y de Pruebas .. 124

Figura 21 Modelo Conceptual Propuesto 125

LISTA DE TABLAS

Tabla 1. Modelos y Estándares Nivel de Calidad de Procesos..... 33

Tabla 2. Niveles de Madurez y de capacidad (Parada, 2010) 34

Tabla 3. Modelos y Estándares a Nivel de Calidad del Producto 47

Tabla 4. Áreas de trabajo del Modelo de Mc Call..... 50

Tabla 5 Modelos del Flujo de Trabajo UP 95

Tabla 6 Operacionalización de Variables - Objetivo Específico Nro 1 108

Tabla 7 Operacionalización de variables - Objetivo Específico Nro. 3 108

Tabla 8 Operacionalización de Variables - Objetivo Específico Nro. 4 108

Tabla 9 Operacionalización de Variable - Objetivo Específico Nro. 5..... 109

Tabla 10. Junta Directiva Mercantil Seguros 116

Tabla 11. Presupuesto tentativo del trabajo de investigación..... 120

INTRODUCCION

Los sistemas de información iniciaron su camino en las empresas en la década de los noventa con el fin de cerrar la brecha entre administradores, gerentes, ejecutivos, ingenieros y técnicos en informática, de manera que el conocimiento de los segundos pueda facilitar y enriquecer el trabajo de los primeros**.** El uso de los sistemas de información debe ser parte de la labor gerencial, con la cual se busca facilitar la información para el seguimiento de los procesos y el planteamiento de nuevas estrategias en todos los campos. Se debe introducir la informática en la toma de decisiones (acción gerencial), lo cual requiere una actitud positiva hacia la tecnología para aplicar cambios, tanto a nivel de los equipos como la estructura de la organización, sus procesos y procedimientos.

El propósito de la ingeniería del software es generar y mantener sistemas de software dentro de las restricciones de tiempo, funcionalidad y costos acordados con el cliente. Las metas de esta disciplina tecnológica son mejorar la calidad de los productos desarrollados y aumentar la productividad de los ingenieros de software. Existen diferentes modelos, pero podemos encontrar una serie de actividades que son comunes en todos, ellas son: (i) Especificación, (ii) Diseño, (iii) Validación y (iv) Evolución. En la actividad de Validación están presentes las Pruebas, lo cual implica el preparar una estrategia de pruebas, integrando los métodos de diseño de casos de prueba del software en una serie bien planeada de pasos que desembocará en la eficaz construcción de software.

En Venezuela las compañías de Seguros están operando en un entorno económico adverso, por lo que las tendencias del mercado en general han obligado a cambiar sus estrategias competitivas. Se busca ganar la confianza de los clientes y así establecer vínculos comerciales que garanticen su fidelidad. Siempre se estará tratando de aumentar la “cartera” de clientes pero a diferencia de años anteriores, ahora se trabaja en mantenerla. Por ello, el consumidor de hoy tiene exigencias a la hora de considerar una empresa, ya que necesita ser tomado en

serio, que le ofrezcan información, respuestas inmediatas, reconocimiento, atención personalizada, ofertas diferenciales y además que sean transparentes y honestas con él.

Por otro lado la competencia en el sector es agresiva, cada empresa aseguradora busca un posicionamiento firme que le permita desarrollar su actividad de forma progresiva y estable, donde cada empresa de seguros aplica su propia estrategia para mantenerse en el mercado y poder ser competitiva. Existen 49 empresas de Seguros y Mercantil Seguros está ubicada dentro de las dos primeras empresas aseguradoras del país. Para mejorar esta posición, es una necesidad aplicar soluciones integrales, innovadoras y adecuadas en defensa de su patrimonio y servicios. Su estabilidad gira en pro de su mejora continua, para poder cumplir satisfactoriamente con los compromisos hacia la comunidad asegurada y protegiendo así los intereses presentes y futuros.

En ese sentido, con el propósito de colaborar con el desarrollo tecnológico, se espera brindar una adecuada actividad de pruebas en el ciclo de desarrollo del software de Mercantil Seguros, para así mejorar los tiempos de entrega de los productos y optimizar las planificaciones. Para ello se realizó el presente trabajo de investigación, en el cual se diseñó una metodología de pruebas dentro del ciclo de desarrollo del software, la cual será utilizada por la Gerencia de Tecnología y Sistemas de Mercantil Seguros.

La presente investigación es un diseño mixto (documental y de campo) de tipo proyecto factible, ya que recoge la información que se maneja en la Gerencia de Tecnología y Sistemas aunado al conocimiento y experiencia de la autora; para esto se utilizaron las técnicas de análisis documental y análisis de contenido. Los instrumentos utilizados en la presente investigación se describen en el correspondiente capitulo.

A continuación se describe el contenido y propósito de cada uno de los capítulos que conforman esta investigación:

El Capítulo I “El Problemas de Investigación” presenta el planteamiento y formulación de la investigación así como su objetivo general y objetivos específicos, justificando la investigación e indicando sus limitaciones. Además presenta su alcance y limitaciones.

El Capítulo II “Marco Teórico” explica de manera detallada los antecedentes de la investigación conformado por tesis doctorales, de maestría y tesis de especialización, así como las bases teóricas legales y ética de la investigación.

El Capítulo III “Marco Metodológico” describió aspectos que permiten el desarrollo de la investigación, tales como tipo y diseño de investigación, población de estudio, operacionalización de objetivos, técnicas e instrumentos para levantamiento, procesamiento y análisis de datos y las fases de la investigación.

El Capítulo IV “Marco Organizacional” presenta toda la información correspondiente a la empresa en la cual se realizó la investigación: historia, misión, visión y estructura organizativa.

El Capítulo V “Aspectos Administrativos del Proyecto” presenta los recursos necesarios, el presupuesto estimado y el cronograma de trabajo para el desarrollo de la investigación.

El Capítulo VI “Desarrollo de Objetivos de la Investigación” siendo la parte central de la investigación ya que presenta de manera desglosada las actividades realizadas para alcanzar los objetivos de la investigación.

El Capítulo VII “Conclusiones y Recomendaciones” describe las conclusiones obtenidas al finalizar el proceso investigativo así como las recomendaciones dadas a fin de implementar con éxito la guía metodológica de pruebas.

Finalmente se presentan referencias y anexos utilizados en el desarrollo de la investigación.

CAPITULO I

EL PROBLEMA DE INVESTIGACION

El presente capítulo tiene como objeto desarrollar el contexto en el cual se ubica el problema, las interrogantes de la investigación, los objetivos del trabajo, la importancia o justificación del estudio y las limitaciones o alcance del trabajo de investigación.

1.1. Planteamiento del Problema

En el mundo en que vivimos, el desarrollo de herramientas tecnológicas se hace presente cada vez más en nuestras actividades, siendo las aplicaciones informáticas una de ellas por lo que su uso trascendental nos conlleva a imaginarnos lo difícil que sería prescindir de ellas.

Actualmente, en esta época de globalización, donde las nuevas tecnologías han convertido todo el planeta en un único y gigantesco mercado que funciona permanentemente; la calidad de los productos o servicios se convierte en un elemento diferenciador y, en gran medida, en la clave principal del éxito o fracaso de toda organización.

El software está en todo lugar, está presente en casi cada aspecto y momento de la vida de las personas. Algunas veces el software es obvio, pero en otras no. Cuando se está en un banco o se utiliza un cajero automático, se puede observar el funcionamiento de los sistemas del software, pero cuando se maneja un auto moderno, es fácil de olvidar que hay más poder computacional y presencia de software de los autos que en la mayoría de las naves espaciales construidas hasta la década de los ochentas (Rueda Sandoval & Black, 2011).

Por ello es fácil de olvidar una experiencia cuando un software funciona, mientras que se recuerda claramente uno que no funcionó, producto de una falla o defecto. Para el caso de una compañía desarrolladora de software el mal funcionamiento de su producto los lleva a sufrir daños en su reputación y en consecuencia falta de confianza por parte de los interesados del

negocio. Este resultado se repite en el caso que los clientes sean empleados internos de una compañía. En los estándares de Basilea II se considera esta situación tanto como un riesgo tecnológico o como un riesgo reputacional, que se debe minimizar.

Para los individuos, las sociedades y los estados existen amenazas puestas por los defectos. Las personas pueden perder sus trabajos cuando las fallas en un software resultan en un revés económico para su empleador. En algunos casos las personas pueden perder sus vidas debido al software, esto lo podemos observar en las aerolíneas. También puede perder sus derechos civiles a causa de una falla, cuando las máquinas de elecciones computarizadas han fracasado.

Puede decirse que muchas personas han sufrido la pérdida de la privacidad, el robo de la identidad, debido a que el software usado por sus bancos ha tenido defectos. Misiones o incluso guerras podrían ser perdida debido a los defectos a medida que el software se ha hecho más frecuente en los entornos de los campos de batallas (Rueda & Black, 2011).

Por tal motivo, se considera importante que las empresas se reorganicen de tal forma que den garantía a sus clientes y usuarios, siendo un factor imprescindible la capacidad que tengan para satisfacer de manera integral los requerimientos del cliente, debido a que la competitividad de una empresa y la satisfacción del cliente están determinadas por la calidad del producto, el precio y la calidad del servicio; es decir, se es más competitivo si se puede ofrecer mejor calidad, a bajo precio y en menor tiempo (Guajardo, 1996). Sin embargo, para concebir dicho concepto, es preciso que las organizaciones mejoren cada vez más sus procesos de manera que logren la excelencia, garantizando así la satisfacción total del cliente; no obstante, ha sido necesaria la evolución del significado dado a la palabra calidad a lo largo del tiempo para lograrlo, involucrando un avance incremental desde la inspección masiva al control de calidad, que utilizaba métodos estadísticos (James, 1997), hasta un salto cualitativo importante cuando se pasa a los enfoques de aseguramiento de la calidad (AC) y de gestión de calidad total (GCT), en los que el énfasis se pone en la prevención de los errores y la mejora de los procesos para evitar que los reproceso ocurran (Moreno, 2001).

Se denota por las citas anteriores que el software contiene defectos y es aquí donde se hacen las siguientes interrogantes: ¿de dónde vienen estos defectos y qué es lo que hacen? Los

defectos no son puestos en los sistemas por pequeñas hadas que vuelan alrededor de los centros de datos y lanzan los defectos en las computadoras y en el código. Los defectos están en el sistema porque alguien los coloco allí; Alguien cometió un error, una equivocación, un descuido, cuyo resultado fue la inserción de un defecto o "bug" en el sistema.

Un defecto o un error pueden ser insertados en cualquier momento durante el ciclo de vida del desarrollo del software:

a) En los requerimientos

b) En las especificaciones de diseño

c) En el código, ya sea en la lógica de negocios o en la interfaz de usuario

d) En la documentación, ya sea en la documentación electrónica o impresa.

En relación a ello (Rueda & Black, 2011, pág. 34) afirman: … *"para que una falla ocurra, la parte defectuosas del sistema tiene que ser ejecutada, en ese momento el defecto podría provocar que el sistema funcione mal. Asumiendo que las precondiciones correctas están vigentes, el sistema podría fallar en hacer lo que debería hacer bajo esas condiciones. Si esa falla es visible o llega a ser visible después, ya sea a un cliente, a un usuario o a algún otro interesado del negocio, ese comportamiento podría resultar en la insatisfacción de la calidad del sistema*".

Además, se tienen muchos sistemas interactuando de punta a punta, donde una transacción tiene que ser manipulada por un número de sistemas ante que la transacción se complete correctamente. Cada operación podría ser simple, pero la secuencia total de operaciones puede llegar a ser compleja, y esa complejidad puede conducir a malas expectativas de parte de una o más personas. Esto conduce a un defecto en la interacción de operaciones o en secuencia de ellos.

Así mismo, habiendo considerado los defectos y las fallas, se denota que el sistema está sujeto a varios riesgos. Los mismos están relacionados ampliamente a la calidad en un sentido amplio de la cual podría decir: su implementación, la versión, la restricción de los recursos y las pruebas. Las pruebas no gestionan mucho estas áreas de riesgos, sino más bien son

afectadas por estos riesgos. Sin embargo cuando se piensa reducir las fallas, es cuando se acerca el momento de las pruebas como un medio de gestionar los riesgos de calidad. Las pruebas hacen esto de varias formas:

Una forma es proporcionar la información para guiar el proyecto en el área de riesgo de calidad y otra es enfocarse en las pruebas acerca de los riesgos de calidad más importantes. Aun otra es de localizar y dejar a las personas que reparen los problemas importantes. Si bien es cierto que una es "apta para el uso". La otra es "conformidad con los requisitos".

A causa de esto, el autor antes mencionado expresa una reflexión literal, tomada del diccionario Español Sopena, sobre la palabra Prueba, la cual indica: ... *Examen o experimentación para comprobar el buen funcionamiento de alguna cosa o su adecuación a un determinado fin, Demostración de la verdad de alguna cosa, o de su existencia.* (Rueda & Black, 2011,pág 55).

Con lo anterior claramente definimos que "apta para el uso" implica que aquellos que usan, emplean, adquieren o necesitan el software son aquellos que deberían determinar si tiene o no calidad. La definición "conformidad con los requisitos", por el contrario, necesita simplemente que el sistema trabaje de la manera especificada en algún documento.

En muchos casos, las pruebas tienen uno o más objetivos generales: podrían ser llevadas a cabo para encontrar defectos, y luego proporcionar a los programadores la información que ellos necesitan para corregir estos defectos. También ser llevadas a cabo para dar a la gente confianza en el nivel de calidad del sistema.

Es importante la detección de defectos y fallas en tiempos tempranas del desarrollo de software, a través de la ejecución de pruebas. Hay que tener claro la relación y diferencia entre las pruebas, el aseguramiento y mejoramiento de la calidad. Los grupos de pruebas son comúnmente referidos como grupos de "aseguramiento de la calidad" o grupos de "QA". Mientras que las pruebas pueden y deberían ser parte de una estrategia más grande del aseguramiento de la calidad, las pruebas por si solas no aseguran la calidad, pero considero que son importantes para asegurar la calidad. Sólo con la atención diligente a la calidad en todo el ciclo de vida del software y en toda la organización podemos asegurar la calidad.

Mercantil Seguros no escapa a esta situación. La compañía tiene como misión ser la mejor aseguradora del país, ofrecer una gran variedad de productos y servicios de alta calidad al mejor costo, al igual que ofrecer eficientes respuestas a sus clientes, proveedores y productores.

Para lograr esta misión, la compañía se ha visto en la necesidad de realizar ajustes a su Sistema, el cual es llamado RECTOR, el cual maneja el ochenta por ciento de las operaciones del negocio a nivel nacional. Estos ajustes a nivel de sistemas son ejecutados a través de la Gerencia de Tecnología y Sistemas, la cual cuenta con personal altamente especializado.

En la revisión de los proyectos planificados versus los ejecutados por la Gerencia de Tecnología y Sistemas, se puede observar que las fechas de entrega no han sido cumplidas en un gran número de ellos. La presión de tiempo está presente en el desarrollo de cada proyecto de la compañía, por lo cual al momento de retrasar la entrega de los desarrollos trae como consecuencia el descontento por parte de los usuarios finales y en ocasiones fallas en producción de estos nuevos desarrollos.

La Gerencia de Tecnología y Sistemas cuenta con un Departamento de Pruebas y Certificación, el cual se encarga de realizar la actividad de pruebas a los desarrollos de software internos y también a los que han sido adquiridos por la compañía. Cuentan con instrumentos de pruebas, sistema para el registro de defectos y se cuenta con una metodología de trabajo.

Los integrantes de dicho departamento se han encontrado con proyectos, que al momento de comenzar con las pruebas de aceptación reportan defectos y fallas, los cuales no deben de estar presentes en este momento de la revisión. Esto trae como consecuencia que los proyectos tengan que regresar nuevamente a la etapa de programación, con lo cual se incumplen los tiempos de entrega de los desarrollos, afectando a los usuarios finales.

Es por eso que este trabajo buscó la problemática que se presenta en la detección de defectos y fallas dentro del desarrollo de software, la aplicación de pruebas adecuadas en cada etapa de desarrollo; validar los métodos y mejores prácticas existentes en cuanto a ellas, permita la

detección de defectos de manera oportuna, disminuyendo las incidencias una vez que se encuentren en producción los nuevos desarrollos; la disminución de desviaciones en los planes de entrega por causa de detección fallas y defectos en etapa de aceptación, lo cual es de gran importancia para recobrar la confianza de las áreas de negocio y poder ayudar a la compañía a lograr sus objetivos estratégicos.

1.2. Formulación del Problema

De acuerdo al problema planteado el autor de la presente investigación traza la siguiente interrogante principal: ¿Cuál es la metodología de pruebas que resuelve la situación de la Gerencia de Tecnología y Sistemas de Mercantil Seguros?

Así mismo, plantea las siguientes interrogantes secundarias, a saber: ¿Cuáles normas, mejores prácticas y estándares internacionales de pruebas pueden servir de guía para la elaboración de la propuesta? ¿Cuáles de las prácticas actuales del Departamento de Pruebas y Certificación deben permanecer o ser mejoradas? ¿Qué técnicas y herramientas deben ser implementadas para mejorar la actividad de pruebas? ¿Qué procedimientos deben ser incorporados para mejorar la actividad de pruebas?

1.3. Objetivos de la Investigación

Arias (Arias, 2006) indica que el objetivo de la investigación es un enunciado que expresa lo que se desea indagar y conocer para responder a un problema planteado. Bernal (2006) señala que el objetivo general debe reflejar la esencia del planteamiento del problema y la idea expresada en el título del proyecto de investigación y que los objetivos específicos se desprenden del general y debe ser formulados de forma que estén orientados al logro del objetivo general.

1.3.1. Objetivo general

Diseñar una metodología de pruebas que apoye el ciclo de desarrollo del software adaptado a la Gerencia de Tecnología y Sistemas de Mercantil Seguros, basado en normas y mejores prácticas de Ingeniería del Software en la disciplina de pruebas.

1.3.2. Objetivos específicos

1.3.2.1. Elaborar un modelo conceptual que soporte el trabajo de investigación, basado en normas y mejores prácticas de Ingeniería del Software en la disciplina de pruebas.

1.3.2.2. Describir la situación actual de pruebas en la Gerencia de Tecnología y Sistemas de Mercantil Seguros.

1.3.2.3. Identificar las técnicas y herramientas relativas a la disciplina de pruebas que pueden ser aplicados al problema de estudio, sustentado en el marco teórico de esta investigación.

1.3.2.4. Elaborar una guía metodológica de prueba de software, basado en normas y mejores prácticas de Ingeniería del Software en la disciplina de pruebas, previamente seleccionadas.

1.4. Justificación e Importancia de la Investigación

Según Hernández (Hernández, Fernández, & Baptista, 2010) la justificación de la investigación comprende la exposición de las razones, es decir, el por qué y para qué del estudio, demostrando así la importancia y la necesidad de la investigación.

El proyecto planteado realiza un estudio de todo lo concerniente a la disciplina de Pruebas del software en el ciclo de desarrollo, dando importancia al aseguramiento de la Calidad del software como promotor y garante de calidad en el proceso de desarrollo del software.

Desde el punto de vista social, la empresa involucrada en este estudio pertenece al sector privado y pretende ser la mejor aseguradora del país, además ofrece una gran variedad de productos y servicios de alta calidad al mejor costo, al igual que eficientes respuestas a sus clientes, proveedores y productores.

Para la investigación se tomó como base de estudio la Gerencia de Tecnología y Sistemas de Mercantil Seguros, profundizando en los factores que logren alcanzar mayor eficiencia, que permita tomar decisiones acertadas y oportunas, así como los requerimientos generales del mismo.

En ese marco, el empleo de una información veraz, oportuna y exacta permite a cualquier organización una toma de decisiones más acertada y por consiguiente, el éxito de la gerencia, tomando en consideración que los sistemas de software son elementos determinantes para el logro de la eficiencia en las organizaciones modernas.

La Ingeniería del Software abarca actividades macro de suma importancia en el desarrollo del mismo, dentro de la que se encuentran las pruebas del software (Pressman, 2010), la cual vela, entre otros aspectos, porque exista la calidad esperada en el producto resultante, esto es, el fiel cumplimiento de los requerimientos acordados por el cliente.

Por lo anteriormente expuesto se comprueba lo valioso del estudio de la disciplina de prueba en el ciclo de desarrollo del software, la cual reside en la necesidad de detectar las fallas, defectos y sus riesgos en etapas tempranas, para con ello poder contribuir en el desarrollo de software con alta calidad. Con esta acción se espera asegurar la calidad y disminuir las entregas a destiempo de los proyectos motivado a fallas y defectos en etapas finales de su desarrollo, lo cual afecta a las áreas de negocios en alcanzar sus objetivos estratégicos y así cumplir con sus clientes, proveedores y asesores de seguros.

Por último, al realizar una investigación sobre la ejecución de un proceso, sus resultados muy bien pueden ser utilizados en otras investigaciones, como insumo básico para proponer o mejorar metodologías sobre las pruebas, fundamentadas en normas, políticas y procedimientos, apoyado en sistemas de información, interrelacionados y vinculados con las estrategias organizacionales, contribuyendo así con la amplitud de los conocimientos.

1.5. Alcance y Limitaciones

La propuesta de Trabajo de Especial de Grado tiene como alcance el diseño de una metodología para la disciplina de pruebas en el desarrollo del software desarrollados por la Gerencia de Tecnología y Sistemas de Mercantil Seguros.

Para el desarrollo de la propuesta se contó con acceso a la información de la Gerencia de Tecnología y Sistemas de Mercantil Seguros, con la cual se presentó la situación actual de dicha gerencia.

Para la realización del marco teórico se contó principalmente con el uso de Internet, libros, bases de datos, publicaciones, revistas, trabajos de grado y material de cursos.

Los recursos financieros y materiales utilizados están relacionados al desarrollo del documento final, reproducción del material, encuadernación, unidades de crédito y al tiempo que se invierta en la fase de diagnóstico, en el levantamiento de información y análisis de los resultados.

En cuanto a las limitaciones tenemos:

- El tiempo de dedicación del autor para esta investigación.
- El tiempo de atención para esta investigación por parte del personal de la Gerencia de Tecnología y Sistemas de Mercantil Seguros
- Solo se contó para la evaluación, con la información de dos proyectos de la gerencia.
- No se tiene acceso a la información de carácter confidencial.

CAPITULO II

MARCO TEORICO

El presente capítulo comprende aspectos teóricos y conceptuales del objeto de la investigación. El marco teórico está compuesto por: a) los antecedentes, se refieren a la revisión de trabajos previos sobre el tema en estudio, pueden ser tesis, trabajos de ascenso, resultados de investigaciones, ponencias, conferencias, congresos, entre otro; b) los fundamentos teóricos representan aquellos enfoques o corrientes desarrolladas por autores sobre el tema tratado en la investigación.

2.1. Antecedentes de la investigación

Los antecedentes reflejan los avances y el estado actual del conocimiento en un área determinada y sirven de modelo o ejemplo para futuras investigaciones. (Arias, 2006)

Como antecedentes de la investigación se incluyen los siguientes:

Tesis doctoral realizada por Hugo Parada (Parada, 2010), titulada: Contribución a los procesos de prueba de software y servicios. En ella se plantea el diseño e implementación de un marco metodológico para la gestión de proyectos de prueba, soportado por un conjunto integrado de herramientas que faciliten su aplicación al ciclo de desarrollo.

Este trabajo sirve de apoyo para el desarrollo de las bases teóricas y para formular la propuesta metodológica de la presente investigación.

También se toma como referencia la investigación de María Mendoza (Mendoza, 2010) para optar al título de Magister en Sistemas de Información, titulado: Modelo de referencia para la selección de herramientas de pruebas como soporte al proceso de desarrollo de software en PYMES venezolanas, donde se plantea el dar formalidad a la disciplina de pruebas como apoyo al proceso de desarrollo de software y el uso de herramientas automatizadas a PYMES venezolanas, por su importancia en el desarrollo del software.

Igualmente se hace referencia a la tesis de la licenciada Fernanda Scalone (Scalone, 2006) titulada: Estudio comparativo de los modelos y estándares de calidad del software, realizada

en Buenos Aires, Argentina, en la cual se encuentran los diferentes modelos y estándares de calidad del software, realizando una comparación y sus recomendaciones en cuanto a ellos.

El trabajo mencionado anteriormente nos sirve de apoyo para las bases teóricas de la presente investigación y la formulación de una propuesta metodológicas de pruebas.

Las investigaciones mencionadas anteriormente contribuyen para el cumplimiento de los objetivos.

2.2. Bases Teóricas

Según (Arias, 2006), las bases teóricas implican un desarrollo amplio de los conceptos y proposiciones que conforman el punto de vista o enfoque adoptado, para sustentar o explicar el problema planteado.

Para sustentar teóricamente la presente investigación es necesario tener conocimientos vinculados al Proceso Unificado de desarrollo de software (UP), a los modelos de calidad, el área de aseguramiento de la calidad, los diferentes tipos de pruebas, normas y estándares de prueba, mejores práctica, técnica y herramientas.

2.2.1. Calidad del Software

Las organizaciones a nivel mundial, persiguen conseguir un alto nivel de calidad en sus productos, lo cual se traduce en beneficios. A este concepto no escapan las industrias del software, ya que la modernidad de los negocios las hacen el soporte en su desarrollo, por lo que tienen un significativo valor en la sociedad.

En cuanto a la definición de calidad del software Mendoza (Mendoza, 2010) presenta las siguientes:

a) la real academia española define calidad como la propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor.

b) Según la ASQ (American Society for Quality, por sus siglas en inglés) plantea que en el sentido técnico de la calidad tiene dos (2) significados: (1) Las características de un

producto o servicio que refieren su capacidad de satisfacer lo indicado o las necesidades implícitas; (2) Un producto o servicio libre de deficiencias.

c) Para (Crosby, 1980) la definición de calidad la ve como la conformidad con los requisitos, es decir, gerenciar la calidad es una manera sistemática de garantizar que las actividades organizadas suceden en la manera que se planean.

Así mismo, (Mendoza, 2010) indica que todos estos conceptos están inmersos en el ámbito de la legislación Venezolana, donde la Ley del sistema Venezolano para la Calidad (Asamblea Nacional, 2002) en su Capítulo I, Art. 4, define calidad como el "Grado en que un conjunto de características inherentes a bienes y servicios cumple con unas necesidades o expectativas establecidas, generalmente implícitas u obligatorias (requisitos)". Todo esto permite establecer que los elementos presentes en la calidad son: características, cumplimiento/satisfacción, requisitos. Esto quiere decir que siempre estarán presentes expectativas que deben ser cubiertas, de acuerdo a las necesidades expresadas (directa o indirectamente) por los clientes y/o usuarios del producto o servicio resultante. A su vez, el grado en que son cubiertas, determinará qué tan bueno o malo es el producto resultante (mientras más necesidades sean cubiertas será visto como de mejor calidad).

En ese orden de idea, (Rueda & Black, 2011) también señala que según Unhelkar (Unhelkar, 2003) de una manera más formal, establece los ingredientes esenciales en la creación de "buena" calidad, describiéndolos de la siguiente manera: 1.-Un producto, 2.- Un proceso de producción (el proceso o método usado en la creación del producto es tan importante como el producto mismo), 3.- Un mecanismo de comunicación (para comunicarse con el dominio del desarrollo y a través de la industria –usuarios, productos, procesos), 4.- Una mejora continua de los ingredientes anteriores (es un gran esfuerzo producir un buen producto de calidad, mejorar lo que se logra es importante).

A continuación se indicarán conceptos de calidad de software: Según el IEEE (IEEE, 1990) en su estándar 610.12-1990, la calidad de software es definida como "el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o

expectativas del cliente o usuario. Aquí los elementos presentes en el concepto de calidad del software son: requisitos, necesidades y expectativas".

La "Guide to the Software Engineering Body of Knowledge", conocida como SWEBOK (IEEE, 2004), afirma que la importancia de este concepto es que los requerimientos del software definen las características de calidad requeridas del software e influencian los métodos y criterios de aceptación para determinar esas características. Encontramos que esta definición, al igual que la de UP, hacer particular atención a los métodos y criterios de aceptación que se determinan en el proceso de calidad del software y adicionalmente establecen con claridad los requerimientos de software, ya que los mismos serán la guía para determinar la calidad del mismo.

Como recomienda (Scalone, 2006) revela que la calidad del software es "la concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente". La Calidad del Software (CS) es una disciplina más dentro de la Ingeniería del Software. El principal instrumento para garantizar la calidad de las aplicaciones sigue siendo el Plan de Calidad, el cual se basa en normas o estándares genéricos y en procedimientos particulares. Los procedimientos pueden variar en cada organización, pero lo importante es que estén escritos, personalizados, adaptados a los procesos de la organización y que se sean cumplidos.

Por otro lado Mendoza (Mendoza, 2010) indica citando a Sommerville quien afirma que calidad del software significa, sencillamente que un producto debe resolver/cubrir su especificación. Sin embargo al hablar sobre la administración de la calidad del software, presenta algunos problemas y/o realidades de los sistemas:

- Hay una tensión entre los requisitos de calidad del cliente (eficacia, confiabilidad, etc.) y los requisitos de calidad del desarrollador (capacidad de mantenimiento, reutilidad, etc.);
- Algunos requisitos de calidad son difíciles de especificar de forma inequívoca;

- Las especificaciones del software son generalmente incompletas y a menudo contrarias.

Por lo indicado anteriormente, el autor está de acuerdo en que al aplicar calidad del software se permitirá obtener calidad, siempre que las actividades se ejecuten en los tiempos indicados, a pesar de las especificaciones de requerimiento, ya que éstas presentar imperfecciones.

Adicional a estas variadas opciones, se tiene presente también la de Pressman (Pressman, 2005) la cual es más explícita e incluye propios conceptos de software, pero la cual, según el propio autor, puede ser ampliada o modificada con todos los conceptos presentados, hasta lograr una definición más clara: "Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados, y con las características implícitas que se espera de todo software desarrollado profesionalmente". De esta definición, se pueden observar tres puntos importantes que deben tomarse en consideración, como lo expresa Pressman (Pressman, 2005):

- Los requisitos de software (explícitos) son la base de las medidas de la calidad. De hecho, una falta de correspondencia entre estos requerimientos, ya da por sentado una falta de calidad importante.
- Existen un conjunto de estándares especificados constituidos por un conjunto de criterios de desarrollo que guiarán la mejor manera de hacer las cosas, y cuya ausencia se traducirá en falta de calidad.
- Existe otro tipo de requisito de software que normalmente no se mencionan formalmente (implícitos), los cuales deben tomarse en cuenta en unión con los requisitos explícitos para que no se pueda dudar de la calidad del software que se construye.

Scalone (Scalone, 2006) menciona que la Calidad del Software debe implementarse en todo el ciclo de vida del mismo. Las distintas actividades para la implantación del control de calidad en el desarrollo de software son: (1) Aplicación de metodología y técnicas de desarrollo, (2) Reutilización de procesos de revisión formales, (3) Prueba del software, (4) Ajustes a los

estándares de desarrollo, (5) Control de cambios, mediciones y recopilación de información; y (6) Gestión de informes sobre el control de calidad

Mendoza (Mendoza, 2010) señala que en todos estos conceptos lo que se quiere tratar de minimizar es la variación que pueda existir entre las expectativas de los clientes y los resultados que emita el producto final (software). Controlar esta variación es la clave de un producto de alta calidad, y se hace controlando desde el proceso en sí, los recursos utilizados y los atributos de calidad que deberían caracterizar el software final. Por lo anterior, estaremos basando esta investigación en la definición de Pressman, por ser la que engloba los elementos que dirigen la disciplina de pruebas.

A pesar de lo anterior Kruchten (Kruchten, 2000) afirma que diferentes proyectos de desarrollo del software fallan de diversas formas, siendo posible identificar algunos problemas que se presentan como síntomas latentes en organizaciones que buscan implementarlos:

- Comprensión inexacta de las necesidades de los usuarios finales.
- Las necesidades del usuario o la empresa no se cumplen.
- Procesos de software normalmente improvisados.
- Los requerimientos cambian rápidamente.
- Los módulos no se integran.
- Difícil de darle mantenimiento.
- Baja calidad o experiencia del usuario final.
- Descubrimiento tardío de fallas.
- Pobre calidad del software.
- Cada miembro de los equipos hacen todo a su manera, haciendo imposible reconstruir quién cambio qué, cuándo, dónde y por qué.
- Desempeño deficiente bajo carga.
- Problemas de construcción y liberación.

Por ello, es importante que al implementar un modelo o estándar de calidad se realicen actividades coordinadas que permitan dirigir y controlar una organización en lo relativo a la calidad, estamos definiendo Gestión de la Calidad (Scalone, 2006)

2.2.1.1. **Gestión de la Calidad**

Scalone (Scalone, 2006) señala que la Gestión de la Calidad es un conjunto de actividades de la función general de la Dirección que determina la calidad, los objetivos y las responsabilidades. Se basa en la determinación y aplicación de las políticas de calidad de la empresa (objetivos y directrices generales), la cual se puede aplicar a nivel de la empresa en el marco de la gestión de cada proyecto.

Lo que se busca al administrar la calidad es, en primer lugar, entender las expectativas del cliente referente a la calidad, para así generar un plan para satisfacerlas. Dado que la calidad está definida por el cliente, podría parecer que es completamente subjetiva por lo que se requiere examinar cada una de las características individuales del software y determinar una o más métricas que pueden recolectarse para reflejar dichas características. (Scalone, 2006)

Un producto de software consistente, de alta calidad no puede producirse a partir de un proceso malo, por lo que se hace necesario contar con un ciclo constante de medir la calidad y actualizar el proceso, para con ello generar las métricas y a partir de ellas mejorar los procesos. Uno de los propósitos de la administración de la calidad del software es encontrar errores y defectos en el proyecto tan pronto como sea posible. Entonces, un buen proceso de administración de calidad tomará más esfuerzo y costo. De cualquier manera, habrá una gran recompensa al tiempo que el proyecto avanza.

Desde el punto de vista de la calidad, Scalone (Scalone, 2006) menciona que la Gestión de la Calidad del Software está formada por 4 partes, las cuales son: (a) Planificación, (b) Control, (c) Aseguramiento de la Calidad y (d) Mejora de la Calidad del Software. A continuación se da una explicación:

- **Planificación de la Calidad del Software**. es la parte encargada de realizar el proceso administrativo de desarrollar y mantener una relación entre los objetivos y recursos de

la organización y las oportunidades cambiantes del mercado (ISO / IEC90003, 2004). El objetivo es modelar y remodelar los negocios y productos de la empresa, de manera que se combinen para producir un desarrollo y utilidades satisfactorias. La estructura de la planificación puede ser la siguiente: (1) Introducción al Producto, una descripción del producto, su objetivo en el mercado y expectativas de calidad del producto; (2) Planes del producto, Fechas críticas respecto de la liberación del producto y responsabilidades del producto respecto de su distribución y servicio; (3) Descripciones del proceso, Procesos de desarrollo y servicios que serían usados en el desarrollo y en la administración; (4) Objetivos de Calidad, Objetivos y planes de calidad del producto, los cuales incluyen la identificación de los atributos de calidad del producto; (5) Manejo del riesgo, principales riesgos que pueden afectar la calidad del producto. Esta información puede ser presentada en diferentes documentos. El plan de calidad define los atributos de calidad más importantes del producto a ser desarrollado y define el proceso de evaluación de la calidad. En la Planificación de la Calidad del Software se debe determinar: (1) Rol de la Planificación, (2) Requerimientos de la CS, (3) Preparación de un Plan de CS, (4) Implementación de un Plan de Calidad del Software y (5) Preparar un Manual de Calidad.

- **Control de la Calidad del Software.** Son las técnicas y actividades de carácter operativo, utilizadas para satisfacer los requisitos relativos a la calidad, centradas en 2 objetivos fundamentales: (1) mantener bajo control un proceso y (2) eliminar las causas de los defectos en las diferentes fases del ciclo de vida. Está formado por actividades que permiten evaluar la calidad de los productos de software desarrollados. El aspecto a considerar en el Control de la Calidad del Software es la “Prueba del Software”. La prueba es el proceso de ejecutar un programa con intención de encontrar defectos. Es un proceso destructivo que determina el diseño de los casos de prueba y la asignación de responsabilidades. La prueba exitosa es aquella que descubre defectos. El “caso de prueba bueno” es aquel que tiene alta probabilidad de detectar un defecto aún no descubierto. El “caso de prueba exitoso” es aquel que detecta un defecto aún no descubierto.

- **Aseguramiento de la Calidad del Software.** Según la Norma ISO 9000:2000, el aseguramiento de la calidad es la parte de la gestión de la calidad orientada a proporcionar confianza en que se cumplirán los requisitos de calidad. El Aseguramiento de Calidad del Software, es el conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza que el software satisfará los requisitos dados de calidad. Este aseguramiento se diseña para cada aplicación antes de comenzar a desarrollarla y no después. El aseguramiento de la calidad del software comprende: (1) un enfoque de gestión de calidad, (2) métodos y herramientas de Ingeniería del Software, (3) revisiones técnicas formales aplicables en el proceso de software, (4) una estrategia de prueba multiescala, (5) el control de la documentación del software y de los cambios realizados, (6) procedimientos para ajustarse a los estándares de desarrollo del software y (7) mecanismos de medición y de generación de informes. El aseguramiento de calidad se refiere a validar los procesos usados para crear los productos. Es una herramienta especialmente útil para administradores y patrocinadores, ya que permite discutir los procesos usados para crear los productos para determinar si son razonables. Este aseguramiento tiene la participación de los Ingenieros de Software, que realizan el trabajo técnico y un grupo de SQA (Software Quality Assurance) que tiene la responsabilidad de la planificación de aseguramiento de la calidad, supervisión, mantenimiento de registros, análisis e informes. Las Actividades del SQA son: (1) Establecimiento de un plan de SQA para un proyecto, (2) Participación en el desarrollo de la descripción del proceso de software del proyecto, (3) Revisión de las actividades de Ingeniería del Software para verificar su ajuste al proceso de software definido, (4) Auditoria de los productos de software designados para verificar el ajuste con los definidos como parte del proceso del software, (5) Asegurar que las desviaciones del trabajo y los productos del software se documentan y se manejan de acuerdo con un procedimiento establecido, y (6) Registrar lo que no se ajuste a los requisitos e informar a sus superiores. Además de estas actividades, el grupo de SQA coordina el control y la gestión de cambios y; ayuda a recopilar y analizar las métricas del software.

Uno de los roles primarios llevados a cabo por el aseguramiento de la calidad y control de calidad es observar lo que está siendo realizado por otros, evaluarlo, criticarlo y mejorarlo. Bajo este concepto de ideas, se propone que el personal de calidad debe ser independiente de los diseñadores y desarrolladores. Esta independencia asegura que la actividad de calidad continuará observando todo el proceso desde otra perspectiva.

- **Mejora de la Calidad del Software.** Según la Norma ISO 9000:2000, la mejora de la calidad es la parte de la gestión de la calidad orientada a aumentar la capacidad de cumplir con los requisitos de la calidad. Los requisitos pueden estar relacionados con cualquier aspecto tal como la eficacia, la eficiencia o la trazabilidad. La Mejora de la Calidad del Software es la parte de la Gestión de la Calidad que contribuye, por medio de las mediciones, a los análisis de los datos y auditorias, a efectuar mejoras en la calidad del software. Una Auditoria de Calidad tiene como objetivo mostrar la situación real para aportar confianza y destacar las áreas que pueden afectar adversamente esa confianza. Otro objetivo consiste en suministrar una evaluación objetiva de los productos y procesos para corroborar la conformidad con los estándares, las guías, las especificaciones y los procedimientos. Las razones para realizar una auditoría son: (1) Establecer el estado del proyecto, (2) Verificar la capacidad de realizar o continuar un trabajo específico, (3) Verificar qué elementos aplicables del programa o Plan de Aseguramiento de la Calidad han sido desarrollados y documentados y (4) Verificar la adherencia de esos elementos con el programa o Plan de Aseguramiento de la Calidad. El propósito y la actividad de la auditoria es recoger, examinar y analizar la información necesaria para tomar las decisiones de aprobación. La auditoría es realizada de acuerdo con los planes y procedimientos documentados. El plan de auditoría establece un procedimiento para dirigir la auditoria y para las acciones de seguimiento sobre las recomendaciones de la auditoria. Al realizar la auditoria, el personal de la misma evalúa los elementos del software y los procesos para contrastarlos con los objetivos y criterios de las auditorias, tales como contratos, requerimientos, planes, especificaciones o procedimientos, guías y estándares.

La Calidad de Software es resultado del movimiento global dentro del proceso de mejoramiento continuo de los modelos y/o estándares de producción en todos los sectores industriales, en particular, cuando éste se concentra en la producción de sistemas de información y software especializado, por lo que es importante conocer los modelos y/o estándares de calidad.

2.2.1.2. **Modelos y Estándares de la Calidad del Software**

Los Modelos de Calidad son aquellos documentos que integran la mayor parte de las mejores prácticas, proponen temas de administración en los que cada organización debe hacer énfasis, integran diferentes prácticas dirigidas a los procesos clave y permiten medir los avances en calidad. (Scalone, 2006)

Los Estándares de Calidad son aquellos que permiten definir un conjunto de criterios de desarrollo que guían la forma en que se aplica la Ingeniería del Software. Los estándares suministran los medios para que todos los procesos se realicen de la misma forma y son una guía para lograr la productividad y la calidad. (Scalone, 2006)

Los modelos y/o estándares permiten que las empresas realicen sus tareas y funciones teniendo en cuenta la calidad, ya que desempeña un rol determinante para la competitividad de la empresa. Por ello implantar modelos o estándares de calidad tiene como objetivo principal que las empresas desarrollen sistemáticamente, productos, bienes y servicios de mejor calidad y cumplan con las necesidades y deseos de los clientes. Para esto, se requiere de un Modelo / Estándar que: permita: (1) unir la misión de la empresa y el esfuerzo de cada área en una sinergia de resultados hacia la competitividad y la calidad de clase mundial; y (2) tener procesos y procedimientos ágiles; y comprensibles para todos los involucrados, pasando por las etapas de desarrollo, prueba, producción y satisfacción del cliente.

Al hablar de la calidad se encuentran dos (2) enfoques, calidad de producto y calidad del proceso, por lo que se presentarán una serie de modelos y estándares asociados a cada uno de estos enfoques:

- **Modelos y Estándares para el nivel de calidad de Procesos**

A continuación se presenta una tabla donde se mencionan algunos modelos y estándares a nivel de calidad de procesos, presentando a continuación su explicación.

Tabla 1. Modelos y Estándares a Nivel de Calidad de Procesos. (Santana Cecilia, 2018)

Modelos a Nivel del Proceso	Estándares a Nivel de Calidad
CMMI	ISO 90003:2004
Bootstrap	ISO 12207:1995
Personal SW Proceess (PSP)	ITIL – Information Technology Infrastructure Library

Modelos a nivel del proceso

- **Modelo CMMI**

Según menciona Parada (Parada, 2010) el Instituto de Ingeniería de Software (SEI) basado en trabajos previos sobre el mejoramiento de procesos y con la premisa de que "la calidad de un proceso o de un producto está altamente influenciada por la calidad de los procesos usados para desarrollarlo y mantenerlo", publicó en 1.995 un libro titulado "modelo de madurez y calidad (CMM)"; una guía para el mejoramiento de los procesos de desarrollo de software.

CMM fue desarrollado por múltiples disciplinas, entre las más destacadas: modelos para ingeniería de sistema, ingenieros en software, adquisición de software, desarrollo y gestión de personal, desarrollo integrado de productos y procesos (IPPD). Aunque su aplicación tuvo éxito en la industria, el uso de muchos modelos generó un problema, debido a que se incrementaban los esfuerzos, al implantarlos por disciplinas específicas. Como iniciativa para solucionarlo surgió el proyecto CMMI cuyo objetivo fue combinar los modelos más importantes, modelo de capacidad y madurez para software (SW-CMM), modelo de capacidad para ingeniería de sistemas (SECM) y el modelo de capacidad y madurez para el desarrollo integrado de productos (IPD-CMM). Estos tres modelos fueron elegidos por su amplio reconocimiento y adopción en las comunidades de ingeniería de software y de sistemas y por sus diferentes enfoques para mejorar los procesos de una empresa.

También Parada (Parada, 2010) indica que el modelo puede ser aplicado de dos formas: por etapas y continuo: la representación continua permite a una organización seleccionar un grupo de áreas y mejorar los procesos relacionados con ellas. Se usan niveles de capacidad para caracterizar las mejoras relativas a un área de proceso individual. La representación por etapas usa un conjunto predefinido de áreas de proceso y define un camino de mejorar para una organización. Este camino es caracterizado por niveles de madurez. Cada nivel de madurez proporcionaría un conjunto de áreas de proceso que representan diferentes funciones de las empresas.

Los niveles de capacidad consisten en unas metas genéricas y prácticas genéricas relacionadas al área de proceso que pueden mejorar los procesos de una organización asociados con un área de proceso.

Los niveles de madurez consisten en las prácticas específicas y genéricas relacionadas para un sistema predefinido de las áreas de proceso que mejoran el funcionamiento total de la organización. Se miden por el logro de las metas específicas y genéricas asociados con cada uno cada conjunto de áreas de proceso predefinidos.

Tabla 2. Niveles de Madurez y de capacidad (Parada, 2010)

Nivel	Representación continua Nivel de capacidad	Representación por etapa Niveles de madurez
Nivel 0	Incompleto	N/A
Nivel 1	Ejecutado	Inicial
Nivel 2	Gestionado	Gestionado
Nivel 3	Definido	Definido
Nivel 4	Gestionado cuantitativamente	Gestionado cuantitativamente

También encontramos que CMMI presenta áreas de procesos, las cuales indica Parada (Parada, 2010), son un conjunto de prácticas relacionadas de un área específica, que cuando se implementan de manera colectiva, satisfacen un conjunto de metas que son importantes para lograr el mejoramiento de esa áreas. También señala que las áreas de proceso se agrupan en cuatro categorías: Gestión de procesos, Gestión de proyectos, Ingeniería y Soporte.

- **Gestión de procesos**. Las áreas de proceso de este grupo contienen las actividades relacionadas con la definición, planificación, despliegue, implementación, seguimiento, control, evaluación, medición y mejoramiento de los procesos.
- **Gestión de proyectos**. Las áreas de procesos de gestión de proyectos cubren las actividades de gestión de proyecto relacionadas con: planificación, seguimiento y control.
- **Ingeniería**. Las áreas de proceso de ingeniería cubren las actividades de desarrollo y mantenimiento que son compartidas por las diferentes disciplinas de la ingeniería; se aplican al desarrollo de algún producto o servicio en el dominio de desarrollo. Hacen énfasis en el desarrollo concurrente y se enfocan en todas las fases de la vida del producto.
- **Soporte.** Son las áreas que cubren las actividades que soportan el desarrollo y mantenimiento del producto. En general estas áreas tratan de los procesos que se dirigen hacia el proyecto y pueden tratar procesos que aplican de forma más general a la organización.

En cuanto a las áreas de aseguramiento de la calidad y de pruebas en el modelo CMMI, nos indica Cardona (Cardona V, 2009) que en CMMI existen tres áreas de proceso (PA, por sus siglas en inglés) que están dirigidas explícitamente a asegurar la calidad del software; estas son:

- **QA o Aseguramiento de la Calidad**: hace parte del segundo nivel de madurez de CMMI y su objetivo principal es que los procesos y los elementos de trabajo cumplan los procesos; de ésta forma la norma sugiere la realización de ciertas tareas para cumplir con dicha área de proceso las cuales incluyen:
    a) Evaluar objetivamente la ejecución de los procesos, los elementos de trabajo y servicios contra las descripciones de procesos, estándares y procedimientos.
    b) Identificar y documentar los elementos no conformes.

c) Proporcionar información a las personas que están usando los procesos y a los gestores de los resultados de las actividades del aseguramiento de la calidad.
d) Asegurar que los elementos no conformes son corregidos.

Generalmente en la industria del software, al momento de implantar ésta área de proceso se tiende a subcontratar las pruebas de software bajo el modelo de "Outsourcing". En otros casos la subcontratación no se da; en su lugar se establece un área de pruebas al interior de la empresa ligada al proceso productivo. En general los autores aseguran que la clave de éxito o fracaso de los pruebas de software están asociadas a la subcontratación o no de las mismas. Algunos aseveran que la subcontratación es vital para que la calidad del software sea conforme debido a que no se sesgan los resultados de las pruebas. Contrariamente, otros autores dicen que los procesos de aseguramiento de la calidad, específicamente los relacionados con las pruebas de software, deben ir inmersos totalmente en el proceso productivo de la organización, dado que la retroalimentación y su interrelación agregan madurez a todo el proceso.

La verificación y la validación, pretenden en cualquier estándar (y particularmente en CMMI) convertirse en una herramienta para la colaboración y el incremento del rendimiento de los equipos de trabajo, permitiendo así la conformidad del producto final basada en altos estándares de calidad.

- **Verificación**: hace parte del tercer nivel de madurez en CMMI. Su propósito principal es el de asegurar que los productos de trabajo seleccionados responden a los requisitos especificados. Sus objetivos o metas específicas son: preparar la verificación, realizar revisiones por terceros y verificar los productos de trabajo seleccionados.
- **Validación**: Hace parte del tercer nivel de madurez en CMMI. Su propósito es el de demostrar que un producto o componente satisface su uso, en el ambiente operativo planeado. Sus objetivos o metas específicas son: preparar la validación y realizar la validación de los productos o componentes de trabajo

Aunque ambas áreas de proceso no son consideradas como pruebas de software directamente, si tiene un valor relevante a la hora de hacer que los productos de software posean características de alta calidad. Las prácticas que se derivan permiten realizar pruebas de software sobre productos más maduros y confiables.

Es importante tener presente, que los procesos de pruebas no componen totalmente área de proceso de aseguramiento de calidad; es así como las pruebas proveen los elementos y prácticas de mejora con el fin de retroalimentar los procesos.

2.2.1.2.1.1. Modelo BOOTSTRAP

Según, Scalone (Scalone, 2006) este proyecto fue creado por la Comisión Europea como parte del programa ESPRIT (ESPRIT 5441 BOOTSTRAP: A European Assessment Method to Improve Software Development). La administración y el mantenimiento del programa Bootstrap corresponden al Grupo Europeo de Interés Económico del Instituto Bootstrap (Bootstrap Institute European Economic Interest Group, BI EEIG) de Milán, Italia. El interés principal del programa Bootstrap es evaluar y mejorar la capacidad de las Unidades Productoras de Software (SPU, Software Producing Units). Este proyecto al igual que otros, tiene como principio el reducir costos y mejorar la calidad previendo problemas. Su objetivo es desarrollar un método para la evaluación de procesos de desarrollo de software (SW). Inicialmente se basó en el modelo de madurez de CMM añadiendo conceptos de calidad de ISO 9000. A esto incluyó conceptos para poder evaluar desarrollos de SW de otras industrias distintas a la militar y cambiar su cobertura de evaluación para tomar desde pequeñas UPS hasta grandes corporaciones. Para lograr esto, ha puesto especial énfasis en los conceptos de ISO 9000; generando guías para mejoras en procesos de desarrollo de SW; analizado evaluaciones y mejoras de los procesos de desarrollo; y manteniendo una base de datos de soporte.

Por esta causa, (Scalone, 2006) señala que los Objetivos de la metodología Bootstrap son: (1) Proporcionar soporte para la evaluación de la capacidad de los procesos utilizando un conjunto de prácticas de Ingeniería del Software (SW). (2) Incluir estándares de Ingeniería del Software reconocidos internacionalmente como fuentes para la identificación de las prácticas a

considerar. (3) Dar soporte a la evaluación, indicando como el estándar de referencia ha sido aplicado en la organización evaluada. (4) Asegurar la fiabilidad y capacidad de repetición de la evaluación. (5) Identificar las fortalezas y debilidades de los procesos de la organización evaluada. (6) Dar soporte a la creación y aplicación de un plan de mejora que genere unos resultados aceptables y fiables, de forma que las acciones del plan de mejora permitan alcanzar los objetivos de la organización. (7) Ayudar a incrementar la eficacia de los procesos poniendo en práctica los requisitos del estándar en la organización.

También señala Scalone (Scalone, 2006) que el modelo Bootstrap define el paradigma Organización-Metodología-Tecnología que se usa en Bootstrap para los niveles de evaluación y agrupación de resultados. Dicho modelo ha sido estructurado en correspondencia con la arquitectura de procesos definida en la ISO 15504 V2.0.

El modelo de procesos de Boostrap 3.2 tiene la siguiente estructura:

1- Organización
ORG.1 Ingeniería del Negocio
ORG.2 Gestión de Recursos Humanos
ORG.3 Gestión de la Infraestructura
2- Metodología
2.1- Ciclo de Vida Dependiente
ENG.0 Definición de desarrollo
ENG.1 Análisis de Requerim. del Sistema
ENG.2 Diseño de la Arquitectura del Sist.
ENG.3 Análisis de los Requerim. del SW
ENG.4 Diseño de la Arquitectura del SW
ENG.5 Diseño Detallado del SW
ENG.6 Implementación y Prueba del SW
ENG.7 Prueba e Integración del SW
ENG.8 Prueba e Integración del Sistema
ENG.9 Mantenimiento
ENG.10 Migración
ENG.11 Retiro
2.2- Ciclo de Vida Independiente
2.2.1- Gestión
MAN.0 Gestión
MAN.1 Gestión del proyecto
MAN.2 Gestión de la Calidad
MAN.3 Gestión de riesgos
MAN.4 Gestión de subcontrato
MAN.5 Gestión de reuso
2.2.2- Soporte
SUP.1 Documentación
SUP.2 Gestión de la configuración
SUP.3 Aseguramiento de la calidad
SUP.4 Verificación
SUP.5 Validación
SUP.6 Revisión completa
SUP.7 Auditoria
SUP.8 Resolución de problemas
2.2.3- Cliente Proveedor
CUS.1 Adquisición
CUS.2 Gestión de la necesidad del cliente
CUS.4 Operación de software
CUS.5 Atención al cliente

CUS.3 – Suministro
2.3- Relacionado a proceso

PRO.1 Definición de proceso	PRO.3 Evaluación de proceso
PRO.2 Mejora de proceso	PRO.4 Medición

3- Tecnología

TEC.1 Innovación tecnológica	TEC.3 Soporte tecnológico para los procesos independientes del ciclo de vida
TEC.2 Soporte tecnológico para los procesos del ciclo de vida	TEC.4 Herramienta de integración

Cada proceso tiene un conjunto de prácticas base asociadas, que describen las actividades esenciales de un proceso específico, la realización de las prácticas base indica el grado de alcance de la finalidad del proceso. Cada atributo de proceso tiene un conjunto de prácticas de gestión asociadas, que son las que implementan o institucionalizan un proceso de una manera general. La realización de las prácticas de gestión indica la consecución del atributo en esa instancia del proceso.

Respecto del Proceso de Evaluación, se puede decir que el modelo Bootstrap se basa en evaluar las unidades de producción de software (UPS) de la organización, a través de sus proyectos para hacer un cambio a toda la organización. El proceso de evaluación es parte de la mejora. Los resultados de la evaluación dan la entrada principal para el plan de acción de mejora y una realimentación para las actividades de mejora implementadas. Durante una evaluación Bootstrap los procesos organizacionales son evaluados para definir cada proceso.

La evaluación de la capacidad de los procesos está basada en el modelo de procesos Bootstrap. Dentro de este proceso, hay 4 etapas principales: (1) Preparación, (2) Ejecución de la evaluación, (3) Determinación del nivel de madurez y capacidades; y (4) Presentación de los resultados de la evaluación.

Bootstrap podría ser principalmente un método de evaluación y solo sugiere mejoras. Esto da una pequeña ayuda una vez que se produce la mejora. La metodología está creada para auto-evaluación y evaluación de tercera parte, pero es también posible usarla para evaluación de segunda parte. Aunque muchos usuarios de Bootstrap dicen tener problemas con los cuestionarios, la crítica general del modelo es en su mayoría positiva. La metodología tiene

una gran ventaja, compara los resultados de la evaluación con los resultados de sus competidores. Parece que da un gran resultado cuando prioriza qué necesidades se deben mejorar primero.

2.2.1.2.1.2. Personal Software Process (PSP)

Según Scalone (Scalone, 2006) el Personal Software Process (PSP) es un proceso de software definido y medido diseñado para ser usado por medio de un Ingeniero de Software individual. El PSP fue desarrollado por Watts Humphrey y tiene como objetivo guiar el planeamiento y desarrollo de los módulos de software o pequeños programas; y es adaptable a otras tareas del personal. Es una tecnología de SEI (Software Engineering Institute) que trae disciplina a las prácticas de los Ingenieros de Software, mejorando la calidad del producto, aumentando los costos y reduciendo el tiempo del ciclo de desarrollo del software.

El proceso de PSP consiste de un conjunto de métodos, formularios y scripts que muestran a los Ingenieros de Software cómo planificar, medir y administrar su trabajo. El PSP está diseñado para ser usado en algún lenguaje de programación o metodología de diseño, y puede ser usado en varios aspectos del trabajo de software. Consiste en un proceso de nivel 5 de CMM diseñado para calcular el costo individual. Se aplica en la mayoría de las tareas de desarrollo de software como ser: (1) definición de requerimientos, (2) diseño de la arquitectura, (3) desarrollo del módulo, (4) producción de la documentación, (5) pruebas del sistema , (6) mantenimiento del sistema y (7) desarrollo de pequeños programas. Es un prerrequisito del planeamiento de la organización para producir el TSP (Team Software Process).

El proyecto PSP estuvo apuntando a demostrar que un proceso de nivel 5 de CMM podría ser usado por un particular para desarrollar software de alta calidad. PSP requiere de la participación de todos los niveles del management. Una estrategia efectiva es primero involucrar a los principales Ejecutivos y Gerentes para luego entrenar a los Ingenieros respecto de PSP. El PSP muestra a los Ingenieros como: (1) administrar la calidad de sus

proyectos, (2) mejorar las estimaciones y planificaciones; y (3) reducir los defectos de sus productos.

Scalone (Scalone, 2006) indica que el diseño de PSP está basado en los siguientes principios de planificación y calidad: (1) Todo Ingeniero es diferente y para ser más efectivo, los Ingenieros deben planificar su trabajo y deben basar sus planes en sus datos personales (2) Para mejorar el performance, los Ingenieros deben usar los procesos bien definidos y medidos (3) Para producir productos de calidad, los Ingenieros deben ser responsables de la calidad de sus productos. Los productos superiores no son producidos por medio de errores, los Ingenieros deben hacer lo posible por realizar un trabajo de calidad (4) Menor costo en encontrar y arreglar los defectos de un proceso (5) Es más eficiente prevenir defectos que encontrar y arreglarlos.

El principio de PSP es que para producir sistemas de software de calidad, todo Ingeniero que trabaje en el sistema, debe hacer un trabajo de calidad. El PSP está diseñado para ayudar a los profesionales de software a utilizar las prácticas de Ingeniería. También muestra cómo planear y seguir el trabajo, cómo usar un proceso definido y medido; y sigue el performance de estos objetivos. El PSP muestra a los Ingenieros cómo administrar la calidad desde el comienzo del trabajo, cómo analizar los resultados de cada trabajo, y cómo utilizar los resultados para mejorar el proceso en el próximo proyecto.

La estructura del proceso PSP comienza con los requerimientos, y con el primer llamado “Planificación”. Hay un script de planificación que sirve de guía para este trabajo y un resumen de la planificación para registrar los datos de la planificación. Los Ingenieros registran el tiempo y los datos de los defectos. Al final del trabajo, durante la última etapa (post mortem), los Ingenieros sumarizan tanto los datos de los defectos como los tiempos, miden el tamaño del programa e ingresan estos datos en el resumen de plan. Luego, se entrega el producto terminado con el resumen de la planificación.

El PSP está basado en las mismas prácticas industriales que el SEI CMM y ha sido adaptado a varias tareas de Ingeniería de Software, tales como desarrollo de requerimientos de software, especificaciones de software y casos de prueba.

Estándares a nivel del proceso

2.2.1.2.1.3. Estándar ISO 90003: 2004

Sobre este tema, (Scalone, 2006) demuestra que la ISO 90003:2004 provee una guía para las organizaciones respecto de la aplicación de ISO/IEC 9001:2000 en la adquisición, suministro, desarrollo, operación y mantenimiento de software y servicios de soporte. Esta norma no agrega o cambia los requerimientos de ISO/IEC 9001:2000. Las guías de ISO 90003:2004 no tienen el propósito de ser utilizadas como criterio de evaluación en una certificación de SGC (Sistema de Gestión de la Calidad).

La aplicación de ISO 90003:2004 es apropiada para un software que: (1) Forma parte de un contrato comercial con otra organización, (2) Es un producto disponible para un sector del mercado, (3) Es usado para soportar los procesos de una organización y (4) Está relacionado a servicios de software.

La Norma cuenta con 5 capítulos que especifican actividades que deben ser consideradas cuando se implemente el SGC. Los capítulos son: (1) Sistema de Gestión de la Calidad, (2) Responsabilidad de la Dirección, (3) Gestión de los Recursos, (4) Realización del Producto y (5) Medida, Análisis y Mejora.

La Estructura de la Norma ISO 90003:2004 es: (1) Ámbito, (2) Normas para la consulta, (3) Términos y definiciones, (4) Sistema de gestión de la calidad, (5) Responsabilidad de la dirección, (6) Gestión de los recursos, (7) Realización del producto, (8) Medición, análisis y mejora; y (9) Bibliografía

ISO 90003:2004 es independiente de la tecnología, modelos del ciclo de vida, procesos de desarrollo, secuencia de actividades y estructura organizacional utilizada en la organización. Esta Norma permite la aplicación de ISO/IEC 9001:2000 y en particular, ISO/IEC 12207, ISO/IEC TR 9126, ISO/IEC 14598, ISO/IEC 15939 e ISO/IEC TR 15504.

2.2.1.2.1.4. ISO/IEC 12207:1995

La disciplina del software necesita migrar de esta proliferación a un marco común que pueda ser usado para “hablar el mismo lenguaje” al crear y administrar software. Esta norma provee

este marco común, el cual cubre el ciclo de vida del software desde su conceptualización hasta su retiro, y consiste de procesos para adquirir y suministrar productos y servicios de software. Este marco permite controlar y mejorar estos procesos.

ISO/IEC 12207 puede ser usado para: (1) Adquirir, suministrar, desarrollar, operar y mantener software, (2) Soportar las funciones arriba mencionadas mediante el aseguramiento de calidad, administración de la configuración, revisiones conjuntas, auditorias, verificación, validación, resolución de problemas y documentación; (3) Administrar y mejorar tanto al personal como a los procesos de la organización, (4) Establecer la administración del software y los ambientes de Ingeniería basados en los procesos de ciclo de vida que se adapten para servir a las necesidades del negocio, (5) Ayudar a un mejor entendimiento entre clientes y proveedores; y entre las partes involucradas en el ciclo de vida de un producto de software y (6) Facilitar la comercialización global del software.

ISO/IEC 12207 contiene procesos, actividades y tareas a ser aplicadas durante la adquisición de un sistema que contiene software, un producto de software o un servicio de software; y durante el suministro, desarrollo, operación y mantenimiento de productos de software. Este estándar internacional provee un proceso que puede ser utilizado para definir, controlar y mejorar los procesos de ciclo de vida del software. Este estándar es escrito para los consumidores de sistemas y productos de software, y para los suministradores, desarrolladores, operadores, mantenedores, administradores, responsables de calidad de software y usuarios de productos de software.

ISO/IEC 12207 describe la arquitectura de los procesos de ciclo de vida del software, pero no especifica los detalles de cómo implementar o realizar las actividades y tareas incluidas en los procesos. El software no prescribe un modelo particular de ciclo de vida o un método de desarrollo de software. Este estándar agrupa las actividades que deben ser realizadas durante el ciclo de vida del software en: 5 Procesos Principales, 8 Procesos de Soporte, y 4 Procesos Organizacionales. Cada proceso del ciclo de vida está dividido en un conjunto de actividades donde cada actividad está dividida en un conjunto de tareas. Para la aplicación de ISO 12207 se necesitan de los siguientes factores: (1) Políticas organizacionales, (2) Estrategia de adquisición, (3) Concepto de soporte, (4) Modelos del ciclo de vida, (5) Partes involucradas,

(6) Actividad del ciclo de vida del sistema, (7) Características a nivel sistema y (8) Características a nivel del software con los estándares ISO 15504 y la ISO 12207.

2.2.1.2.1.5. ITIL – Information Technology Infrastructure Library

Scaolone (Scalone, 2006) indica que ITIL® (IT Infrastructure Library) es el marco de procesos de Gestión de Servicios de TI más aceptado. ITIL proporciona un conjunto de mejores prácticas, extraídas de organismos punteros del sector público y privado a nivel internacional, que han sido recogidas por la Oficina Gubernativa de Comercio Británica (OGC, Office of Goverment Comerce). Este framework o marco de procesos es utilizado por cientos de organizaciones en el mundo y ha sido desarrollado reconociendo la dependencia creciente que tienen éstas en la tecnología para alcanzar sus objetivos.

ITIL es un conjunto de buenas prácticas de dirección y gestión de servicios de tecnologías de la información en lo referente a personas, procesos y tecnología, desarrollado por la OGC del Reino Unido, que cumple y desarrolla la norma BS15000 de la BSI (British Standards Institution). La realización de las buenas prácticas especificadas en ITIL hace posible que los departamentos y organizaciones puedan reducir costes, mejorar la calidad del servicio tanto de clientes externos como internos y aprovechar al máximo las habilidades y experiencia del personal, mejorando su productividad.

El principal objetivo de ITIL es alinear el negocio con los Sistemas de Información. El soporte a Servicio definido por ITIL ayuda a la empresa a gestionar el hardware, software y los recursos humanos para asegurar la continuidad y no interrupción de los servicios ofrecidos por el negocio. ITIL define que el objetivo principal del soporte de TI es: “Ofrecer el mejor servicio posible, sin interrupciones”.

ITIL consta de un conjunto de libros que permiten mejorar notablemente la calidad de los servicios de tecnologías de la información que presta una organización a sus clientes o a un departamento de su organización. Existe una organización internacional integrada por grupos de proveedores y usuarios, denominada itSMF (IT Service Management Forum) que promueve y desarrolla el uso de buenas prácticas en administración de servicios de tecnologías de la información.

La filosofía ITIL impulsa la adopción de procesos, de manera que puedan adaptarse para aplicarlos tanto en organizaciones grandes como en pequeñas. Esto es aplicable especialmente en aquellas empresas que han integrado clientes y proveedores en sus operaciones a través de redes de datos, como Internet. En este caso, la integración de procesos tecnológicos y de negocio resulta indispensable. ITIL trata acerca de los procesos que deben realizarse dentro de la organización para que la gestión y operación sobre las infraestructuras promocione un servicio óptimo a los clientes con unos costes justificables.

El objetivo principal de la Gestión de Servicios es asegurar que los Servicios IT (Information Technology) están alineados con las necesidades del negocio. Un servicio, en la economía, es el equivalente no material de un bien que tiene los siguientes atributos: intangible, pérdida de tiempo, falta de homogeneidad, fluctuaciones de la demanda e implicación del comprador.

También indica, (Scalone, 2006) que la OGC analizando la Gestión de Servicios para encontrar los requerimientos de los clientes y contrastándola con los grupos de usuarios, diseñó un esquema que abarca cinco elementos: (1) The Business Perspectiva, (2) Deliver IT Services, (3) Support IT Services, (4) Manage the Infrastructure y (5) Managing Applications.

<u>The Business Perspectiva - Perspectiva del Negocio</u>

Estos procesos están orientados a ayudar a los administradores del negocio a entender la provisión de servicios de TIC (Technology Information Center). Abarca administración de la continuidad del negocio, alianzas y outsourcing, sobrevivir al cambio; y alineación del negocio y TIC.

<u>Deliver IT Services - Entrega de Servicios</u>

Estos procesos se encargan de observar las necesidades que tiene el negocio de servicios de TIC, cuidando la calidad con que estos son entregados. La calidad de un servicio debe ser cuantificable y alcanzable, esto se logra a través del establecimiento de acuerdos del nivel de cada servicio y están sustentados en la información que proporcionan otros procesos tales como la capacidad de los recursos que están involucrados en los servicios, la disponibilidad calculada y la medida, con sus relaciones entre sí y las causas de incumplimiento en tal caso que se presente.

Support IT Services - Soporte de Servicios

En este segmento de procesos, la preocupación está dirigida al acceso que tienen los usuarios a los servicios que soportan sus funciones de negocio. Los usuarios de cada uno de los servicios de TIC que se entregan en la organización requieren de un soporte para su correcta operación y atención cuando se producen fallos. Este elemento abarca: administración de incidentes, administración de problemas, administración de versiones, administración de cambios y administración de la configuración.

Manage the Infrastructure - Administrar la Infraestructura

Se centra en la gestión de la administración de sistemas (máquinas, redes, sistemas operativos, etc).

Managing Applications - Administrar las aplicaciones

Ofrece un conjunto de buenas prácticas para la gestión de todo el ciclo de vida de las aplicaciones, centrándose sobre todo en la definición de requisitos e implementación de soluciones.

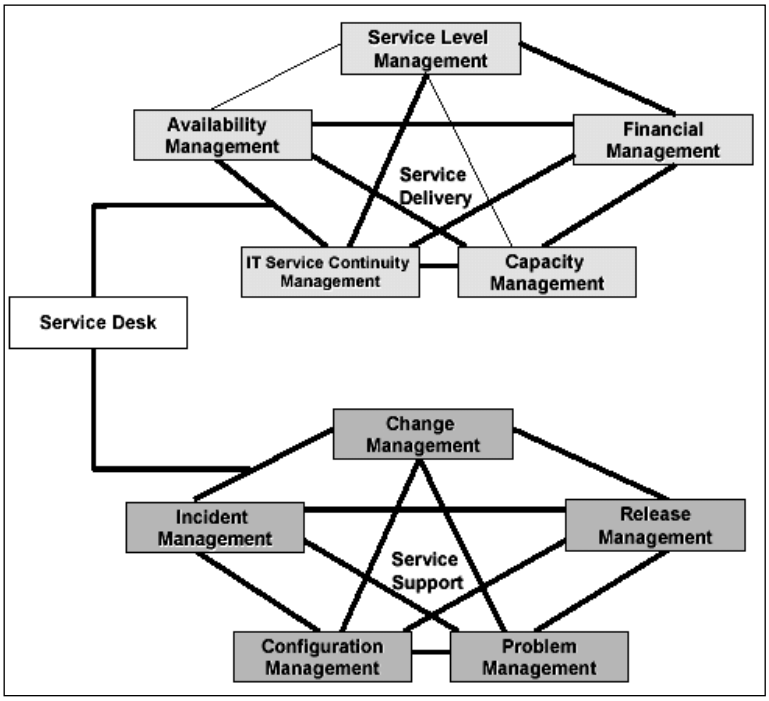


Figura 1. Áreas de los Procesos de ITIL (Scalone, 2006)

Algunos de estos elementos tienen bien definidas sus fronteras pero, en otros, su delimitación queda difuminada (sobre todo en los niveles altos) debido a que influyen en el marco de actuación de los demás o toman requerimientos de éstos.

Con ITIL se definieron unos modelos de proceso que fueron probados antes de llevarlos a la práctica y que conforman, al día de hoy, las mejores prácticas para la Gestión de Servicios de TI. El contenido de las Mejores Prácticas de ITIL está en continua evolución a medida que surgen nuevos desafíos y es aplicable a cualquier organización de TI independientemente de su tamaño o de la tecnología que emplee.

Las mejores prácticas de la Gestión de Servicios de TI han evolucionado desde 1989, año en que se publicaron las primeras librerías de ITIL. Su utilización se soporta mediante una certificación y una estructura de formación que han sido adoptadas a nivel mundial para reconocer la competencia profesional en Gestión de Servicios de TI.

Adoptar las buenas prácticas ITIL no requiere adoptar e implementar todos los grupos de procesos simultáneamente. Esta libertad para escoger los grupos de procesos que requiere una determinada organización, es uno de los principales motivos para adoptar ITIL en empresas independientemente de su tamaño.

La conjunción de ITIL con estándares de control y madurez de gestión en TI (COBIT, CMM) y con estándares para establecimiento de procesos de negocio y toma de decisión (Balance Scorecard, etc.) permite medir el rendimiento de la organización de TI, resultando más sencillo, para la Dirección, el poder establecer objetivos y toma de decisiones.

2.2.1.2.2. Modelos y Estándares para el nivel de calidad del producto

A continuación se presenta un cuadro con los modelos y estándares a nivel de calidad del producto.

Tabla 3. Modelos y Estándares a Nivel de Calidad del Producto (Santana Cecilia, 2018)

MODELOS	ESTANDARES
Modelo de Gilb	ISO/IEC 9126-1:2001 – Quality Model
Modelo GQM(Goal – Question – Metric)	ISO/IEC 25000:2005 - SQuaRE
Modelo de McCall	

Modelos a nivel de calidad del producto

2.2.1.2.2.1. Modelo de GILB

Scalone (Scalone, 2006) señala que el modelo de Gilb plantea la creación de una especificación de requisitos de calidad para cada proyecto que deben escribir conjuntamente el usuario y el analista. Es un modelo que permite determinar una lista de características que definen la calidad de la aplicación. Estas pueden ser de 2 tipos: (1) Originales y (2) de modelos tradicionales.

Las características se pueden medir mediante varias sub características o métricas detalladas. Para cada una de ellas, se deben especificar los siguientes conceptos: nombre y definición de la característica, escala o unidades de medición, recopilación de datos o prueba, valor previsto, valor óptimo, valor en el sistema actual y comentarios.

En la década del ochenta, se comenzaron a usar modelos particulares de evaluación para cada empresa o proyecto, implantándose el concepto de calidad relativa. Este enfoque se ha asociado a la filosofía QFD (Quality Function Deployment), o el despliegue de la función de la calidad, que se aplica al ámbito de la gestión de la calidad industrial y en el que se han basado modelos posteriores.

2.2.1.2.2.2. Modelo GQM (Goal – Question - Metric)

Para este modelo GQM (objetivo-pregunta-métrica /goal – question - metric) de Basili y Rombach (1998), Scalone (Scalone, 2006) dice que es una propuesta de objetivos / metas orientado a la definición de modelos de calidad. Se propone el paradigma GQM para evaluar la calidad de cada proyecto. Este modelo utiliza una propuesta para definir un modelo de calidad hasta obtener las métricas respectivas con el análisis e interpretación de los datos de las mediciones respectivas. Plantea el enfoque de medición para evaluar la calidad del software basado en la identificación de objetivos a lograr. El enfoque de GQM basa la mejora en la definición clara de procesos y productos. Proporciona la estructura para obtener los objetivos cruciales del proyecto. Consta de 3 etapas: 1- Listar los objetivos principales del

desarrollo y mantenimiento del proyecto; 2- Para cada objetivo, se deben obtener las preguntas que deben contestarse para saber si se están cumpliendo los objetivos; 3- Decidir qué medir para poder contestar las preguntas de manera adecuada, es decir, desarrollar un conjunto de métricas que ayuden a responder la pregunta. Las medidas individuales obtenidas se relacionan para poder ser utilizadas en el contexto del proyecto completo. Otro aspecto preponderante en el enfoque GQM es la interpretación de los datos recolectados en función de las preguntas a partir de las cuales se derivaron esas medidas.

La idea fundamental del GQM es la medición orientada a objetivos / metas, la cual está basada en el contexto. De acuerdo a esto, el modelo de medición tiene 3 niveles:

(1) Nivel Conceptual (Goal): un objetivo / meta es definido para un propósito específico en base a las necesidades de la organización, teniendo en cuenta una variedad de razones, desde distintos puntos de vista relacionados a un ambiente en particular. Un objetivo / meta representa el nivel máximo de característica de calidad.

(2) Nivel Operacional (Question): es un conjunto de preguntas que son utilizadas para caracterizar la forma de realización de una meta específica. Cada característica de nivel máximo es redefinida en las subcaracterísticas usando un conjunto de preguntas.

(3) Nivel Cuantitativo (Metric): es un conjunto de datos que está asociado a toda pregunta de manera cuantitativa. Para cuantificar una sub característica se utiliza un conjunto de métricas. La interpretación de las métricas es utilizada para responder a las preguntas y concluir si la meta u objetivo se ha cumplido.

El modelo GQM es un enfoque útil para decidir qué medir. Es un enfoque orientado a metas, por lo tanto, permite a los tomadores de decisión, elegir aquellas métricas que se relacionen a las metas más importantes de los problemas más urgentes.

2.2.1.2.2.3. Modelo de McCALL

El modelo fue propuesto por Mc Call en 1977 y está orientado a los desarrolladores de Sistemas, para ser utilizado durante el proceso de desarrollo y abarca tres 3 áreas de trabajo. Organiza los factores en tres ejes o puntos de vista desde los cuales el usuario puede

contemplar la calidad de un producto, estos son: (1) Operación del producto, (2) Revisión del producto y (3) Transición del producto. Cada punto de vista se descompone en una serie de factores que determinan la calidad de cada una de ellos. Cada factor determinante de la calidad, se descompone, a su vez, en una serie de criterios o propiedades que determinan su calidad. Los criterios pueden ser evaluados mediante un conjunto de métricas. Para cada criterio deben fijarse unos valores máximo y mínimo aceptables para cada criterio.

Tabla 4. Áreas de trabajo del Modelo de Mc Call (Santana Cecilia, 2018)

ÁREA DE TRABAJO	DESCRIPCION
Operación del producto	Requiere que pueda ser comprendida rápidamente, operada eficientemente y que los resultados sean aquellos requeridos por el usuario.
Revisión del producto	Está relacionada con la corrección de errores y la adaptación de los sistemas. Esto es importante porque es generalmente considerada como la parte más costosa en el desarrollo de software.
Transición del producto	Puede que no sea muy importante en todas las aplicaciones. Sin embargo, la orientación a procesamiento distribuido y el rápido cambio en el hardware es probable que incremente su importancia.

Las tres (3) áreas indicadas en el cuadro anterior que contempla el Modelo de Mc Call abarcan los siguientes aspectos: (Pressman, 1998)

Operación del Producto:

- Corrección. Hasta dónde satisface un programa su especificación y logra los objetivos de la misión del cliente.
- Fiabilidad. Hasta dónde se espera que un programa lleve a cabo su función pretendida con la exactitud requerida.
- Eficiencia. Cantidad de recursos informáticos y código necesarios para que un programa realice su función.
- Integridad. Hasta dónde se controla el acceso al software o a los datos por personas no autorizadas.
- Usabilidad (Facilidad de manejo). Esfuerzo necesario para aprender, operar, preparar los datos de entrada e interpretar las salidas (resultados) de un programa.

Revisión del Producto:

- Mantenibilidad (Facilidad de mantenimiento). Esfuerzo necesario para localizar y arreglar un error del programa (definición limitada).
- Flexibilidad. Esfuerzo necesario para modificar un programa operativo.
- Facilidad de prueba. Esfuerzo necesario para probar un programa para asegurarse que realiza su función pretendida.

Transición del producto:

- Portabilidad. Esfuerzo necesario para transferir el programa de un entorno de sistema de hardware y/o software a otro.
- Reusabilidad (capacidad de reutilización). Hasta dónde se vuelve a emplear un programa (o partes de un programa) en otras aplicaciones, en relación al empaquetamiento y alcance de las funciones que realiza el programa.
- Interoperatividad. Esfuerzo necesario para acoplar un sistema con otro.

Estándares de Calidad a nivel del producto

Según Scalone (Scalone, 2006), se puede encontrar los siguientes estándares para el nivel de calidad del producto:

2.2.1.2.2.4. ISO/IEC 9126-1:2001 – Quality Model

Esta parte de la ISO 9126 describe el modelo de calidad del producto de software. La primera parte del modelo especifica seis (6) características de calidad interna y externa, las cuales están divididas en sub características, son manifestadas externamente cuando el software es utilizado como parte de un sistema, y son un resultado de atributos internos del software.

La calidad externa evalúa que el software satisfaga las necesidades del usuario teniendo en cuenta las condiciones especificadas. Esta calidad es medible en el comportamiento del producto. La calidad interna evalúa el total de atributos que un software debe satisfacer teniendo en cuenta condiciones especificadas. Esta calidad es medible a partir de las características intrínsecas. Las características definidas son aplicables a todo tipo de software.

Las características y sub características proveen una terminología consistente respecto de la calidad del producto del software. Esta Norma permite especificar y evaluar la calidad del software desde distintas perspectivas, las cuales están asociadas a la adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad, y auditoria del software.

Puede ser usada por desarrolladores, evaluadores independientes y grupos de aseguramiento de la calidad responsable de especificar y evaluar la calidad del software.

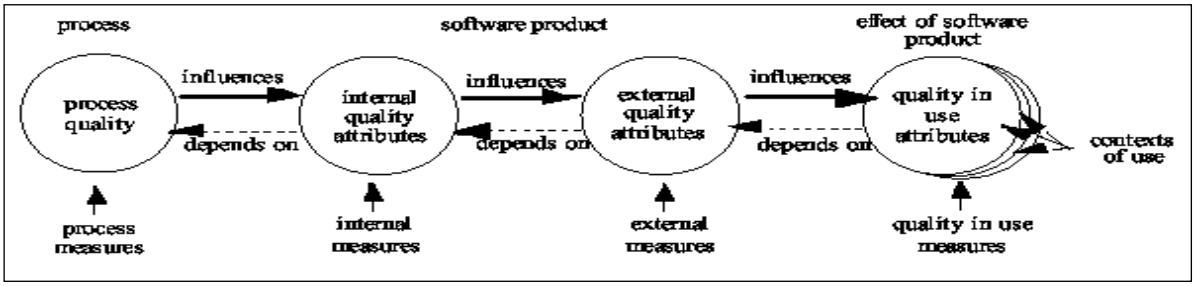


Figura 2. Calidad en el ciclo de vida según ISO/IEC 9126-1

También indica Scalone (Scalone, 2006) que la evaluación de los productos de software que satisfacen las necesidades de calidad del software es uno de los procesos del ciclo de vida de desarrollo del software. La calidad del producto de software puede ser evaluada por medio de la medición de atributos internos, externos o a través de la calidad en uso. El objetivo es que el producto tenga el efecto requerido en un contexto particular de uso. La calidad del proceso contribuye a mejorar la calidad del producto, y la calidad del producto contribuye a mejorar la calidad en uso. Evaluar y mejorar la calidad de un proceso contribuye a mejorar la calidad del producto; y esto contribuye a mejorar la calidad en uso. De manera similar, evaluar la calidad en uso puede mejorar la calidad del producto, y evaluar un producto puede mejorar un proceso.

El modelo de calidad de ISO 9126-1 establece tres (3) niveles: Característica, Sub característica y Métricas. Existen métricas internas y externas. Las métricas internas pueden ser aplicadas a un software no ejecutable durante el diseño y la codificación. Las métricas externas se utilizan en el software ejecutable. El modelo de calidad interna y externa está

formado por las siguientes características: Funcionalidad, Confiabilidad, Facilidad de uso, Eficiencia, Facilidad de mantenimiento y Portabilidad.

Funcionalidad. La funcionalidad es la extensión o cantidad de posibilidades provistas por un sistema. Uno de los problemas más difíciles es saber cuánta funcionalidad es suficiente. Lo que para una persona puede ser un rasgo insignificante puede ser una característica indispensable para otra. La solución aquí es trabajar una y otra vez en la consistencia del sistema en general, tratando de establecer cada cosa en un molde global. Un buen sistema de software está basado en un conjunto reducido de ideas poderosas; aún si contiene muchas funciones especializadas, estas son derivadas como consecuencia de dichos conceptos básicos. Las sub características de la "Funcionalidad" son: Aptitud, Precisión, Interoperatividad, Conformidad, Seguridad y Trazabilidad.

Confiabilidad. La fiabilidad o confiabilidad del software es la probabilidad que un programa realice su objetivo satisfactoriamente (sin fallos) en un determinado periodo de tiempo y en un entorno concreto (denominado perfil operacional). No hay duda que la fiabilidad de un programa de computadora es un elemento importante de su calidad general. Si un programa falla frecuentemente en su funcionamiento, no importa si el resto de los factores de calidad son aceptables.

La fiabilidad del software, a diferencia de otros factores de calidad, puede ser medida o estimada mediante datos históricos o de desarrollo. La fiabilidad del software se define en términos estadísticos como la probabilidad de operación libre de fallos de un programa de computadora en un entorno determinado y durante un tiempo específico.

Los modelos de fiabilidad del software entran en dos grandes categorías: 1. Modelos que predicen la fiabilidad como una función cronológica del tiempo (calendario); 2. Modelos que predicen la fiabilidad como una función del tiempo de procesamiento transcurrido (tiempo de ejecución de CPU).

Las sub características de la "Confiabilidad" son: Madurez, Tolerancia a fallas, Facilidad de Recuperación, Disponibilidad y Degradabilidad.

Facilidad de uso. La facilidad de uso consiste en la simplicidad con que la gente puede aprender a usar el software y aplicarlo para resolver problemas. También incluye la facilidad de instalación, operación y monitoreo. La definición acentúa los diferentes niveles de experiencia de los usuarios potenciales. Este requerimiento representa uno de los mayores retos para los diseñadores de software interesados en facilidad de uso: cómo proveer orientación a los usuarios novatos y cómo no aburrir a los usuarios expertos, al mismo tiempo.

El proceso de facilidad de uso consiste en un ciclo de diseño-evaluación-rediseño con un análisis previo donde se definen los niveles de facilidad de uso que se desean alcanzar. Este proceso asegura que el producto de software obtenido está cercano al óptimo en cuanto a facilidad de uso, entendiendo por óptimo los niveles deseados de facilidad de uso.

Este proceso ayuda a conseguir las respuestas a las preguntas sobre los usuarios y las tareas que desempeñan en la fase de análisis, y da una base sobre la que se realiza el diseño de la interacción en la fase de diseño.

La medición de facilidad de uso utiliza cinco escalas diferentes y no incluye utilidad. Ellas son: tiempo de la tarea, errores, aprendizaje, aprender de nuevo, satisfacción y logro de la meta.

Las sub características de la Facilidad de Uso son: comprensibilidad, facilidad de aprendizaje, operatividad, explicitud, adaptabilidad al usuario, atractividad, claridad, facilidad de ayuda y amistoso al usuario.

Eficiencia. La eficiencia es la habilidad del software para poner la cantidad mínima de demanda sobre los recursos de hardware como sea posible, tales como el tiempo de procesador, espacio ocupado en memorias internas o externas, ancho de banda usado en dispositivos de comunicación. Normalmente se identifica eficiencia como velocidad de ejecución. Esta interpretación no es correcta. El software es eficiente si realiza un uso racional de todos los recursos de hardware. La eficiencia incluye la utilización equilibrada de: tiempo de CPU, memoria principal, memoria secundaria, canales de entrada/salida, velocidad de ejecución y tiempo de respuesta.

El tema de la eficiencia debe estar balanceado con otros objetivos como la extensibilidad y la reutilización. Sin embargo, no hay que disminuir la importancia de la eficiencia puesto que nadie quiere estar esperando demasiado las respuestas del sistema o verse obligado a estar comprando más memoria para ejecutar un programa. Estos aspectos muestran que la Ingeniería de Software es una actividad compleja, ya que requiere tomar en cuenta muchos requerimientos, algunos de los cuales, como exactitud, son abstractos y conceptuales, mientras que otros como eficiencia, son concretos y ligados a las propiedades del hardware.

Las sub características de la “Eficiencia” son: Respecto al tiempo y Respecto a los recursos.

Facilidad de Mantenimiento. El mantenimiento del software representa más esfuerzo que cualquier otra actividad de la Ingeniería del Software. La facilidad de mantenimiento es la facilidad con la cual se puede corregir un programa si se encuentra un error, adaptarlo si su entorno cambia, o mejorarlo si el cliente desea un cambio en los requisitos. Una métrica sencilla orientada al tiempo es el tiempo medio entre cambios (TMEC), el tiempo que lleva analizar el cambio requerido, diseñar una modificación apropiada, implementar el cambio, probarlo y distribuirlo a todos los usuarios. Generalmente, los programas fáciles de mantener tendrán un menor TMEC que los programas que no son fáciles de mantener para tipos de cambios equivalentes.

Una definición amplia de facilidad de mantenimiento es la de "facilidad de comprender, corregir, adaptar y mejorar el software”. Existen tres tipos de mantenimiento: mantenimiento correctivo: corregir errores, mantenimiento adaptativo: modificar el software de acuerdo con el entorno; y mantenimiento perfectivo: añadir nueva funcionalidad. El mantenimiento preventivo no está tan extendido y consiste en cambiar el producto pensando en mejoras futuras. Las sub características de la Facilidad de Mantenimiento son: Facilidad de análisis, Facilidad de cambio, Estabilidad y Facilidad de prueba.

Portabilidad. La portabilidad consiste en la facilidad de transportar productos de software a varios ambientes de hardware y software. Esta característica se ocupa de variaciones no solo del hardware sino de algo más general, la combinación hardware-software, que es la máquina que en realidad programamos, que incluye el sistema operativo, el sistema gráfico, y otras

herramientas fundamentales. El término plataforma es usado para denotar un tipo de máquina hardware-software. Las sub características de la “Portabilidad” son: Adaptabilidad, Facilidad de instalación, Conformidad y Facilidad de reemplazo.

La ISO/IEC 9126 está formada por:

Parte 1 – Modelo de Calidad,

Parte 2 – Métricas Externas,

Parte 3 – Métricas Internas y

Parte 4 – Calidad en Uso.

2.2.1.2.2.5. ISO/IEC 25000:2005 – SquaRE

SQuaRE (Software Quality Requirements and Evaluation) es una nueva serie de normas que se basa en ISO 9126 y en ISO 14598 (Evaluación del software). Uno de los principales objetivos de la serie SQuaRE es la coordinación y harmonización del contenido de ISO 9126 y de ISO 15939:2002 (Measurement Information Model). ISO 15939 tiene un modelo de información que ayuda a determinar que se debe especificar durante la planificación, performance y evaluación de la medición, Para su aplicación, cuenta con los siguientes pasos: recopilar los datos, preparación de los datos y Análisis de los datos.

La integración de ISO 9126 e ISO 15939 permiten plantear un proceso de 4 pasos: (1) Identificación de los requerimientos relacionados a la calidad del producto, es decir, seleccionar la parte del modelo de calidad (ISO/IEC 9126-n) que resulta relevante para la evaluación de calidad; (2) Identificación del contexto de interpretación, es decir, selección de los valores de referencia y determinación de los target especificados en un contexto determinado; (3) Uso de las medidas derivadas de la etapa de preparación de los datos; (4) comparación y análisis de los resultados obtenidos respecto de un conjunto de valores de referencia.

SQuaRE incluye un estándar de requerimientos de calidad. Está compuesto por 14 documentos agrupados en cinco (5) tópicos: (1) Administración de la Calidad – 2500n, (2) Modelo de Calidad – 2501n, (3) Medidas de Calidad – 2502n, (4) Requerimientos de Calidad – 2503n y (5) Evaluación de la Calidad – 2504n.

Los beneficios de utilizar SQuare son: (1) El modelo representa la calidad esperada del producto de software, (2) Plantea el desdoblamiento de las necesidades o expectativas en calidad en uso, calidad externa y calidad interna, (3) Permite una mayor eficacia en la definición del software, (4) Plantea la evaluación de productos intermedios, (5) Propone una calidad final a través de las evaluaciones intermedias, (6) Permite efectuar un rastreo entre las expectativas, requisitos y medidas de evaluación; y (7) Mejora la calidad del producto.

Estos últimos son los estándares que fueron presentados por Scalone (Scalone, 2006) los cuales nos sirvieron de referencia en esta investigación.

Para la presente investigación se consideran las pruebas como punto central de la investigación, y que son la clave para la metodología que se espera proponer, por lo que a continuación se realiza una revisión del tema.

2.2.2. Pruebas de Software

Las pruebas constituyen el proceso de encontrar diferencias entre el comportamiento esperado, especificado por los modelos del sistema y el comportamiento observado del sistema (Bruegge y Dutoit, 2002). Las pruebas son un intento de simulación del sistema con respecto a los modelos del mismo. Su objetivo es diseñar situaciones que manifiesten los defectos que hay en el sistema y que revelen los problemas. Esta actividad es contraria a todas las actividades del desarrollo de software, pues están orientadas al quebrantamiento del sistema, es decir, lograr que el sistema falle (demostrar la presencia de defectos) de forma planeada. Méndez (Mendez, 2006) lo resume de la siguiente forma: Una diferencia interesante entre las Pruebas y otras disciplinas del proceso de desarrollo de software es que esencialmente es una tarea que encuentra y pone de manifiesto las debilidades del producto de software. Esto puede lograrse descartando ideas preconcebidas sobre la corrección del software que se acaba de desarrollar y

se supere cualquier conflicto de intereses que aparezcan cuando se descubran los errores presentes (Pressman, 2005).

Según Parada (Parada, 2010) una actividad que representa un alto nivel de complejidad es la definición de la metodología de pruebas a aplicar. Al respecto los estándares del dominio de pruebas proponen un conjunto de buenas prácticas y enuncian las fases y los ciclos de actividades que estas deberían tener, pero no definen los procedimientos y las técnicas que las implementan. Por ello es necesario identificar los artefactos y las relaciones entre las fases de pruebas y las fases del ciclo de desarrollo, así como las decisiones que se derivan de los resultados de las pruebas y como usarlas para controlar el proceso.

Probar es una actividad realizada para la evaluación de la calidad en productos y para mejorarla, mediante la identificación de los defectos y problemas. Este concepto toma fuerza sobre el hecho de que la disciplina de pruebas debe estar formalizada y organizada mediante una estrategia dentro de un proyecto de desarrollo de software, dentro de un alcance definido, para que debidamente organizado se logren los objetivos planteados. Esta planificación de pruebas y actividades de diseño en sí mismos, constituyen un aporte útil para los diseñadores al poder exponer las debilidades potenciales (como descuidos o contradicciones de diseño, omisiones o ambigüedades en la documentación).

En líneas generales, las pruebas de un software giran alrededor de los casos de prueba, que según Leffingwell y Widrig (2006) son esenciales para todas las actividades de prueba porque:

- Son la base para diseñar y ejecutar los procedimientos de pruebas.
- La profundidad de las pruebas es proporcional al número de casos de pruebas.
- El diseño y desarrollo, y los recursos necesarios son gobernados por los casos de pruebas requeridos.

Bajo esta premisa, los casos de prueba (especificaciones de un conjunto de entradas, condiciones de ejecución e identificación de resultados los resultados esperados producto de las combinaciones de los dos primeros elementos), determinarán el ambiente necesario para ser ejecutado, podrá demostrar defectos (o su contraparte, casos de éxito) al final del proceso

de pruebas apoyándose o no en herramientas de pruebas que lo gestionen formalmente, dirigido a diversos tipos de prueba y técnicas aplicadas según corresponda, en los diversos niveles de prueba que sean determinados por el equipo del proyecto/software objeto de prueba mediante procedimientos (instrucciones) de prueba claramente establecidos y documentados siguiendo una estrategia de pruebas plasmada en un plan de prueba.

Profundizando algunos conceptos, se tienen las siguientes definiciones tomadas en su mayoría de RUP (2005).

Caso de Prueba: Define un conjunto de entradas de prueba, condiciones de ejecución, y resultados esperados, identificados con el objetivo de evaluar algunos aspectos particulares de un elemento destino de la prueba.

Caso de Uso: Una especificación de un tipo de interacción con el sistema. Define un conjunto de instancias de guión de uso, donde cada instancia es una secuencia de acciones que lleva a cabo un sistema que producen un resultado observable de valor para un actor concreto.

Plan de prueba: Define los objetivos de las pruebas en el ámbito de la iteración (o el proyecto) los elementos de destino, el enfoque que se adopta, los recursos necesarios y los entregables que se deben generar.

Srcipts de Pruebas: Es un guión o conjunto de instrucciones paso a paso que realiza una prueba, habilitando su ejecución. Permiten automatizar tareas creando pequeñas utilidades.

Suite de Pruebas: Es un conjunto de Scripts que se agrupan con el objetivo de gestionar y secuenciar la ejecución de las pruebas ordenadamente.

Datos de prueba: Define una recopilación de valores de entrada de prueba que se consumen durante la ejecución de una prueba, y los resultados esperados que sirven de referencia para objetivos comparativos durante la ejecución de una prueba.

Condiciones: Descripción de una condición que ejercerá durante esta prueba.

Resultados de las pruebas: Resume el análisis de uno o más registros de prueba y solicitudes de cambio, proporcionando una valoración relativamente detallada de la calidad de los elementos destino de la prueba y el estado del esfuerzo de prueba.

Resultados esperados: Corresponde a las expectativas que se tienen de los resultados.

Resultados obtenidos: Corresponde a lo realmente obtenido como resultado. Si lo que se espera es igual al obtenido, entonces el caso de prueba se considera exitoso.

Ambiente de pruebas: Respecto a la aplicación a probar, se puede incluir el hardware sobre el que se ejecuta, el software necesario para interconectarse con otros sistemas o requerimientos de la aplicación propiamente dicho, la base de datos necesaria para operar la aplicación, entre otros entes específicos.

Herramientas de pruebas: Aquellas que se pueden utilizar para aumentar la eficacia de las pruebas.

Tipos de Prueba: Tienen un objetivo específico y una técnica de soporte. Cada tipo de prueba se centra en la prueba de uno o varios atributos o características del destino de la prueba. SWEBOK (2004) plantea que los casos de prueba pueden ser diseñados para comprobar que las especificaciones funcionales se apliquen correctamente, que son comúnmente conocidas como pruebas de conformidad, pruebas de precisión o pruebas funcionales. Sin embargo, muchas otras propiedades no funcionales pueden ser probadas tales como el rendimiento, fiabilidad y facilidad de uso, entre otros. En todo caso, el objetivo de la prueba varía de acuerdo a la aplicación objeto de prueba.

De esta forma algunos tipos de prueba se clasifican según la dimensión de calidad en Funcionales, No funcionales, de la siguiente forma:

- Dimensión de funcionalidad: Prueba de función, prueba de seguridad y prueba de volumen.
- Dimensión de utilidad: Pruebas de utilización, prueba de integridad, prueba de usabilidad.

- Dimensión de fiabilidad: Prueba de estructura y de tensión.
- Dimensión de rendimiento: Prueba de puntos de referencia, Prueba de contienda, Prueba de carga y perfil de rendimiento, prueba back to back.
- Dimensión de Capacidad de soporte: Prueba de configuración, prueba de instalación, pruebas de recuperación.
- Otras: Pruebas alpha y beta, pruebas de regresión,

Niveles de Pruebas: Las pruebas se aplican a diferentes tipos de destinos, en fases o niveles diferentes de esfuerzo de trabajo. Estos niveles suelen distinguirlos los roles que están más capacitados para diseñar y dirigir las pruebas. Los niveles de prueba son conocidos como: Prueba de desarrollador (unitarias), Pruebas integrales, Pruebas certificadas y Pruebas del sistema (aceptación del usuario).

Técnicas de Pruebas: Normalmente las técnicas de pruebas se clasifican en Caja blanca (que proporcionan información del código que fue diseñado) y Caja negra (que se centran solo en el comportamiento de entrada y salida del software). Sin embargo, SWEBOK (2004) plantea que es difícil encontrar una base homogénea para clasificar las técnicas de pruebas, pero plantea una clasificación que se basa en el cómo se generan los casos de prueba y se presenta a continuación:

a) Intuición y experiencia de los ingenieros de software.

a. Pruebas Ad hoc

b. Pruebas exploratorias

b) Las especificaciones del software

a. Partición de equivalencia

b. Análisis del valor frontera

c. Tabla de decisión

d. Máquinas basadas en estados finitos

e. Pruebas desde especificaciones formales

f. Pruebas aleatorias

c) La estructura del código

a. Criterio basado en el flujo de control

b. Criterio basado en el flujo de datos

c. Modelos de referencia para pruebas basadas en código

d) Técnicas basadas en fallos (reales o artificiales-simuladas)

a. Adivinar el error

b. Pruebas de mutación

e) La naturaleza de la aplicación

a. Pruebas orientadas a objetos

b. Pruebas basados en componentes

c. Pruebas de GUI

d. Pruebas de programas concurrentes

e. Pruebas de protocolo de conformidad

f. Pruebas de sistemas en tiempo real

g. Pruebas de sistemas de seguridad crítica

Muchas de las pruebas indicadas arriba están incluidas dentro de las técnicas de pruebas dinámicas, las cuales serán utilizadas en esta investigación. Las pruebas dinámicas se

clasifican en: (a) basadas en especificaciones, (b) basadas en estructura y (c) basadas en experiencia.

a) Basadas en la especificación: Son conocidas como técnicas de pruebas de caja negra o conducidas por entradas/salidas, porque tratan el software como una caja negra con entradas y salidas, pero no tienen conocimiento de cómo está estructurado el programa o componente dentro de la caja. Esencialmente, el técnico de pruebas se concentra en qué hace el software y no en cómo lo hace. Las técnicas basadas en especificaciones obtienen los casos de prueba directamente de las especificaciones o de otros tipos de artefactos que contengan lo que el sistema debería hacer.

   A continuación, se detallarán las técnicas basadas en la especificación más comunes.

   - **Particionamiento de equivalencia**: Esta técnica consiste en dividir un conjunto de condiciones de prueba en grupos o conjuntos que puedan ser considerados iguales por el sistema. Esta técnica requiere probar sólo una condición para cada partición. Esto es así porque se asume que todas las condiciones de una partición serán tratadas de la misma manera por el software.

   - **Análisis de valor frontera**: El análisis del valor frontera está basado en probar los valores frontera de las particiones. En esta técnica, también, se cuenta con fronteras válidas (en las particiones válidas) y fronteras no válidas (en las particiones no válidas).

   - **Tablas de decisión**: Una tabla de decisión lista todas las condiciones de entrada que pueden ocurrir y todas las acciones que pueden surgir de ellas. Están estructuradas por filas, con las condiciones en la parte de arriba de la tabla y las posibles acciones en la parte de abajo. Las reglas de negocio, que incluyen combinaciones de condiciones para producir algunas combinaciones de acciones, se incluyen en la parte de arriba de un extremo a otro. Cada columna, por lo tanto, representa una regla de negocio individual y muestra cómo se combinan las condiciones de entrada para producir acciones. De esta manera, cada columna representa un posible caso de prueba, ya que identifica ambas entradas y salidas.

   - **Transición de estados**: La técnica de transición de estados se utiliza con sistemas en los que las salidas son desencadenadas por cambios en las condiciones de entrada, o

cambios de “estado”. Las pruebas de transición de estados se usan cuando alguno de los aspectos del sistema se pueden describir en lo que se denomina una “máquina de estados finitos”. Esto significa que el sistema puede estar en un número (finito) de estados, y las transiciones de un estado a otro están determinadas por las reglas de la “máquina

- **Pruebas de casos de Uso**: Las pruebas de caso de uso son una técnica que ayuda a identificar casos de prueba que ejerciten el sistema entero transición a transición desde el principio al final. Un caso de uso es una descripción de un uso particular del sistema por un actor (un usuario del sistema). Cada caso de uso describe las interacciones que el actor tiene con el sistema para conseguir una tarea concreta (o, al menos, producir algo de valor al usuario). Los casos de uso pueden descubrir defectos de integración, defectos causados por la interacción incorrecta entre diferentes componentes. Los casos de uso describen el flujo de proceso a través de un sistema basado en su uso más probable. Esto hace que los casos de prueba obtenidos de los casos de uso sean particularmente útiles a la hora de encontrar defectos en el uso real del sistema

b) Basadas en la estructura: Las pruebas estructurales son una aproximación al diseño de casos de prueba donde las pruebas se derivan a partir del conocimiento de la estructura e implementación del software. Esta aproximación se denomina a veces pruebas de caja blanca. A continuación, se detallarán las técnicas basadas en la estructura más comunes.

- **Pruebas de sentencia**: El objetivo de las pruebas de sentencia es ir probando las distintas sentencias a lo largo del código. Si se prueban todas y cada una de las sentencias ejecutables del código, habrá una cobertura de sentencia total.

- **Pruebas de decisión**: El objetivo de estas pruebas es asegurar que las decisiones en un programa son realizadas adecuadamente. Para probar una decisión, es necesario ejecutarla cuando la condición que tiene asociada es verdadera y también cuando es falsa. De esta forma, se garantiza que todas las posibles salidas de la decisión se han probado.

- **Pruebas de caminos**: Las pruebas de caminos son una estrategia de pruebas estructurales cuyo objetivo es probar cada camino de la ejecución independientemente.

Las pruebas de caminos no prueban todas las posibles combinaciones de todos los caminos en el programa.

c) Basadas en la experiencia: Las técnicas basadas en experiencia son aquellas a las que se recurre cuando no hay una especificación adecuada desde la que crear casos de prueba basados en especificación, ni hay tiempo suficiente para ejecutar la estructura completa del paquete de pruebas.

- **Adivinar errores**: Las técnicas basadas en experiencia usan la experiencia de los usuarios y de los técnicos de pruebas para determinar las áreas más importantes de un sistema y ejercitar dichas áreas de forma que sean consistentes con el uso que se espera que tengan.

- **Pruebas exploratorias**: Técnicas de pruebas más sofisticadas que se realizan sobre la base del conocimiento y experiencia de los técnicos de pruebas; dicha base es un factor decisivo para el éxito de las pruebas.

2.2.2.1. **Disciplina de Pruebas**

Mendoza (Mendoza, 2010) señala "*que los modelos de ciclo de vida del software hacen que las actividades para su desarrollo y sus dependencias sean visibles y manejables*". Por su parte RUP indica que una disciplina es una colección de actividades correspondidas que a su vez se relacionan con un "motivo de preocupación importante" dentro del proyecto total donde estén enmarcadas.

Mendoza (Mendoza, 2010) indica que "La disciplina de pruebas en el software, está fuertemente ligada a la calidad y los esfuerzos que se puedan ejecutar para alcanzarla, garantizando que los procesos y productos resultarán con los niveles esperados de calidad". Por su parte RUP menciona que su importancia radica en que un acercamiento continuo a la calidad, iniciada temprano en el ciclo de vida del software, puede aminorar los costos de terminar y mantener el software significativamente. Esto reduce altamente el riesgo asociado de implantar software de mala calidad (Rational, 2003).

Al mismo tiempo, (Kruchten, 2000) afirma que la disciplina de pruebas en un proceso iterativo que ocurre en todas las fases del ciclo de vida, permitiendo tener una

retroalimentación temprana sobre la calidad del producto que será usada para mejorar el producto tanto en su diseño como en su construcción. Con todo lo mencionado anteriormente se puede evidenciar que la actividad de probar no es una actividad final del ciclo de vida del sistema, sino que debe ser abordado desde la fase del desarrollo. Mendoza (Mendoza, 2010) señala que por su parte, involucrando el capital humano que participa en la disciplina de pruebas apoya el hecho de iniciar pruebas desde etapas tempranas incluyendo a los probadores quienes deben participar desde el comienzo de un ciclo de vida del proyecto, para que puedan comprender exactamente lo que están probando y puedan trabajar en unión con los stakeholders para crear requerimientos comprobables. Se entiende como stakeholders, cualquier persona o representante de una organización quien tiene un interés creado en los resultados de un proyecto o cuya opinión debe ser complacida. (Kruchten, 2000).

Mendoza (Mendoza, 2010) señala que la disciplina de pruebas actúa como abastecedor de servicios a las otras disciplinas en muchos aspectos. Las pruebas se centran sobre todo en la evaluación o evaluar la calidad del producto, que es lograda con estas prácticas base (Rational, 2003):

- Encontrar y documentar los defectos en calidad del software.
- Aconsejar sobre la calidad percibida del software.
- Validar y probar las asunciones hechas en especificaciones del diseño y del requerimiento a través de demostración concreta.
- Validar que el producto de software trabaja según lo diseñado.
- Validar que los requerimientos están implementados apropiadamente.

Podemos evidenciar la importancia de la disciplina de pruebas, ya que existen diferentes metodologías y tendencias en el desarrollo de software (UP, XP (eXtream Programing), MFS, Eclipse EFP, ITIL, entre otros) las cuales hacen uso de ella.

En este trabajo se adopta el enfoque dinámico de las Pruebas, tomando como definición de Prueba la de SWEBOK (IEEE, Swebok: Guide to the Software Engineering Body of Knowledge, 2004). En SWEBOK se definen Prueba, Prueba de Software, Verificación dinámica y comportamiento esperado de la siguiente manera:

Prueba es una actividad realizada para evaluar la calidad del producto y mejorarla, identificando defectos y problemas.

La Prueba de software es la verificación dinámica del comportamiento de un programa contra el comportamiento esperado, usando un conjunto finito de casos de prueba, seleccionados de manera adecuada desde el dominio infinito de ejecución.

Dinámica: implica que para realizar las pruebas hay que ejecutar el programa para los datos de entrada.

El comportamiento esperado: debe ser posible decidir cuando la salida observada de la ejecución del programa es aceptable o no, de otra forma el esfuerzo de la prueba es inútil. El comportamiento observado puede ser revisado contra las expectativas del usuario, contra una especificación o contra el comportamiento anticipado por requerimientos implícitos o expectativas razonables.

2.2.2.2. **Verificación, Validación y Prueba**

Los términos Verificación y Validación suelen usarse indistintamente en algunos contextos. Para las Pruebas del Software ambos términos indican conceptos diferentes:
La definición dada por (IEEE 610.12-1990) de ANSI/IEEE, 1990 de Verificación y Validación es la siguiente:

- Verificación: proceso de evaluación de un sistema o componente para determinar si un producto de una determinada fase de desarrollo satisface las condiciones impuestas al inicio de la fase.

- Validación: proceso de evaluación de un sistema o componente durante o al final del proceso de desarrollo para determinar cuándo se satisfacen los requerimientos especificados

Boehm (1984) usa dos preguntas para clarificar la diferencia entre Verificación y Validación

- Verificación: ¿Estamos elaborando correctamente el producto?
- Validación: ¿Estamos elaborando el producto correcto?

CMMI (2006) define el propósito de la Verificación y la Validación de la siguiente manera:

- El propósito de la Verificación es asegurar que los Productos Internos seleccionados cumplen con su especificación de requerimientos. Los métodos de verificación pueden ser, entre otros: inspecciones, revisiones por pares, auditorias, recorridas, análisis, simulaciones, pruebas y demostraciones.
- El propósito de la Validación es demostrar que un producto o componente de producto cumple su uso previsto cuando es puesto en su ambiente previsto. Deben ser seleccionados los productos internos (por ejemplo: requerimientos, diseño, prototipos) que mejor indican cuán bien el producto y los productos internos deben satisfacer las necesidades del usuario.

En general en la bibliografía, la Prueba es tomada como una parte del proceso de Verificación y Validación y bajo el entendido de que la Prueba requiere ejecutar el código. En este trabajo se utiliza el enfoque de SWEBOK (2004), donde se define que: La salida observada de la ejecución del programa puede ser comparada mediante:

- Prueba para la verificación: El comportamiento observado es revisado contra las especificaciones.
- Prueba para la validación: El comportamiento observado es revisado contra las expectativas del usuario

Alguna de las tendencias más actualizadas en pruebas se conoce como Desarrollo dirigido por pruebas TDD (Test Driven Developmet o Test Driven Design, por sus siglas en ingles), técnica de desarrollo de software que consiste en dos prácticas: a) Escribir los casos de pruebas (para pruebas unitarias) primero antes que el código que implementará dicho caso de prueba con el fin de que pase con éxito y b) Refactorizar (mejorar el diseño del código, elevando la flexibilidad y tolerancia al cambio, entregando más rápido y con menos depuración) incrementalmente. Esta es una de las prácticas clave en la Programación extrema, pero no necesariamente se implementa junto a ella (Astels, 2003) y su objetivo es "código limpio que funciona". Los demás tipos de pruebas, en sus diferentes niveles, no se reemplazan, solo se complementan.

Un enfoque interesante en la disciplina de pruebas, es presentado en el Perfil de Pruebas de UML (UML Testing Profile) que define un lenguaje para el diseñar, visualizar, especificar, analizar, construir y documentar los artefactos en las pruebas de sistemas en varios dominios (OMG, 2005). Está basado en la especificación UML 2.0, uniendo el meta-modelo UML y re-usando la sintaxis UML, extendiendo a UML con conceptos específicos en pruebas agrupados en 4 categorías: Arquitectura de prueba (que contiene los elementos que participan en una prueba y sus relaciones, conceptos de estructura y configuración de las pruebas), Datos de prueba (estructuras y significado de los valores que serán procesados en una prueba), Comportamiento de pruebas (que se ocupa de las observaciones y actividades durante una prueba) y tiempo (conjunto de conceptos relativos al tiempo con el fin de cuantificar las ejecuciones de pruebas).

Este perfil está enfocado a una particular tecnología de pruebas llamada "Pruebas funcionales de Caja negra", permitiendo la captura de toda la información necesaria para el enfoque de caja negra y así evaluar la correcta implementación del sistema. El perfil de pruebas UML se destina a apoyar de una manera efectiva, eficiente, en la medida posible, las pruebas automatizadas del sistema en función de las implementaciones de los modelos computacionales de UML (Cavarra, 2002).

Este perfil constituye una iniciativa importante en la formalidad de la disciplina de pruebas dentro del proceso de desarrollo de software, bajo el enfoque de la validación de sistemas de

tipo caja negra (solo entradas/salidas), pues detalla las notaciones, documentación y pasos que deben cumplirse al momento de probar sistemas. Algunos tecnicismos son aplicables a proyectos basados en la notación UML y apoyan la automatización de pruebas, además de que puede acoplarse con un marco de trabajo como JUnit usado dentro de la popular programación extrema (XP). El JUnit es un marco de trabajo para que los desarrolladores realicen pruebas unitarias en Java inicialmente, pues ya se ha traducido a varios lenguajes.

Sin embargo, es importante notar el amplio alcance en los diferentes niveles de pruebas, puesto que se pueden especificar casos de prueba a nivel de unidad, integración, sistema y a nivel de la organización.

2.2.2.3. **Fases de las Pruebas**

A continuación se presentan las fases de las pruebas definidas por distintos autores en el ciclo de vida específico para las Pruebas, en las cuales se presentan actividades de planificación, diseño y las de ejecución de las pruebas, las cuales nos fueron presentadas por el autor Beatriz Pérez (Perez L., 2006),

Edward Kit (Kit, 1995) define las etapas del proceso de prueba y sus principales actividades. Rex Black (Black, 2002) se focaliza en la planificación de las pruebas, definiendo las fases sin entrar en detalle en sus actividades. Cem Kaner, James Bach y Bret Pettichord (Kaner, Bach, & Pretichord, Lecciones aprendidas en Pruebas de Software, 2001) recopilan 293 lecciones aprendidas sobre las pruebas, sin organizarlas en un proceso. Hetzel (Hetzel, 1998) define las fases de prueba acompañando el ciclo de vida del software. Cem Kaner, Jack Falk y Hung Quoc Nguyen (Kaner, Falk, & Nguyen, Testing Computer Software, 2da edición ISBN 0471358460, 1999) describen las técnicas de prueba que son útiles en cada etapa del ciclo de vida del software, sin agruparlas en fases o definiendo actividades específicas. James A. Whittaker (Whittaker, 2000)y el International Software Testing Qualifications Board (ISTQB, 2013) definen las fases del proceso de prueba, sin entrar en el detalle de las actividades que las componen. TMap (Pol, Teunissen, & van Veenendaal, 2002) es el proceso de prueba definido por la empresa Sogeti, define las fases y el detalle de las actividades de prueba.

Edward Kit (Kit, 1995) divide las actividades de prueba en las siguientes fases: Planificación, Diseño de la Arquitectura de las pruebas, Desarrollo del Testware, Ejecución de las pruebas, Evaluación de las pruebas y Mantenimiento del Testware. A continuación se detalla cada una de estas fases:

- **Planificación de las pruebas**: Se realiza un plan global para todas las actividades de prueba (unitario, integración, usabilidad, funcional, del sistema y de aceptación). El objetivo del plan de pruebas es definir el alcance, enfoque, recursos y agenda de las actividades de prueba.
- **Diseño de la arquitectura de las pruebas**: Es el proceso de diseñar una estructura para las pruebas. Se define cómo se organizan, se categorizan y se estructuran las pruebas y el repositorio de pruebas. Los principales puntos a definir incluyen: Organización de las pruebas respecto a su fuente (requerimientos, funciones, etc.), categorización de las pruebas y convenciones para agruparlas, estructura y convenciones de nombres para el repositorio de pruebas, agrupación de las pruebas según períodos de ejecución.
- **Desarrollo del Testware**: El testware es diseñado para cumplir con los objetivos de detectar tantos errores como sea posible y minimizar el costo de desarrollar las pruebas, de ejecutarlas y de mantenerlas.

  Las tareas para desarrollar el testware son:

  <u>Diseño detallado</u>: Es el proceso de especificar los detalles del enfoque de las pruebas para los ítems de prueba e identificar los casos de prueba asociados. Los pasos para realizar esta tarea incluyen: identificar los ítems que deben ser probados, asignar prioridades a esos ítems basándose en los riesgos, desarrollar el diseño de las pruebas de alto nivel para grupos de pruebas relacionados, desarrollar casos de pruebas individuales a partir del diseño de alto nivel.

  <u>Implementación</u>: Es el proceso de llevar cada especificación a un caso de prueba pronto para ser ejecutado. Los entregables de salida son: Casos de prueba, datos de prueba, Especificaciones de los procedimientos de prueba y Matriz de cubrimiento.

- **Ejecución de las pruebas**: Proceso de ejecutar todos los casos de prueba seleccionados y observar los resultados. La ejecución de las pruebas incluye las siguientes tareas: Selección de los casos de prueba, configuración, ejecución, análisis posterior, registro de actividades, resultados e incidentes, determinar si las fallas fueron causadas por errores en el producto o en las pruebas, medir el cubrimiento lógico interno. Los principales entregables de la ejecución son: test logs, reportes de incidentes y reporte de cubrimiento lógico.
- **Evaluación de las pruebas**: La evaluación incluye:

  Evaluación del cubrimiento de las pruebas: Se evalúan los casos de prueba en su conjunto y se decide si es necesario o no realizar más pruebas, se estudia el cubrimiento en varios niveles: funciones, requerimientos, lógica.

  Evaluación de los errores del producto: Evaluar la calidad del producto respecto a la ejecución de las pruebas realizadas.

  Evaluación de la efectividad de las pruebas: Evaluar las pruebas respecto al criterio de completitud y decidir cuándo parar o agregar más pruebas y seguir, se basa en los dos anteriores.

Rex Black (Black, 2002) divide el esfuerzo de prueba en las siguientes fases: Planificación, Configuración, Desarrollo de las pruebas y Ejecución de las pruebas, las cuales se describen a continuación:

- **Planificación**: En esta etapa se realiza la planificación del proyecto de prueba, incluye las actividades de definición de riesgos, estudio del cronograma y del presupuesto, escribir el plan de pruebas y seleccionar las herramientas para las pruebas.
- **Configuración**: Incluye obtener el hardware y otros recursos necesarios e instalación del laboratorio de pruebas.
- **Desarrollo de las Pruebas**: Instalar las herramientas de prueba y de reportes, crear las suites de prueba y las librerías de casos de prueba y documentar como es el proceso de prueba que pone los elementos del testware en acción.

- **Ejecución de las Pruebas**: Ejecutar las pruebas, registrar su estado y reportar resultados.

Kaner, Bach y Prettichord (Kaner, Bach, & Pretichord, Lecciones aprendidas en Pruebas de Software, 2001) definen que la prueba involucra al menos las siguientes cuatro actividades:

- **Configurar**: Preparar el producto para las pruebas. Ponerlo en el estado de comienzo.
- **Operar**: Ingresar los datos al producto y darle los comandos. Interactuar con él de alguna manera estipulada. De otra forma, no se realiza prueba sino revisión.
- **Observar**: Recolectar la información acerca de cómo el producto se comporta, datos de salida, el estado del sistema, interacción con otros productos, etc.
- **Evaluar**: Aplicar reglas, razonamiento o mecanismos que podrían detectar bugs en los datos que se observan.

Bill Hetzel (Hetzel, 1998) divide el esfuerzo de prueba en las siguientes fases, que ocurren en paralelo con el desarrollo:

- **Análisis**: Se planifica y definen los requerimientos y objetivos de las pruebas.
- **Diseño**: Se especifican las pruebas a ser desarrolladas.
- **Implementación**: Se construyen los procedimientos y los casos de prueba.
- **Ejecución:** Se ejecutan y reejecutan las pruebas.
- **Mantenimiento**: Se guardan y modifican las pruebas a medida que el software cambia.

La integración de la prueba con el ciclo de vida del software se describe a continuación:

- Comienzo del Proyecto: Desarrollar la estrategia de pruebas, establecer el enfoque y esfuerzo de pruebas.
- Requerimientos: Establecer los requerimientos para las pruebas, asignar responsabilidades para probar, diseñar los procedimientos de prueba preliminares y las pruebas basadas en requerimientos, validar los requerimientos.

- Diseño: Preparar un plan del sistema preliminar, completar el plan de pruebas de aceptación y la especificación del diseño, completar las pruebas basadas en el diseño, validar el diseño.
- Desarrollo: Completar el plan de pruebas del sistema, finalizar los procedimientos de prueba y pruebas basadas en el código, completar el diseño de pruebas unitarias o de módulos, probar los programas, integrar y probar los subsistemas, realizar las pruebas del sistema.
- Implantación: Realizar las pruebas de aceptación, probar los cambios y arreglos, evaluar la efectividad de las pruebas.

James Whittaker (Whittaker, 2000) define 4 fases de las pruebas, que permiten a los testers una estructura para agrupar problemas relacionados que deben resolver antes de seguir con la fase siguiente. Las fases definidas son:

- **Fase 1 - Modelar el ambiente del software**: Los testers deben identificar y simular las interfaces que un sistema de software usa y enumerar las entradas que cruzan cada interfase. Los testers deben seleccionar, valores para cada variable de entrada y decidir cómo secuenciar las entradas.
- **Fase 2 - Seleccionar los escenarios de prueba**: Muchos modelos de dominio y partición de variables representan un número infinito de escenarios de prueba, cada uno cuesta tiempo y dinero. Se debe aplicar un criterio de adecuación de los datos de prueba, que debe ser el representante adecuado y económico de todos los posibles test. En general el más adecuado es aquel que encuentra mayor cantidad de defectos.
- **Fase 3 - Ejecutar y evaluar los escenarios de prueba**: Teniendo identificado el conjunto de pruebas, los testers deben ejecutarlos y evaluar los resultados. La ejecución de las pruebas puede ser manual o automatizada. La evaluación de los escenarios involucra la comparación de la salida real, resultante de la ejecución del escenario con su salida esperada como fue documentada en su especificación. La especificación se asume correcta, las desviaciones son fallas. Los testers trabajan junto con los desarrolladores para priorizar y minimizar las pruebas de regresión.

- **Fase 4 - Medir el progreso de la prueba**: Las mediciones no ayudan a saber el progreso de la prueba. Es difícil saber cuando parar de probar, cuando el producto esta pronto para liberarse. Los testers necesitan una medida cuantitativa del número de defectos que quedan en el software y de la probabilidad de que cualquiera de estos bugs sean descubiertos en producción.

International Software Testing Qualifications Board (ISTQB) (ISTQB, 2013)**.** El programa del curso para la certificación del tester en el nivel inicial de International Software Testing Qualifications Board (ISTQB) define el proceso de prueba consistente de las siguientes actividades principales, que si bien son secuenciales, pueden ocurrir concurrentemente.

Planificación y control: La planificación de las pruebas es la actividad donde se definen los objetivos la prueba y las actividades de prueba para resolver esos objetivos. El control de las pruebas es la actividad de comparar el progreso real contra el planificado y reportar el estado, incluyendo desviaciones al plan.

Implica tomar las acciones necesarias para lograr los objetivos. La planificación de las pruebas tiene en cuenta la retroalimentación de las actividades de control y tiene las siguientes tareas principales:

- Determinar el alcance y los riesgos, identificando los objetivos de las pruebas
- Determinar el enfoque de prueba (técnicas, elementos a probar, cobertura, identificando e interconectando los equipos implicados en la prueba, testware)
- Determinar los recursos requeridos para las pruebas (personas, ambiente de prueba, PCs)
- Implementar la política de prueba y/o estrategia de prueba
- Agendar actividades de análisis y diseño de las pruebas
- Agendar la implementación, ejecución y evaluación de las pruebas
- Determinar los criterios de terminación

El control de las pruebas tiene las siguientes tareas principales:

- Medir y analizar los resultados
- Supervisar y documentar el progreso, cobertura de las pruebas y criterios de terminación
- Tomar acciones correctivas
- Tomar decisiones

Análisis y Diseño de las pruebas: Es la actividad donde los objetivos generales de las pruebas se transforman en condiciones de prueba tangibles y diseños de pruebas. El análisis y el diseño de las pruebas tienen las siguientes tareas principales:

- Revisar la base de las pruebas (tal como requerimientos, arquitectura, diseño, interfaces)
- Identificar las condiciones de prueba o requerimientos de prueba y los datos de prueba requeridos basados en el análisis de los elementos a probar, de la especificación, del comportamiento y de la estructura
- Diseñar las pruebas
- Evaluación de la testability de los requerimientos y el sistema
- Diseñar el ambiente de inicialización de la prueba e identificar cualquier infraestructura requerida y herramientas

Implementación y ejecución: Es la actividad donde las condiciones de prueba se transforman en casos de prueba y testware y se configura el ambiente. La implementación y ejecución de las pruebas tiene las siguientes tareas principales:

- Desarrollar y priorizar casos de prueba, crear datos de prueba, escribir procedimientos de prueba y, opcionalmente: preparación de arneses de prueba y escribir pruebas automatizadas
- Crear conjuntos de prueba (test suite) a partir de los casos de prueba para su ejecución eficiente
- Verificar que se haya instalado el ambiente de la prueba correctamente
- Ejecutar las pruebas manualmente o usando las herramientas de ejecución de pruebas, según la secuencia prevista

- Registrar el resultado de la ejecución de las pruebas y registrar la versión del software probada, las herramientas de prueba y testware
- Comparar resultados reales con resultados esperados
- Reportar discrepancias como incidentes y analizarlos para establecer su causa (por ejemplo: un defecto en el código, en los datos de prueba, en el documento de prueba, o un error en la manera que la prueba fue ejecutada)
- Repetir las actividades de la prueba como resultado de la acción tomada para cada discrepancia. Por ejemplo, reejecutar una prueba que falló previamente para confirmar un arreglo, ejecución de una prueba corregida y/o la ejecución de pruebas para asegurarse de que no fueron introducidos defectos en las áreas sin cambios o de que el arreglo del defecto no trajo otros defectos (prueba de regresión)

Evaluar el criterio de terminación y reportes: Es la actividad donde la ejecución de las pruebas es evaluada contra los objetivos definidos. Esto se debe hacer para cada nivel de la prueba. La evaluación del criterio de terminación tiene las siguientes tareas principales:

- Comparar los registros de las pruebas contra los criterios de terminación especificados en el plan de prueba
- Determinar si son necesarias más pruebas o si los criterios de terminación especificados deben ser cambiados
- Escribir un informe para el cliente

Actividades de cierre: Las actividades de cierre recogen los datos de las actividades terminadas para consolidar experiencia, testware, hechos y números. Las actividades de cierre incluyen las siguientes tareas principales:

- Comprobar que los entregables fueron entregados según lo planificado, el cierre de los reportes de los incidentes que siguen estando abiertos, y la documentación de la aceptación del sistema.
- Archivar el testware, el ambiente de la prueba y la infraestructura de prueba para su posterior reutilización

- Analizar lecciones aprendidas para futuras versiones y proyectos, y mejorar la madurez de las pruebas

Estos son las diferentes fases de las pruebas que serán consideradas para el desarrollo de la metodología de pruebas. También se considera el proceso unificado de desarrollo de Software (UP), por lo cual a continuación se indicar la información encontrada sobre el tema.

2.2.3. Modelos para la Mejora de las Pruebas

A continuación se describen cinco modelos de mejora para las pruebas, considerado por la revista española de Innovación, Calidad e Ingeniería del Software, en su volumen 4, No. 4 del 2008, como las más destacables son: TMM, TMMI, TPI, Tmap y TesPAI. (Sanz, Saldaña, Javier, & Gaitero, 2008)

2.2.3.1. **TMM**

Test Maturity Model (TMM) es un modelo que está centrado en el proceso de pruebas, definiendo un conjunto de niveles y objetivos de madurez. Fue creado por el Instituto de Tecnología de Illinois en 1996 y es complementario y compatible con CMM. Se basa en validación de la situación actual del proceso de prueba a través de reglas claras y objetivas, para mejorar los procesos de calidad de pruebas en una organización.

TMM propone cinco (5) niveles de madurez, en cada nivel se define un conjunto de áreas de proceso las cuales contienen las actividades relacionadas con la misma para contribuir a la mejora del proceso.

Lo que se espera obtener con este conjunto de niveles es: (i) definir una jerarquía hacia el proceso de pruebas maduro, (ii) objetivos de mejora en cada uno de los niveles para llevar a la madurez, (iii) sub-objetivos que definan el alcance, (iv) límites y necesidades de logro para alcanzar cada nivel y (v) actividades, tareas y responsabilidades por cada objetivo que lo respalda. A continuación la figura Nro. 3 en la cual se presenta la estructura implementada en

cada uno de los niveles y seguidamente en la figura Nro. 4 la estructura y objetivos de Madurez para TMM.

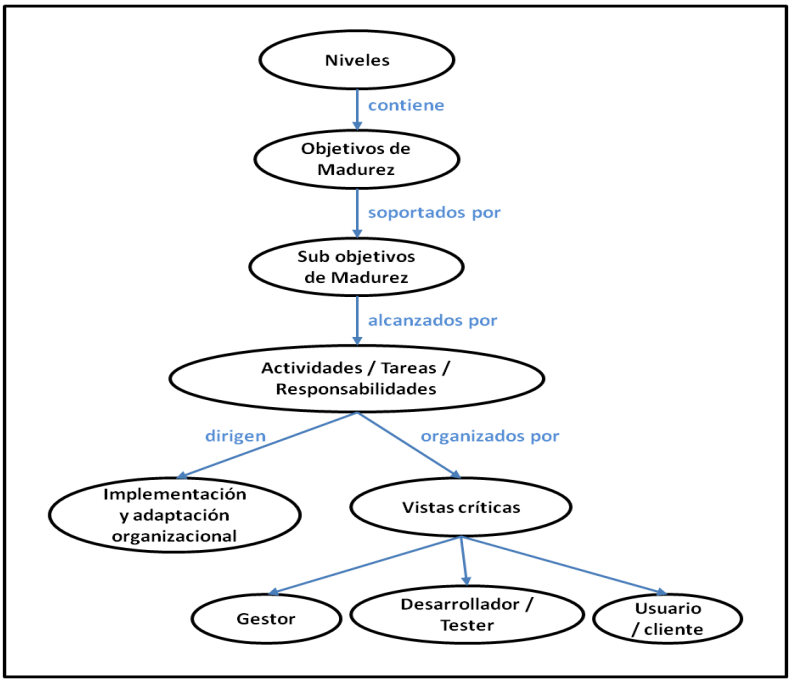


Figura 3 Estructura de cada nivel de madurez (Santana Cecilia, 2018)

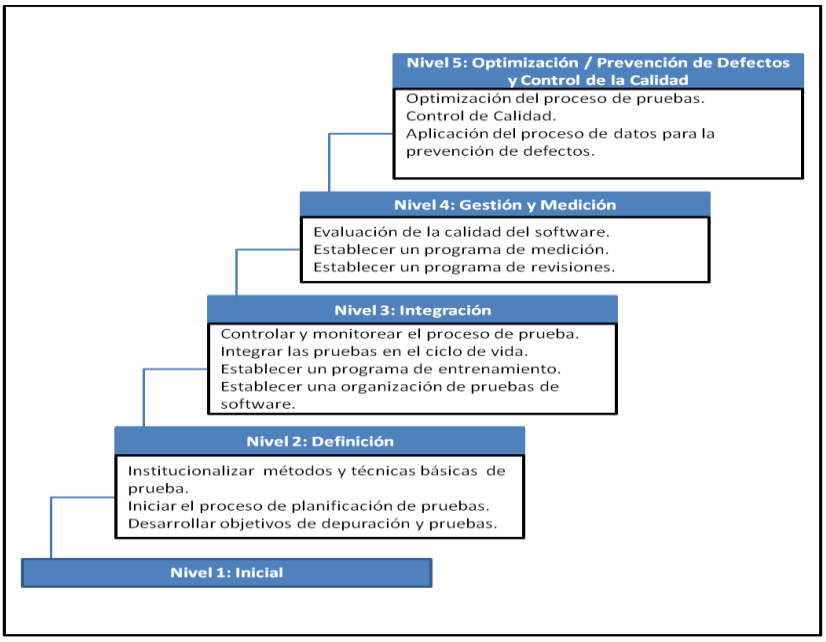


Figura 4. Estructura y Objetivos de Madurez para TMM (Santana Cecilia, 2018)

Según nos indica Ana Sanz (Sanz Esteban, 2011) cada uno de estos niveles tiene una serie de características. El nivel uno se caracteriza porque el proceso de pruebas es un caos, no se encuentra definido se consideran parte de la depuración, las pruebas se desarrollan ad hoc después de haber terminado toda la codificación.

En el nivel dos se separan las pruebas de la depuración, definiéndose una fase después de la codificación. El objetivo principal de este nivel es verificar que el software satisface los requisitos especificados. Sin embargo, surgirán problemas de calidad en este nivel, debido a que la planificación de las pruebas se realiza tarde. Además, los defectos se propagarán desde los requisitos y fases de diseño al código, ya que no existen programas de revisión.

En el nivel tres, las pruebas se encuentran integradas en todo el ciclo de vida del software, no son una fase que se realiza después de la codificación. A diferencia del nivel 2, la planificación de las pruebas comenzará con la fase de requisitos y continuará durante todo el ciclo de vida. Existe una organización de pruebas, un programa de formación y se reconoce a las pruebas como una actividad profesional. Se realizarán revisiones aunque no exista un programa formal de revisión y éstas resulten inconsistentes.

El nivel cuatro se caracteriza porque las pruebas son un proceso medido y cuantificado. Las revisiones de todas las fases del proceso de desarrollo son reconocidas como actividades de pruebas y control de calidad en este nivel. Se prueba el software atendiendo a los atributos de calidad, como fiabilidad, usabilidad y mantenimiento. Las deficiencias en este nivel se producen debido a la carencia en la prevención de defectos. Las actividades de prueba se expanden, incluyen revisiones, inspecciones y walkthroughs a través de las diferentes fases.

Finalmente, el nivel cinco se caracteriza porque el proceso de pruebas es repetible, definido, gestionado y medido, por lo que se pueden definir mecanismos que permitan una mejora

Como este modelo es complemento de CMM se ha quedado obsoleto por la desaparición de este último, por lo que en la actualidad apenas se usa. Sin embargo se ha desarrollado una

evolución de este modelo, como lo es TMMI, consistente con las nuevas versiones de CMMI for Development.

2.2.3.2. **TMMI**

TMMI (Test Maturity Model Integration) es un modelo detallado para mejorar el proceso de prueba, desarrollado por el TMMI Foundation, como se indicó antes es una evolución de TMM (Test Maturity Model). El objetivo de TMMi es soportar las actividades de prueba y la mejora del proceso de prueba tanto en las disciplinas de ingeniería de sistemas y de ingeniería de software. Al igual que CMMI, TMMi trata de proporcionar a las organizaciones la mejora de procesos, en este caso de procesos de prueba, para lo cual es sustancial contar con un mecanismo de evaluación que permita identificar las oportunidades de mejora. Para ello la TMMi Foundation ha elaborado el "TMMi Assessment Method Application Requirements", que contiene los requisitos para la evaluación de TMMi, de forma que se puede conocer el nivel de madurez de una organización en lo referente al proceso de pruebas. (Sanz Esteban, 2011)

TMMI presenta una estructura en la cual para cada nivel de madurez define unas áreas de proceso y por cada uno identifica los objetivos, tanto específicos como generales. La figura que se presenta a continuación evidencia lo indicado antes.

Figura 3. Áreas y Objetivos por Nivel de Madurez (Sanz Esteban, 2011)

Como su antecesor, TMMI define cinco niveles que van desde el llevar a cabo un proceso de pruebas ad-hoc y no gestionada hasta llegar a la optimización, pasando previamente por gestionado, definido y medio optimizado. Cuando se pasa de un nivel a otro se supone haber alcanzado una mejora y además sirve de punto de partida para el siguiente. A continuación se presentan los diferentes niveles de TMMI.

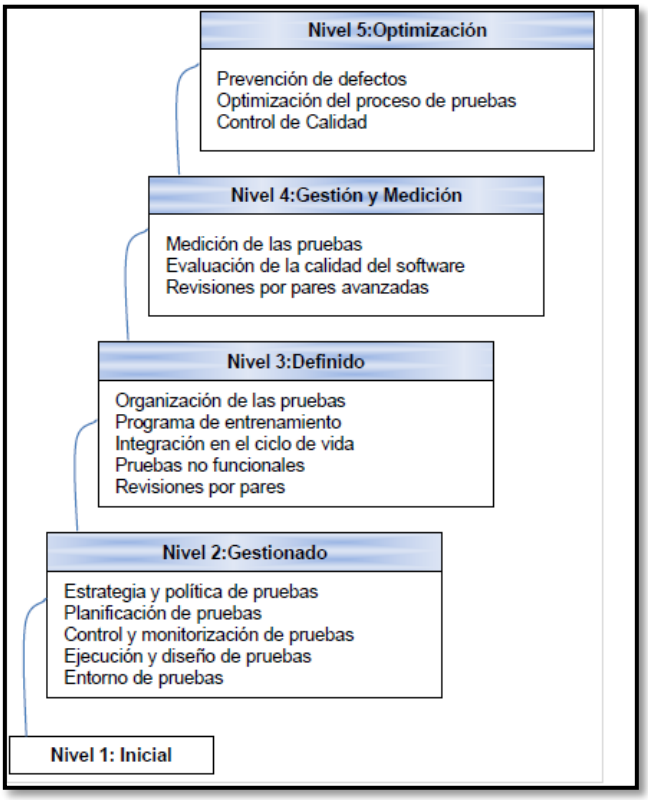


Figura 4. Niveles de TMMI (Sanz Esteban, 2011)

El nivel uno se caracteriza por presentar un caos, no hay planificación y se considera, frecuentemente como parte de la depuración. El éxito depende en gran medida de la competencia de las personas asignadas a las pruebas. (Sanz Esteban, 2011)

El segundo nivel tiene como principal objetivo verificar que el producto satisfaga los requisitos establecidos inicialmente. Está integrado por cinco área de proceso, que son: (1) política y estrategia de pruebas, (2) Planificación de pruebas, (3) Monitorización y Control

de pruebas, (4) Diseño y ejecución de pruebas y(5) Entorno de pruebas. (Sanz Esteban, 2011)

El tercer nivel se encuentra integrado dentro del ciclo de vida de desarrollo y cuenta con sus propios hitos. Las organizaciones que se encuentran en este nivel entienden la importancia de las revisiones y del control de calidad. Las áreas de proceso definidas para este nivel son: Organización de pruebas, Programa de formación en pruebas, Ciclo de vida e integración de pruebas, Pruebas no funcionales y revisiones por pares. (Sanz Esteban, 2011)

Al llegar al cuarto nivel, las pruebas se perciben como evaluación, por lo que las organizaciones ponen en práctica programas de mejorar del proceso de pruebas evaluación su calidad, la productividad y haciendo seguimiento de las mejoras. Este nivel consta de tres áreas de proceso: Medición de pruebas, Evaluación de la calidad del producto y revisión por pares avanzada. (Sanz Esteban, 2011)

El nivel 5 se caracteriza porque la organización es capaz de mejorar continuamente su proceso de pruebas en base a un proceso controlado estadísticamente. Los métodos y técnicas de pruebas son optimizados y existe un foco continuo en la mejora del proceso. Para alcanzar este objetivo se dispone de tres áreas de proceso: Prevención de defectos, Control de Calidad y Optimización del proceso de prueba. (Sanz Esteban, 2011)

Como se pudo observar cada una de los niveles maneja una serie de áreas de procesos, con la finalidad de que sean alcanzados los objetivos establecidos en cada uno de ello.

TMMI es utilizado en muchas ocasiones como modelo de referencia por las organizaciones para ser evaluadas con respecto a su proceso de prueba y determinar la madurez de los mismos.

TMMI es utilizado en muchas ocasiones como modelo de referencia por las organizaciones para ser evaluadas con respecto a su proceso de prueba y determinar la madurez de los mismos.

A continuación se mostrará la estructura de las áreas definidas para TMMI.

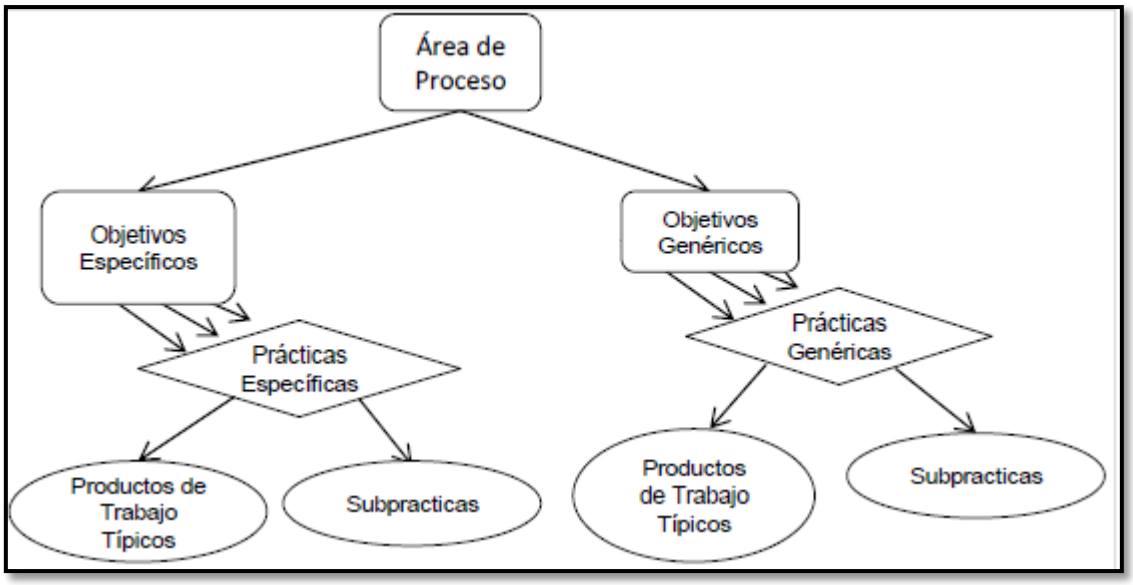


Figura 5. Estructuras de las áreas definidas para TMMI (Santana Cecilia, 2018)

Además, la aplicación de TMMi está fuertemente ligada a CMMI, requiere que las organizaciones sigan CMMI en su representación por etapas como modelo de referencia para la mejora del resto de procesos no relacionados con las pruebas. (Sanz Esteban, 2011)

2.2.3.3. **TPI (Test Process Improvement)**

TPI es un modelo de referencia dirigido al proceso de pruebas. Propone un conjunto de veinte (20) áreas claves con diferentes niveles de madurez. Define una Matriz de Madurez de Pruebas que establece las relaciones existentes entre las diferentes áreas y determinar la relevancia de cada uno de los niveles. Además establece un conjunto de puntos de Comprobación y Sugerencias de Mejoras. (Sanz, Saldaña, Javier, & Gaitero, 2008)

Las veinte áreas claves son: estrategia de pruebas, modelo del ciclo de vida, momento de implicación, estimación y planificación, técnicas de especificación de pruebas, técnicas de pruebas estáticas, métricas, automatización de pruebas, entorno de pruebas, entorno de trabajo, compromiso y motivación, funciones de prueba y entrenamiento, alcance de la

metodología, comunicación, informes, gestión de defectos, gestión de elementos de prueba, gestión del proceso de pruebas, evaluación y pruebas de bajo nivel. (Sanz Esteban, 2011)

Según Sogeti, creador de este mecanismo para la mejora del proceso de pruebas, TPI es un medio que puede proporcionar información valiosa sobre el estado actual del proceso de pruebas que permita la mejora del mismo. Se basa en el concepto de dar pasos pequeños y controlados a partir de prioridades. Debe considerarse como una herramienta para estructurar las acciones de mejora del proceso de pruebas y como un medio para la comunicación. (Sanz Esteban, 2011)

Cada una de las áreas clave puede tener un determinado nivel de capacidad (ascendiendo desde la A hasta la D). El ascenso de nivel implica una mejora en tiempo, dinero y/o calidad. Los requisitos que permiten pasar de un nivel a otro son definidos en forma de preguntas que deben ser respondidas positivamente para alcanzar cierto nivel, siendo necesario haber logrado el nivel anterior para pasar al siguiente; es decir no se podrá estar en un nivel B si no se cumplen los requisitos del nivel A. A este conjunto de preguntas se le denominará puntos de validación, ya que permiten determinar el nivel en el que se encuentra el proceso para cada área clave. Además, como no todas las áreas y niveles son igual de importantes y existen dependencias entre ellas, TPI define una matriz de madurez de pruebas que establece las diferentes relaciones (Sanz Esteban, 2011)

Lo que se busca con la matriz es poder representar los puntos fuertes y débiles en cada una de las áreas del proceso de prueba a fin que sean realizando los correctivos necesarios para mejorar esta actividad.

A diferencia de los modelos anteriores TPI no está relacionado con CMM o CMMI, sino que se basa en la metodología TMap por lo que es importante utilizar esta metodología para la gestión de pruebas.

A continuación se presenta la matriz de madurez de pruebas

Escala / Área Clave	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Estrategia de pruebas		A					B				C		D	
Modelo del ciclo de vida		A			B									
Momento de implicación			A				B				C		D	
Estimación y planificación				A							B			
Técnicas de especificación de pruebas		A		B										
Técnicas de pruebas estáticas				A			B							
Métricas						A			B			C		D
Automatización de pruebas				A				B			C			
Entorno de pruebas				A				B						C
Entorno de trabajo				A										
Compromiso y motivación		A				B						C		
Funciones de prueba y entrenamiento				A			B			C				
Alcance de la metodología					A						B			C
Comunicación			A		B							C		
Informes		A			B		C					D		
Gestión de defectos		A				B		C						
Gestión de elementos de prueba			A			B				C				D
Gestión del proceso de pruebas		A		B								C		
Evaluación							A			B				
Pruebas de bajo nivel					A		B		C					

Figura 6. Matriz de Madurez de TPI (Sanz Esteban, 2011)

2.2.3.4. Tmap - Test Management Approach

Tmap es un marco de trabajo que fue publicado por Sogeti en 1995 y luego actualizado como Tmap Next en el año de 1996. Está basado en una metodología para el desarrollo estructurada de pruebas, por lo que se pueden distinguir cuatro aspectos fundamentales: (1) Su enfoque está dirigido a negocio (Busines Driver Test Management), (2) describe un proceso de pruebas estructurado, (3) proporciona una caja de herramientas: técnicas, infraestructura y organización, para la realización de las pruebas, (4) es una metodología adaptativa. (Sanz, Saldaña, Javier, & Gaitero, 2008)

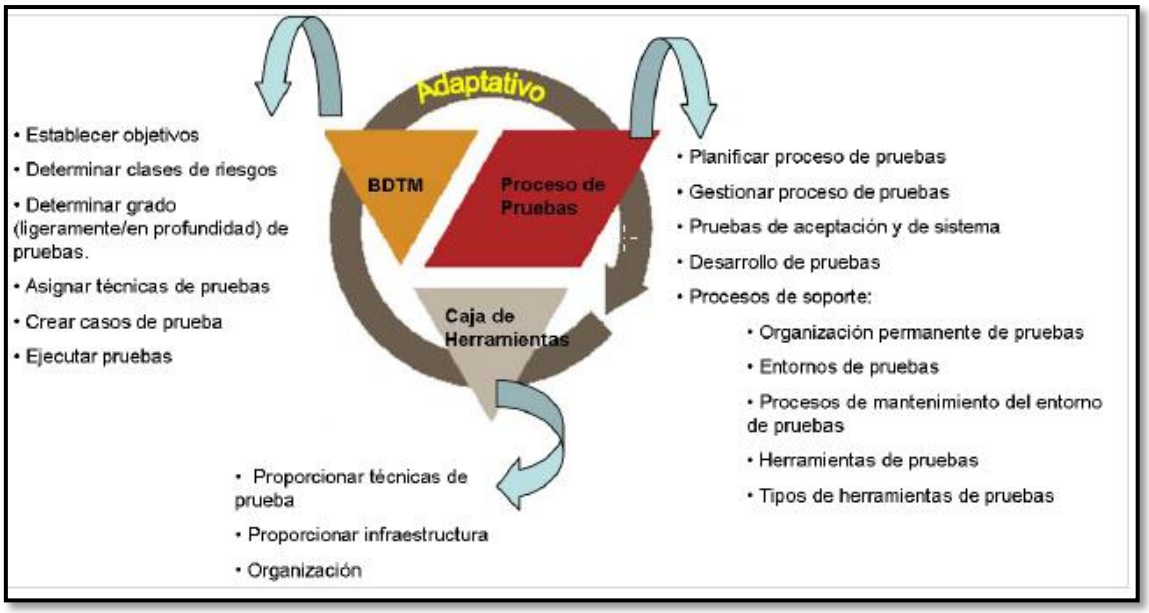


Figura 7. Elementos de Test Management Approach (Sanz, Saldaña, Javier,&Gaitero,2008)

Los autores señalan que es una metodología que te define el cómo: organizar, planificar, ejecutar y controlar las pruebas con la finalidad de obtener un producto de calidad. A continuación se presenta en detalle los aspectos señalados anteriormente (Sogeti, 2013):

1. **Business-driven Test Management.**

Como se menciona, Tmap se caracteriza porque su enfoque está dirigido al negocio, por lo que el cliente debe poder influenciar los cuatro (4) aspectos de control, como lo son: resultados, riesgos, tiempo y costes. Otra de las características es que el seguimiento y control del testing se comunica en el lenguaje del cliente en forma de objetivos, adicionalmente los test se basan en los riesgos del producto y los resultados del testing se hacen visibles (Sanz, Saldaña, Javier, & Gaitero, 2008).

Las propiedades específicas de BDTM son: el esfuerzo total está relacionado con los riesgos del sistema que va a ser probado, la estimación y planificación del proceso de pruebas está relacionada con la estrategia de pruebas definida, y hay que implicar al cliente en el proceso de pruebas. (Sanz Esteban, 2011).

Para lograr los aspectos indicados se cuentas con los siguientes pasos (Sogeti, 2013):

- Formular la misión y recolectar objetivos de las pruebas.
- Determinar las categorías de riesgo.
- Determinar cobertura alta/baja.
- Se realiza una estimación global para las pruebas y la planificación creadas.
- Asignar técnicas de pruebas.
- Proporcionar al cliente información sobre el proceso de test.

2. **Proceso de Test Estructurado)**.

El proceso de pruebas se encuentra dividido en cuatro partes que se mencionan a continuación:

- Plan de Pruebas Maestro. Se compone de 2 fases: planificación de la totalidad del proceso de test; y control de la totalidad del proceso de test. Su principal objetivo es detectar los defectos más importantes con el menor coste y tiempo posibles. (Sogeti, 2013)
- Pruebas de Aceptación y Sistema: Las pruebas de aceptación y sistema son consideradas como procesos autónomos que deben ser organizados; por lo que tienen su propio plan de pruebas, presupuesto y, frecuentemente, su entorno de pruebas propio. Se ejecutan en paralelo al proceso de desarrollo que debe comenzar mientras se crean las especificaciones funcionales. (Sanz Esteban, 2011). Tmap presenta un ciclo de vida que es aplicable a todos los niveles de prueba, y permite coordinar y sincronizar estos niveles. Transmite responsabilidades y tareas entre grupos involucrados, además de dividir el proceso de prueba en diferentes fases, actividades y productos. (Sogeti, 2013)
- Pruebas de Desarrollo. Son una parte integral del trabajo de desarrollos ejecutados por el desarrollador. Ellos no están organizadas como un proceso autónomo de un equipo independiente. A pesar de ello, una serie de diferentes actividades para el proceso de prueba de desarrollo, con su orden y dependencias mutuas, puede ser

identificado y descrito con la ayuda del modelo de ciclo de vida TMap. La elaboración detallada pueden variar según el proyecto u organización y depende, entre otras cosas, sobre el método de desarrollo utilizado y la disponibilidad de ciertas medidas de calidad. Una importante medida de calidad es el concepto de la calidad acordada. Con este fin, las expectativas del cliente en relación con la artesanía y la calidad del producto debe hacerse explícito en la planificación para establecer pruebas de desarrollo. Ejemplos de otras medidas de calidad son: desarrollo basado en pruebas, la programación en parejas, revisión de código, integración continua y el enfoque integrador de la aplicación. (Sogeti, 2013)

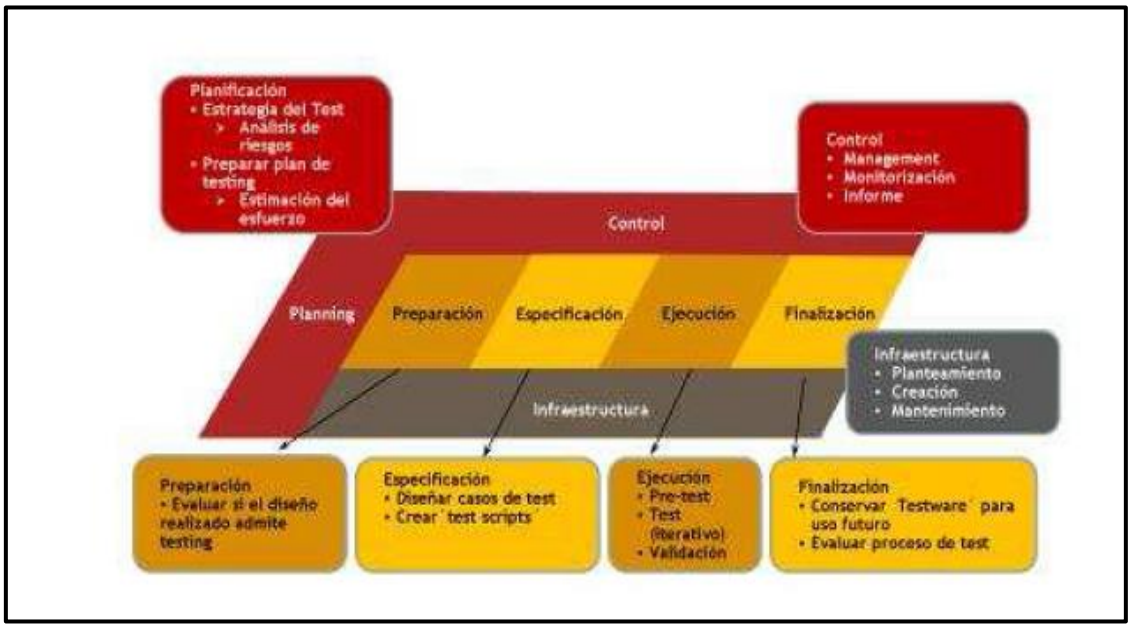


Figura 8. Ciclo de Vida de TMap (Sogeti, 2013)

- Procesos de soporte. Para realizar las pruebas se hace necesario un entorno de pruebas que soporte a las mismas, y que en algunos casos cumpla con un conjunto de requisitos genéricos que garanticen la ejecución fiable de las pruebas. Debe ser representativo, manejable y flexible, y también debe garantizar la continuidad de las pruebas. Además, para prevenir problemas en el entorno de pruebas es necesaria la implantación de procesos para gestionar la configuración y el mantenimiento de entornos. Estos procesos pueden ser: Gestión de configuración, Gestión del cambio,

Gestión de incidencias, Gestión de problemas, y Gestión de datos. (Sanz Esteban, 2011) Muchas veces resulta más eficiente centralizar los procesos de soporte de prueba que implementarlos a nivel de proyecto. El soporte de prueba puede ponerse en práctica de dos formas diferentes, Test Expertise Centre (TEC), o bien Test Factory (TF). (Sogeti, 2013)

3. **Proporcionar caja de herramientas.** TMap soporta la ejecución del proceso de pruebas estructurado mediante una caja de herramientas completa, centrada en: técnicas (¿cómo?), la infraestructura (¿dónde? ¿con qué?) y la organización (¿quién?). (Sanz Esteban, 2011)
   - Las técnicas incluyen: (Sogeti, 2013)
     - Estimación de Test
     - Gestión de Defectos
     - Creación de Métricas
     - Análisis de Riesgos del Producto
     - Diseño de Test
     - Evaluación del Producto
   - La infraestructura. Se definen los entornos de test: DTAP (desarrollo, test, aceptación y producción). Se utilizan herramientas de test que incrementan la productividad, mejoran la calidad del test, y ofrecen más opciones. Y, por último, se definen las condiciones del lugar de trabajo. (Sogeti, 2013)
   - La organización. Se establecen los roles y responsabilidades, las tareas y las autorizaciones. El tester debe reunir las siguientes características: ser buen comunicador, poseer capacidad de análisis, ser objetivo, poseer una actitud crítica positiva y ser creativo. (Sogeti, 2013)

4. **Adaptable**. La adaptabilidad de TMap® Next se resume en:

   - Respuesta a los cambios. En TMap Next, la respuesta a los cambios sucede en las primeras actividades del Master Test Plan. La obtención de información sobre el

entorno en el que se ejecuta el test y el establecimiento de posibles cambios desempeñan un papel importante. Si la estrategia de test no es aceptada por el cliente, el plan se adapta. (Sogeti, 2013)

- Re utilización de productos y procesos. TMap Next ofrece la posibilidad de utilizar productos y procesos rápidamente, gracias a sus técnicas de diseño, checklists, plantillas, etc. En la fase final, se definen las actividades para identificar la reutilización y preservación de los objetos de test (testware). (Sogeti, 2013)
- Aprendizaje desde la experiencia. Al ser un método, TMap Next ofrece el marco necesario para aprender y aplicar lo que ya se ha utilizado con anterioridad. La actividad de evaluación del proceso de tets forma parte del Modelo de Ciclo de Vida. Las métricas también son importantes para mejorar el proceso de test de forma continua. (Sogeti, 2013)
- Probar antes de usar. El mejor instrumento que TMap Next ofrece para probar antes de usar son los ensayos de test (intake). Otra forma de probar es customizar TMap Next para adaptarse a una situación específica. (Sogeti, 2013)

TMap propone el modo de conseguir un proceso de pruebas efectivo y eficiente, permitiendo que la organización cumpla los objetivos claves del negocio. Es efectivo porque intenta encontrar a tiempo los defectos importantes y tiene en cuenta los riesgos del producto y es eficiente, porque es un método aplicable universalmente, que se centra especialmente en la reutilización.

2.2.3.5. **TestPAI**

El TesPAI es un área del proceso de pruebas integrado con CMMI, el cual se encuentra en el nivel tres (3) con los procesos de ingeniería. TestPAI ha sido desarrollado para proporcionar el marco de trabajo necesario para que la mejora del proceso de prueba se desarrolle de forma paralela a la mejora de los otros procesos implementados en la organización. (Sanz, Saldaña, Javier, & Gaitero, 2008).

TestPAI incluye y define todas las prácticas relativas a las pruebas, presenta la misma estructura que CMMI y contempla tanto la representación continua como la representación

por etapas, lo que permite una sencilla y completa integración entre CMMI y TestPAI. (Sanz, Saldaña, Javier, & Gaitero, 2008)

TestPAI define cinco (5) objetivos específicos (SG) y sus correspondientes prácticas específicas (SP). A continuación una pequeña explicación de los cinco (5) objetivos del TestPAI.

- Primer Objetivo: Establecer Objetivos y Políticas. Aquí se definen los objetivos y políticas asociadas a las pruebas, estableciendo sus objetivos y políticas de pruebas que permitan alcanzar los objetivos definidos. (Sanz, Saldaña, Javier, & Gaitero, 2008)
- Segundo Objetivo: Establecer y Mantener el Plan de Pruebas. La planificación implica establecer y analizar los diferentes elementos que afectan a la gestión y desarrollo de las pruebas, tales como definir alcance, analizar riesgos, definir la estrategia o asignar recursos, entre otros. (Sanz, Saldaña, Javier, & Gaitero, 2008)
- Tercer Objetivo: Establecer información necesaria para una ejecución efectiva de las pruebas. Un proceso de pruebas bien definido implica determinar características que no se encuentran descritas en el plan de pruebas, como pueden ser las entradas, salidas o resultados esperados de las pruebas. (Sanz, Saldaña, Javier, & Gaitero, 2008)
- Cuarto Objetivo: Proporcionar la infraestructura adecuada para realizar las pruebas. Una ejecución de pruebas satisfactorias es un elemento importante en la mejora de la calidad del software, y preparar la infraestructura correctamente contribuye a que la ejecución se desarrolle de forma disciplinada. (Sanz, Saldaña, Javier, & Gaitero, 2008)
- Quinto Objetivo: Analizar, evaluar y comunicar los resultados obtenidos con el objeto de poder evaluar el proceso de pruebas y la calidad del producto, siendo a través de las prácticas: establecer el informe de pruebas, evaluar el proceso de pruebas y determinar las acciones correctivas. (Sanz, Saldaña, Javier, & Gaitero, 2008)

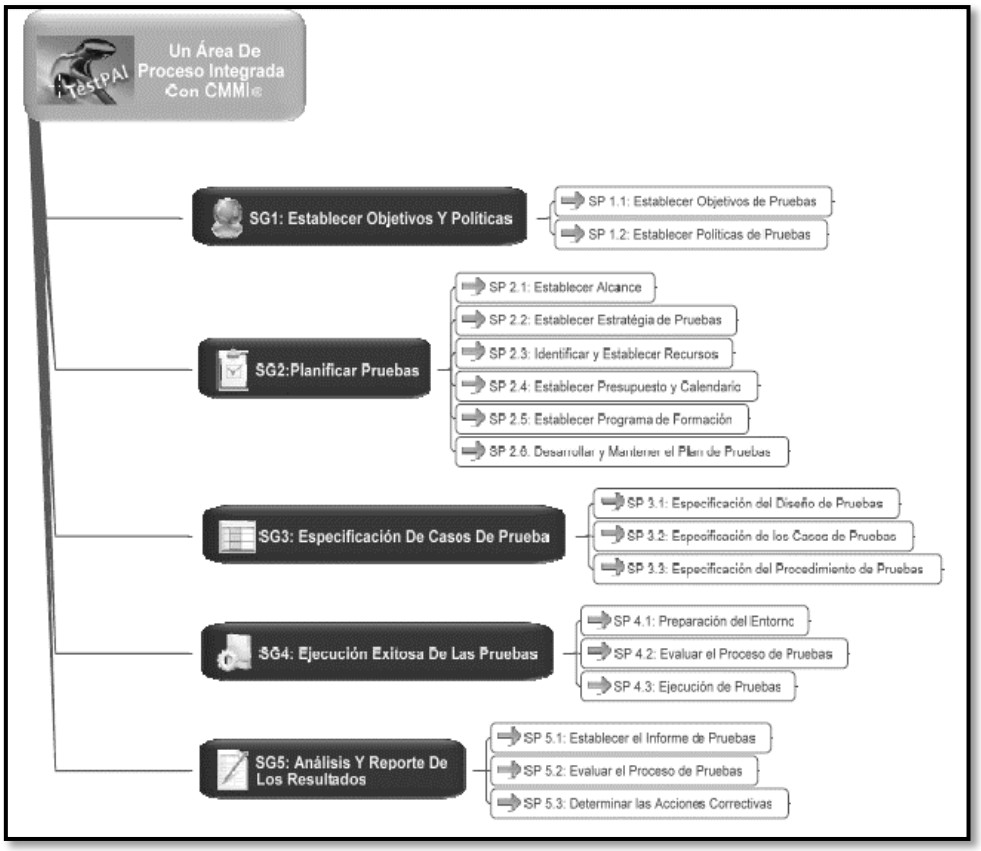


Figura 9. Objetivos y Prácticas Específicas de TestPAI (Sanz, Saldaña, Javier, &Gaitero, 2008)

2.2.4. Las Pruebas de Software basado en el Proceso de Desarrollo

2.2.4.1. El Proceso Unificado de Desarrollo de Software (UP)

El proceso unificado es un método iterativo de diseño de software que describe cómo desarrollar software de forma eficaz, utilizando técnicas probadas en la industria. UP es un marco de desarrollo de software que se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura, enfocado en el riesgo, y por ser iterativo e incremental (Jacobson, Booch, & Rumbaugh, 2000).

UP es una versión libre y abierta del modelo propuesto por Jacobson, Booch y Rumbaugh (Jacobson, Booch, & Rumbaugh, 2000). UP divide el trabajo de desarrollo de software en

cuatro fases: inicio, elaboración, construcción y transición, las cuales se describen a continuación.

- **Fase de Inicio.** En esta fase corresponde definir el negocio. Es la etapa donde se define la factibilidad del proyecto a realizar, se representa el modelo de negocio, visión y metas del proyecto, se identifican actores, conceptos de dominio y deseos de usuario
- **Fase de Elaboración.** En la fase de elaboración se obtiene la visión refinada del proyecto a realizar, la implementación iterativa del núcleo central de la aplicación, la resolución de los riesgos más altos, la identificación de nuevos requisitos y nuevos alcances, y estimaciones más ajustadas. A esta altura existe la posibilidad de detener el proyecto por complejidad técnica.
- **Fase de Construcción.** La fase de construcción es la implementación iterativa del resto de los requisitos de menor riesgo y elementos más sencillos. Es la evolución hasta convertirse en un producto listo, incluyendo todos los requisitos (100%), para entregarse al Cliente. Al final de esta fase el sistema contiene todos los casos de uso que el cliente y la dirección del proyecto han acordado. La mayoría de los casos de uso que no se desarrollaron en la fase anterior se desarrollan en iteraciones, en grupos de requisitos o casos de uso durante esta fase.
- **Fase de Transición.** Es el periodo donde el producto es completamente entregado al cliente para ser testeado y desplegado (instalado).

<u>Organización de las Disciplinas según el Proceso Unificado</u>

El proceso unificado (UP) incorpora una serie de disciplinas en cada una de las fases que ha definido, las cuales van participando a lo largo del ciclo de desarrollo del software.

A continuación se representa cada una de las disciplinas utilizadas en el proceso de desarrollo de software y su nivel de participación en cada una de las fases definidas de UP.

Las disciplinas identificadas son: modelado de negocios, requisitos, análisis, diseño, implementación y pruebas, como también se identifican las disciplinas de apoyo, tales como: configuración y manejo de proyectos. Todas estas disciplinas son representadas con su

correspondiente esfuerzo estimado para cada una de las fases definidas por UP. En este trabajo de investigación se dará atención a la disciplina de pruebas.

Las primeras iteraciones trabajarán riesgos críticos, casos de uso, asunto arquitectónico y orientado a la investigación, mientras que las últimas iteraciones serán implantación, prueba, evaluación y despliegue

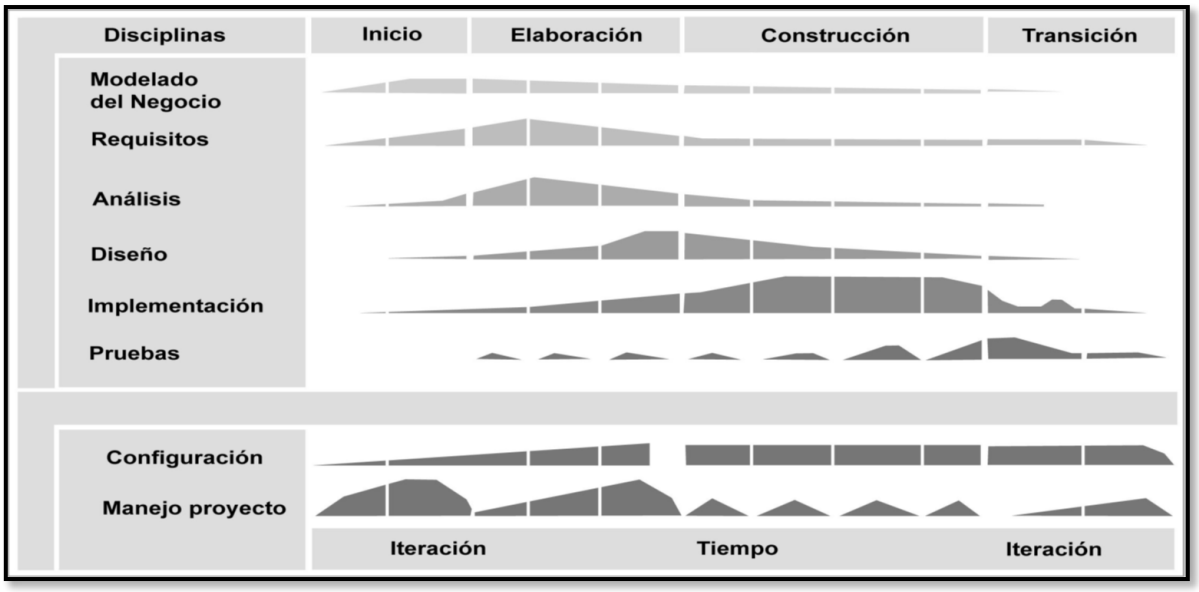


Figura 10. Fases y Disciplinas según el Proceso Unificado. (Jacobson, Booch, & Rumbaugh, 2000)

Según indica Torossi (Torossi, 2012) los flujos de trabajo desarrollan modelos, asociados a cada uno de las disciplinas, siendo estos:

Tabla 5 Modelos del Flujo de Trabajo UP (Torossi, 2012)

Disciplina	Modelos
Requisitos	Modelo de Caso de Uso
Análisis	Modelo de Análisis
Diseño	Modelo de Diseño – Modelo de Despliegue
Implementación	Modelo de Implementación
Prueba	Modelo de Prueba

Adicionalmente, (Torossi, 2012) indica que cada fase finaliza con un hito, el cual se determina por la disponibilidad de un conjunto de artefactos (modelos o documentos) que hayan sido desarrollados hasta alcanzar un estado predeterminado. Entre los objetivos que persiguen los hitos es el de controlar la dirección y progreso del trabajo, para lo cual los directores de los proyectos podrán tomar decisiones en cuanto a continuar a siguientes fases. Al final se obtiene un conjunto de datos, partiendo del seguimiento del tiempo y esfuerzo consumido en cada fase, con los cuales se podrán realizar estimaciones de esfuerzos en el futuro.

Durante la fase de inicio puede hacerse parte de la planificación inicial de las pruebas cuando se define el ámbito del sistema. Sin embargo, las pruebas se llevan a cabo sobre todo cuando una construcción es sometida a pruebas de integración y de sistemas. Esto quiere decir que la realización de pruebas se centra en las fases de elaboración, cuando se prueba la línea base ejecutable de la arquitectura, y de construcción, cuando el grueso del sistema está implementado. Durante la fase de transición el centro se desplaza hacia la corrección de defectos durante los primeros usos y a las pruebas de regresión. En la siguiente figura se presenta un resume por cada una de las disciplinas señaladas por UP y los artefactos esperados en cada fase.

COMPONENTES		FASES			
Disciplina	Artefacto	Inicio	Elaboración	Construcción	Transición
Modelado del negocio	Modelado del dominio		C		
Requerimientos	Modelo de casos de uso	C	R		
	Visión y análisis del negocio	C	R		
	Especificación complementaria	C	R		
	Glosario	C	R		
Diseño	Modelo de diseño		C	R	
	Documento de arquitectura		C		
	Modelo de datos		C	R	
Implementación	Modelo de implementación		C	R	R
Gestión del proyecto	Plan de desarrollo	C	R	R	R
Pruebas	Modelo de pruebas		C	R	
Entorno	Marco de desarrollo	C	R		

Dónde: **C = Comienzo de la construcción del artefacto. (Si un artefacto tiene sólo una "C" significa que se comienza y termina en la misma fase)**; **R = Refinamiento del artefacto (ampliación, corrección).**

Figura 11. Resumen Componentes y Fases de Proceso Unificado (UP) (Torossi, 2012)

Según Ivar Jacobson, Grady Booch y James Rumbaugh (Jacobson, Booch, & Rumbaugh, 2000) es natural mantener el modelo de pruebas a lo largo del ciclo de desarrollo del software, aunque este modelo de pruebas cambia constantemente debido a: (i) eliminación de casos de prueba obsoletos, (ii) el refinamiento de casos de prueba, para el caso de las pruebas de regresión y (iii) la creación de nuevos casos de uso para cada nueva construcción.

Los artefactos que se esperan encontrar en el modelo de pruebas son los siguientes:

- **Plan de prueba**: describe las estrategias, recursos y planificación de la prueba. La estrategia de prueba incluye la definición del tipo de prueba a realizar para cada iteración.
- **Modelo de pruebas**: describe cómo se pruebas los componentes ejecutables en el modelo de implementación con pruebas de integración y de sistemas. Este modelo de pruebas es una colección de casos de prueba, procedimientos de prueba y componentes de prueba.
- **Defecto**: Es una anomalía del sistema.
- **Evaluación de prueba**: Es una evaluación de los resultados de los esfuerzos de prueba tales como la cobertura del caso de prueba, la cobertura de código y el estado de los defectos.

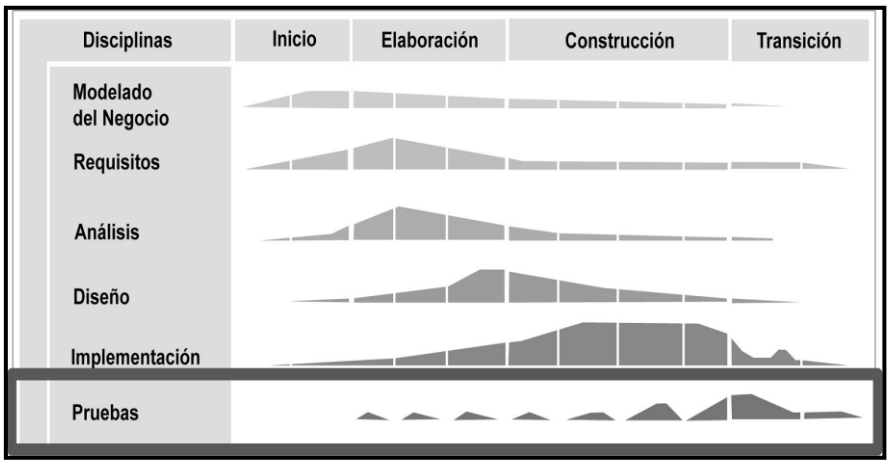


Figura 12. Disciplina de Prueba.(Jaconson, Booch, & Rumbaugh, 2000)

UP define claramente los roles, actividades e hitos importantes para la Disciplina de pruebas, que como puede observarse en la figura anterior, está presente en gran parte de la vida del proyecto y se interrelaciona con las demás disciplinas.

La Disciplina de pruebas define cuatro (4) roles especiales que son: Gerente de pruebas, analista de pruebas, diseñador de pruebas y el probador, quienes realizan una serie de actividades específicas de acuerdo al rol, entre las que destacan: acordar la misión, identificar los motivadores de pruebas, obtener compromiso para probar, estimar y mejorar el esfuerzo de pruebas, identificar requerimientos e ideas de pruebas, determinar los resultados de las pruebas, definir el enfoque de pruebas, definir las configuraciones de los ambientes de pruebas, identificar los mecanismos para probar, definir los elementos a probar, implementar las pruebas con la respectiva suite de pruebas, ejecutar las pruebas y analizar las fallas de pruebas.

Todas estas actividades se apoyan en los artefactos (entregables) donde intervienen los roles, siendo algunos de ellos los mencionados en el párrafo anterior. UP es ampliamente utilizado para dirigir el esfuerzo en pruebas, apoyándose en herramientas que permitan la automatización de pruebas y su gestión.

El proceso unificado reconoce la importancia de la comunicación con el cliente y los métodos encaminados a describir el punto de vista del cliente con respecto a un sistema. El UP enfatiza el importante papel de la arquitectura de software, y ayuda al arquitecto a enfocarse en las metas correctas como el entendimiento, el ajuste a los cambios futuros y la reutilización. Sugiere un flujo de proceso iterativo e incremental y que proporciona el sentido evolutivo esencial en el desarrollo del software moderno (Pressman, 2005).

Todos estos modelos y estándares lo que buscan es asegurar la calidad en los diferentes software, por lo que es importante en esta investigación establecer las pruebas en el e software que pueden ser aplicadas y ayudan Aseguramiento de la Calidad del Software.

2.3. Bases Legales y Éticas

Constitución Nacional de la República Bolivariana de Venezuela

- Las bases legales de esta investigación se encuentran representadas, en primer lugar, en la Constitución nacional de la república Bolivariana de Venezuela (1999), de los Derechos Humanos y Garantías y de los Deberes, Capítulo VI, de los derechos culturales y educativos.

No obstante, el Artículo 110 instituye que*:*

El Estado reconocerá el interés público de la ciencia, de la tecnología, el conocimiento, la innovación y sus aplicaciones y de los servicios de información, así como también garantizará el cumplimiento de los principios éticos y legales que se deberán regir las actividades en cuanto a la investigación científica, humanística y tecnológica, por ser estos, instrumentos fundamentales para el desarrollo económico, social y político de la nación, así como también para la seguridad y soberanía nacional. En tal sentido, el Estado destinará los recursos suficientes y creará el sistema nacional de ciencia y tecnología. Para el sector privado le corresponderá aportar los recursos para las mismas.

Ley Especial Contra Los Delitos Informáticos. República Bolivariana de Venezuela (2001)

- En cuanto a la ley Especial Contra Los Delitos Informáticos es otro sustento legal importante y de donde se selecciona, la protección integral de los sistemas que utilicen tecnologías de información, así como la prevención y sanción de los delitos cometidos contra tales sistemas o cualesquiera de sus componentes, o de los cometidos mediante el uso de dichas tecnologías. Dispone de 33 artículos que van desde varios conceptos básicos tecnológicos y de informática, así como sanciones, responsabilidad de las personas jurídicas, contra los sistemas que utilizan tecnologías de información, acceso indebido, sabotaje o daños a sistemas, espionaje informático, hurto, fraude, entre otros.

La presente investigación, en el marco ético sobre la propiedad intelectual y los derechos de los autores que se citan, se le atribuyen las ideas propias creadas por un individuo y

reconociendo las contribuciones a la ciencia que han sido realizados por ellos. Cuando se indiquen conclusiones, métodos, herramientas referentes a pruebas se identifica el autor de este aporte, ya sea individuo, organismo o empresa.

- La información mostrada en esta investigación sobre Mercantil Seguros es sólo con fines académicos y no se hace referencia lo largo del proyecto de ningún dato financiero, sólo serán usados referencias para comparación con otras empresas aseguradoras.

CAPITULO III

MARCO METODOLOGICO

En este capítulo se describe la estrategia metodológica; el conjunto de pasos a realizados para la consecución del proyecto, es el cómo se hizo y por qué se realizó el estudio para responder la problemática planteada. Adicionalmente se mencionan las razones por las cuales se seleccionó la metodología.

3.1. Nivel y Diseño de la Investigación

La metodología del proyecto incluye el tipo de investigación y el diseño, que se traducen en la estrategia que adopta el investigador para responder al problema planteado.

3.1.1. Nivel de Investigación

El nivel de la investigación, según Arias (Arias, 2006) se refiere al grado de profundidad con que se aborda un objeto o fenómeno. Aquí se indicará si se trata de una investigación exploratoria, descriptiva o explicativa.

Según el manual de la UPEL (UPEL, 2006), el proyecto factible consiste en la investigación, elaboración y desarrollo de una propuesta de un modelo operativo viable a solucionar problemas, requerimientos o necesidades de organizaciones o grupos sociales; puede referirse a la formulación de políticas, programas, tecnologías, métodos o procesos. El proyecto debe tener apoyo en una investigación de tipo documental, de campo o un diseño que incluya ambas modalidades.

En función del objetivo general del presente trabajo de investigación, el cual es diseñar una estrategia metodológica de pruebas en el ciclo de vida del software en la Gerencia de

Tecnología y Sistemas de Mercantil Seguros, bajo las mejores prácticas y estándares internacionales, su nivel de investigación es de un proyecto factible.

El proyecto factible comprende las siguientes etapas generales: diagnóstico, planteamiento y fundamentación teórica de la propuesta; procedimiento metodológico, actividades y recursos necesarios para su ejecución; análisis y conclusiones sobre la viabilidad y realización del Proyecto; y en caso de su desarrollo, la ejecución de la propuesta y la evaluación tanto del proceso como de sus resultados (UPEL, 2006).

Según Balestrini (Ballestrini, 2002) este tipo de estudios prospectivos en el caso de las Ciencias Sociales, sustentados en un modelo operativo, de una unidad de acción, están orientados a proporcionar respuestas o soluciones a problemas planteados en una determinada realidad: organizacional, social, económica, educativa, etc. En este sentido, la delimitación de la propuesta final, pasa inicialmente por la realización de un diagnóstico de la situación existente y la determinación de las necesidades del hecho estudiado, para formular el modelo operativo en función de las demandas de la realidad abordada.

Por lo tanto, para lograr la ejecución de este proyecto se requiere emplear la etapa de diagnóstico, para definir la situación actual que se presenta la actividad de pruebas, por una parte y la fundamentación teóricamente, por la otra parte, con lo cual se potencia el desarrollo tecnológico.

3.1.2. Diseño de la Investigación

El diseño de investigación es la estrategia general que adopta el investigador para responder al problema planteado. En atención al diseño, la investigación se clasifica en: documental, de campo y experimental (Arias, 2006).

Para el UPEL (UPEL, 2006) se entiende por Investigación de Campo, el análisis sistemático de problema en la realidad, con el propósito bien sea de describirlos, interpretarlos, entender su naturaleza y factores constituyentes, explicar sus causas y efectos, o predecir su ocurrencia, haciendo uso de métodos característicos de cualquiera de los paradigmas o enfoques de investigación conocidos o en desarrollo. Los datos de interés son recogidos en forma directa

de la realidad; en este sentido se trata de investigaciones a partir de datos originales o primarios.

Según Arias (Arias, 2006) la investigación de campo es aquella que consiste en la recolección de datos directamente de los sujetos investigados, o de la realidad donde ocurren los hechos (datos primarios), sin manipular o controlar variable alguna, es decir, el investigador obtiene la información pero no altera las condiciones existentes. De allí su carácter de investigación no experimental.

Se entiende por Investigación Documental, el estudio de problemas con el propósito de ampliar y profundizar el conocimiento de su naturaleza, con apoyo, principalmente, en trabajos previos, información y datos divulgados por medios impresos, audiovisuales o electrónicos. La originalidad del estudio se refleja en el enfoque, criterios, conceptualizaciones, reflexiones, conclusiones, recomendaciones y, en general, en el pensamiento del autor (UPEL, 2006).

Para Arias (Arias, 2006) la investigación documental es un proceso basado en la búsqueda, recuperación, análisis, crítica e interpretación de datos secundarios, es decir, los obtenidos y registrados por otros investigadores en fuentes documentales impresas, audiovisuales o electrónicas. Como en toda investigación, el propósito de este diseño es el aporte de nuevos conocimientos.

Según lo planteado anteriormente por los Autores y en función a los objetivos de la presente investigación, se debe diagnosticar una problemática existente, para lo cual es necesario desarrollar una investigación de campo, es decir, tomar los datos en el lugar donde ocurren los hechos. Pero además, se debe diseñar una solución donde los problemas encontrados sean resueltos, a partir de la investigación y la adaptación de un conjunto de conocimientos disponibles, los cuales se encuentran en los documentos y bibliografía existente, lo que implica el desarrollo de una investigación de tipo documental. Por lo cual la presente investigación se fundamenta en un diseño mixto.

3.2. Población y Muestra

Para Palella (Palella, 2006) la población de una investigación es el conjunto de unidades de las que se desea obtener la información y sobre las que se va a generar conclusiones, donde las unidades pueden ser personas, familias, grupos sociales, entre otros. Existen dos alternativas para el estudio de la población; el primero consiste en tomar la totalidad de la misma, el segundo en tomar una parte representativa del conjunto de características de dicha población, esta última se denomina muestra.

Indica Palella (Palella, 2006) que en el muestreo de tipo no probabilístico intencional el investigador establece previamente los criterios para seleccionar las unidades de análisis, por lo cual esta será la utilizada para el presente trabajo de investigación.

Dicha investigación se realizará en la Gerencia de Tecnología y Sistemas de Mercantil Seguros, en la cual se cuenta con un Departamento de Aseguramiento de la Calidad, con un jefe y siete (7) analistas de sistemas, que son las personas responsables de la actividad de pruebas de la Gerencia. Adicionalmente se cuenta con líderes de proyectos, especialistas de sistemas, especialistas de proyectos, quienes en algunas ocasiones realizan la actividad de prueba, por lo que también se considerarán como parte de la muestra de esta investigación.

3.3. Técnicas e Instrumentos de Recolección de Datos

Para desarrollar cualquier tipo de investigación es necesario precisar qué técnicas se van a utilizar y mediante qué medios se van a aplicar dichas técnicas para ejecutar con mayor exactitud la investigación que se realiza. Según Palella (Palella, 2006) las técnicas de recolección de información son las distintas formas o maneras de obtener la información, para el acopiado de datos se utiliza técnicas de observación, entrevistas, prueba, entre otras.

Arias (Arias, 2006) señala que las técnicas de recolección de datos son las distintas formas o maneras de obtener la información. Y es mediante una adecuada construcción de los instrumentos de recolección, que la investigación alcanza el equilibrio necesario entre teoría y hechos. También señala Arias que los instrumentos son medios materiales que se emplean para recolectar y almacenar información.

La UPEL (UPEL, 2006) señala que se entiende por Investigación Documental, el estudio de problemas con el propósito de ampliar y profundizar el conocimiento de su naturaleza, con apoyo, principalmente, en trabajos previos, información y datos divulgados por medios impresos, audiovisuales o electrónicos. Apoyándose en este concepto, se utilizarán las Técnicas Documentales para desarrollar el marco teórico del presente trabajo de investigación, en relación a la revisión y análisis de las fuentes documentales, archivos, documentos, bibliografía, internet, entre otros.

La entrevista, según Palella (Palella, 2006) es una técnica que permite obtener datos mediante un diálogo que se realizan entre dos personas cara a cara: El investigador y el entrevistado, la intención es obtener información que posee este último. Tiene como desventaja que la persona entrevistada dará su opinión subjetiva de aquello que se le pregunte. Por tal motivo es recomendable estructurar previamente la entrevista. Esta técnica es recomendable para poblaciones pequeñas.

En esta investigación se realizará la técnica de Entrevista para conocer la situación actual referente a la actividad de pruebas y así determinar fortalezas, oportunidades de mejora, debilidades y amenazas que deben ser corregidas.

El Benchmarking es una técnica utilizada para comparar dos o más productos o servicios, inicialmente se utilizaba para medir el rendimiento de software o componentes de equipos de computación. Fuera del ámbito informático, se ubica dentro de una organización como administración estratégica, en la cual se define como un proceso sistemático y continuo para evaluar productos, servicios y procesos de trabajo de las organizaciones que son reconocidas como representantes de las mejores prácticas, con el propósito de realizar mejoras organizacionales.

Una definición formal del término: “es un proceso de evaluación continuo y sistemático; un proceso mediante el cual se analizan y comparan permanentemente los procesos empresariales de una organización frente a los procesos de compañías líderes en cualquier parte del mundo, a fin de obtener información que pueda ayudar a mejorar su labor.” (Suarez, Arias, & Fernandez, 1996)

Será aplicada en el trabajo de investigación, al momento de comparar procesos de prueba y encontrar ventajas, desventajas, similitudes y diferencias entre ellas, que faciliten la selección de un proceso o bien un compendio de las mejores prácticas.

3.4. Las variables, sus dimensiones e indicadores

En este punto de la investigación es necesario definir las variables de investigación, establecer sus dimensiones, indicadores, así como su operacionalización.

Según Arias (Arias, 2006) la variable es una característica o cualidad; magnitud o cantidad, que puede sufrir cambios, y que es objeto de análisis, medición, manipulación o control en una investigación.

También Arias (Arias, 2006) indica que según su naturaleza las variables pueden ser cuantitativas y cualitativas. Las cuantitativas son aquellos que se expresan en valores o datos numéricos mientras que las cualitativas son las que se expresan de forma verbal, es decir mediante palabras.

Según Sabino (Sabino, 2007), una variable es cualquier característica o cualidad de la realidad que puede asumir diferentes valores, por lo que pueden ser cualitativas o cuantitativas, tomar valores continuos (infinitos valores) o discretos (valores enteros). Según este autor, existen al menos tres tipos de variables, dependiendo de sus relaciones: a) las variables independientes, las cuales no se encuentran condicionadas por ninguna otra variable; b) las variables dependientes, condicionadas por las variables independientes y c) las variables intervinientes que actúa entre las anteriormente mencionadas.

Cuando una variable es compleja deben ser estudiadas en sus partes constitutivas o dimensiones, por lo cual Arias (Arias, 2006) indica que ésta es un elemento integrante de una variable compleja, que resulta de su análisis o descomposición.

Una vez que han sido establecidas las dimensiones de la variable, hay que establecer los elementos o evidencias que muestren como se manifiesta o comporta dicha variable o dimensión. Estos elementos son los indicadores.

Un indicador es un indicio, señal o unidad de medida que permite estudiar o cuantificar una variable o sus dimensiones (Arias, 2006). En el caso de una variable compleja, cada dimensión que la integra puede tener uno o varios indicadores, de los cuales se derivan las preguntas o ítems del instrumento de recolección de datos.

3.4.1. Operacionalización de variables

Aun cuando la palabra "operacionalización" no aparece en la lengua hispana, se utiliza en investigación científica para designar al proceso mediante el cual se transforma la variable de conceptos abstractos a términos concretos, observables y medibles, es decir, dimensiones e indicadores, Arias (Arias, 2006).

La definición operacional consiste según Palella (Palella, 2006), en identificar las variables a estudiar y establecer el significado que el investigador les atribuye dentro de la investigación. Esta definición conceptual o nominal se limita a explicar el significado de la variable, utilizando para ello un lenguaje sencillo. De esta manera se identifican los elementos y datos empíricos que expresan y especifican el fenómeno estudiado, indican qué hacer para que el investigador pueda realizar sus observaciones. Así la definición operacional asigna significado a la variable, describiéndola en términos observables y comprobables para poder identificarla, a través de la caracterización o tipificación proporcionados por sus indicadores.

La operacionalización de una variable consta de tres etapas básicas:

a) Definición nominal, consiste en establecer el significado de la variable, con base en la teoría y mediante el uso de otros términos;

b) Definición real de la variable: significa descomponer la variable, para luego identificar y determinar las dimensiones relevantes para el estudio; y

c) Definición operacional de la variable: establece los indicadores para cada dimensión, así como los instrumentos y procedimientos de medición (Arias, 2006). A continuación una serie de tablas con la información descrita anteriormente para cada uno de los objetivos específicos.

Tabla 6 Operacionalización de Variables - Objetivo Específico Nro 1 (Santana Cecilia, 2018)

Objetivo Específico	Variables	Definición Nominal	Dimensiones
1. Elaborar un modelo conceptual que soporte el trabajo de investigación, partiendo de normas, mejores prácticas y estándares internacionales en la disciplina de pruebas.	Modelo conceptual	Permite determinar los conceptos y relaciones que se obtendrán en esta investigación, así como los aportes del investigador y el análisis del conjunto de herramientas de prueba.	• Concepto de pruebas. • Conceptos de sistemas de información. • Relaciones entre los conceptos.
Indicadores	**Fuente**	**Instrumento**	**Pregunta base**
• Pruebas de unidad / componentes. • Pruebas de integración / cadena. • Pruebas de sistema. • Pruebas de aceptación / piloto. • Pruebas de mantenimiento. • Pruebas Operacionales. • Validación. • Verificación. • Proceso Unificado del desarrollo de software.	Referencias bibliográficas	Investigación documental	¿Qué es una prueba? ¿Cuáles son las etapas de desarrollo de sistemas y su relación con las pruebas?

Tabla 7 Operacionalización de variables - Objetivo Específico Nro. 2. (Santana Cecilia, 2018)

Objetivo Específico	Variables	Definición Nominal	Dimensiones
2. Describir la situación actual de pruebas en la Gerencia de Tecnología y Sistemas de Mercantil Seguros.	Fortalezas y oportunidades de mejora del proceso actual de prueba	Permite determinar las fortalezas y oportunidades de mejora que pueden ser aprovechadas para la propuesta, así como las debilidades y amenazas que deben ser subsanadas.	• Procesos. • Personal. • Infraestructura.
Indicadores	**Fuente**	**Instrumento**	**Pregunta base**
• Procedimientos documentados • Años de experiencia del personal de prueba. • Plantillas / formatos que existen. • Herramientas técnicas que se utilizan para la actividad de pruebas.	Juicio de experto	• Entrevista • Cuestionario	¿Cómo es el actual proceso de pruebas? ¿Cuáles son sus debilidades y oportunidades de mejoras en el proceso? ¿Cuáles son los entregables de pruebas en cada una de las fases del ciclo de desarrollo de software?

Tabla 8 Operacionalización de Variables - Objetivo Específico Nro. 3 (Santana Cecilia, 2018)

Objetivo Específico	Variables	Definición Nominal	Dimensiones
3. Identificar las técnicas y herramientas relativas a la disciplina de pruebas que pueden ser aplicados al proceso de estudio.	Técnicas y herramientas relativas a la disciplina de	Identificación los estándares, mejores prácticas, herramientas y técnicas, que servirán de	• Normas relacionadas con la disciplina de pruebas. • Estándares de prueba.

	pruebas.	base para el desarrollo de la propuesta.	• Mejores prácticas de pruebas
Indicadores	**Fuente**	**Instrumento**	**Pregunta base**
• IEEE 1028 -1997 • IEEE 829 – 1998 • IEEE 12207 / ISO / IEC 12207 • ISO / IEC 9129-1: 2001 • ISTQB	Referencias bibliográficas	• Modelo conceptual • Resultado de entrevistas • Investigación documental	¿Qué nos indica el estándar IEEE 1028 - 1997? ¿Según el ISTQB cuáles son las diferentes pruebas que deben ser aplicadas en el ciclo de desarrollo del software?

Tabla 9 Operacionalización de Variable - Objetivo Específico Nro. 4 (Santana Cecilia, 2018)

Objetivo Específico	Variables	Definición Nominal	Dimensiones
4. Elaborar una Guía Metodológica de pruebas de software, provenientes de los estándares y mejores prácticas existentes, previamente seleccionadas.	Guía metodológica	Es la organización y sistematización de la propuesta de la actividad de pruebas, de manera que sea efectiva, eficiente y factible en cuanto a su aplicación	• Aspectos generales • Procesos
Indicadores	**Fuente**	**Instrumento**	**Pregunta base**
• Descripción general • Objetivos • Alcance • Entradas de los procesos • Actividades, técnicas y herramientas de cada proceso • Salida de los procesos	Referencias bibliográficas. Juicio de experto	• Técnicas y herramientas de pruebas. • Investigación documental	¿Cuál es la descripción general de la guía metodológica? ¿Cuáles son los objetivos que persigue la guía metodológica? ¿Cuáles son los procesos que serán considerados?

3.5. Técnicas de Procesamiento y Análisis de Datos

Respecto a procesamiento y análisis de datos, Méndez (2009) precisa; que se trata del ordenamiento de la información, la cual debe ser procesada por ítem, agrupada por variables, de manera que permita su presentación en tablas. Esto requiere la realización de cálculos, gráficos, cuadros y tablas. Posterior a este ordenamiento se procesan los datos, de cuyos resultados se nutrirá el posterior análisis e interpretación.

Por otra parte Sabino (2007), profundiza en la forma de hacer dicho procesamiento y análisis. Señala que como primer paso debe separarse los datos numéricos de los cualitativos, ya que el tratamiento de la información difiere uno del otro.

Los datos que se presentan en forma no verbal pueden tener dos destinos distintos, el primero es convertirlo en datos numéricos, en segundo lugar, en caso de no poder hacer dicha conversión, se tratará como información no cuantificada o cualitativa. Una vez adoptado el criterio anterior frente a los datos, se procede a revisarlos en cuanto a su coherencia, en la búsqueda de incongruencias, omisiones o errores, corrigiéndolos si fuera posible.

Los datos numéricos se agrupan en intervalos para su tabulación; los datos verbales que se puedan presentar como datos numéricos, requieren de un proceso de codificación, para luego ser tratados como datos numéricos. Existen datos de contenido verbal que no soportan esta codificación, por lo que es necesario practicar procesos de análisis y síntesis sobre ellos, para este fin se hace necesario ordenar estos datos, preferiblemente en forma esquemática, ordenado por temas, capítulos puntos, para efectuar sobre ellos un proceso de inspección sistemática y profunda.

Para la codificación, se agrupan numéricamente los datos de tipo verbal, a partir de cúmulos de información que tengan una mínima homogeneidad, dividiendo las respuestas en subgrupos, creando de esta manera categorías para cada pregunta formulada, a cada categoría se le asigna un código particular que sirve para agrupar todas las respuestas con características similares, de esta forma según Hurtado (2010-b), el investigador, al no tener unas respuestas preestablecidas, las categorías surgen de la misma información recolectada.

Una vez definidos los códigos entonces se hace posible la tabulación, según Sabino (2007), de esta forma los datos se muestran agrupados y contabilizados por categoría o códigos previamente establecidos. El resultado de la codificación aún no puede presentarse como resultado de la investigación, para ello se hace necesario efectuar procesos de análisis, comparación y síntesis, según sea necesario.

El análisis implica descomponer el todo en sus partes constitutivas, continúa Sabino (2007), la síntesis por el contrario consiste en estudiar las partes y a partir de ellas construir la totalidad

inicial. En el análisis se toma cada uno de los datos o conjunto de datos, se hace necesario interrogarse sobre su significado, explorar y examinar con métodos conocidos, existiendo dos formas de hacerlo:

- Análisis cuantitativo: Este tipo de análisis se efectúa sobre toda la información numérica resultado de la investigación, en este tipo de análisis se correlacionan las variables numéricas, detectando las influencias que existan.
- Análisis cualitativo: Se aplica a la información de tipo verbal, una vez clasificada se comparan, sacando de ellas pequeñas conclusiones específicas o notas, las cuales posteriormente servirán para hacer inferencias sobre las mismas y de esta forma llegar a conclusiones más amplias.

En cuanto a la síntesis, Sabino (2007), señala que es una conclusión final, la cual engloba la totalidad de las apreciaciones que se han venido haciendo a lo largo del trabajo. La síntesis se basa en los análisis previos donde se encuentran registrados los hallazgos parciales.

En la presente investigación y según el tipo de instrumento de recolección de datos, se aplicaron las siguientes técnicas de procesamiento de datos:

- En el caso de las entrevistas, se categorizarán los datos antes de su recolección, se codificarán para agruparlos y realizar sobre ellos un análisis de tipo numérico, lo cual generará tablas y gráficos representativos de cada categoría. Esta información será analizada para descubrir las interrelaciones existentes entre las variables, lo que permitirá elaborar una síntesis o conclusión de los datos recopilados.
- En cuando a la investigación documental se basará en el análisis de los estándares IEEEE 829-1998, IEEE 1028-1997, ISO/IEC 9126-2001, IEEE 12207 / IEC 12207-1996, ISTQ, y UP.

3.6. Fases de la Investigación

Una estrategia es un conjunto de acciones que se llevan a cabo para lograr un determinado fin. La estrategia que da sustento a la presente investigación se encuentra definida teóricamente en el Capítulo II (marco teórico), a nivel práctico, se identifican las fases o etapas del desarrollo

de software, en las cuales se indicarán las entradas y salidas de cada una de ellas, con lo cual se favorecerá el logro de los objetivos planteados. En cada una de las etapas se diagnosticará, evaluará y se consolidará para luego preparar una guía metodológica del proceso de pruebas, que facilite este proceso en el desarrollo de software por parte de Mercantil Seguros.

A continuación se presenta la estructura desagregada de trabajo (EDT) que fue desarrollada para llevar a cabo la presente investigación:

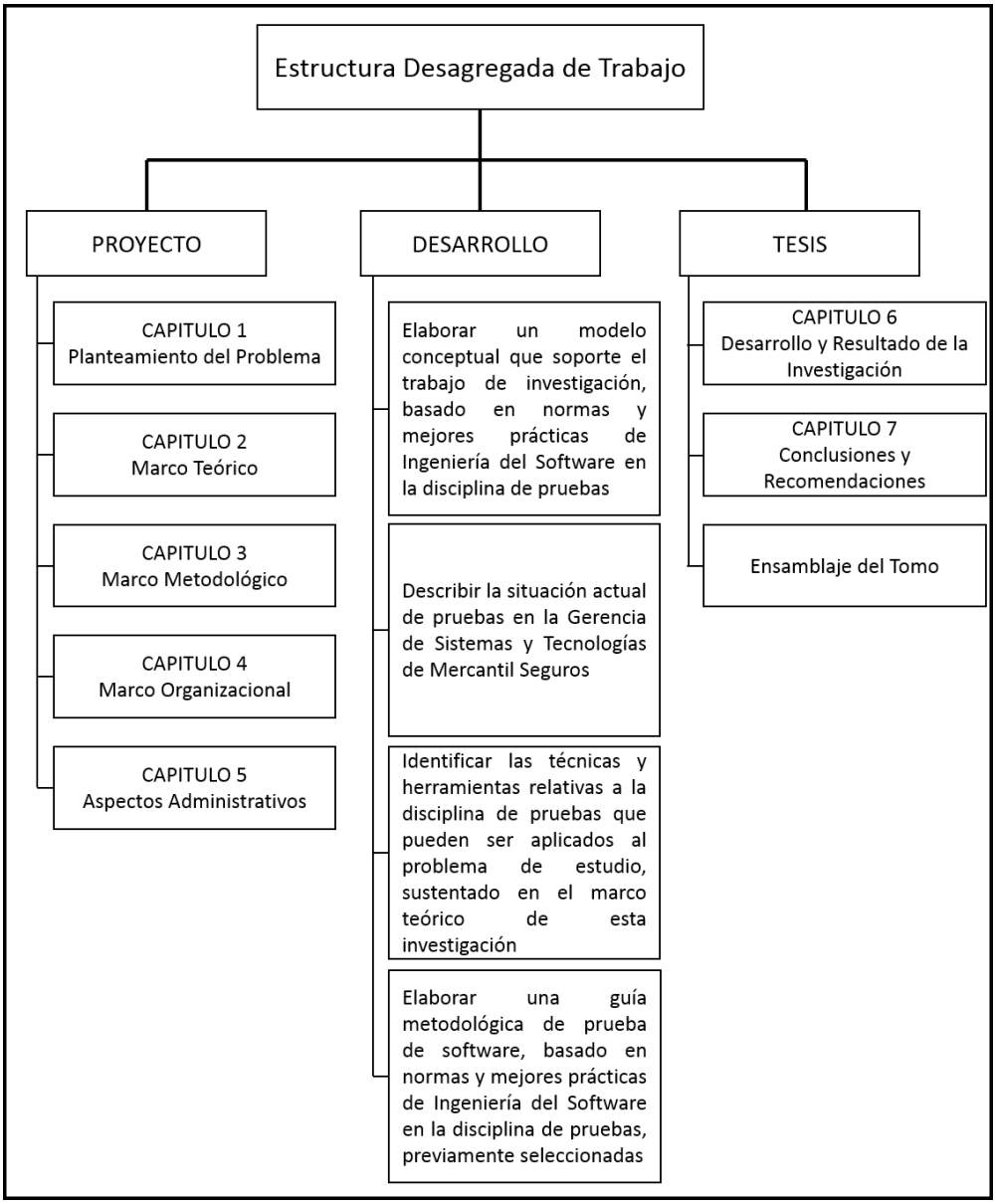


Figura 13. Estructura Desagregada de Trabajo (Santana Cecilia, 2018)

Esta estructura desagregada de trabajo se divide en tres fases:

Fase 1: PROYECTO. En esta fase se elabora el proyecto de investigación, en el cual se presenta el planteamiento del problema, con su marco teórico, metodológico y organizacional, además de la información correspondiente a los aspectos administrativos que serán considerados para la investigación. El entregable de esta fase es el documento que aquí se presenta.

Fase 2: DESARROLLO. Aquí se trabaja con cada uno de los objetivos específicos de la investigación, para lo cual identificaremos las actividades necesarias y los entregables por cada uno de ellas. A continuación se indican esta información por cada objetivo específico:

Objetivo: Elaborar un modelo conceptual que soporte el trabajo de investigación, basado en normas y mejores prácticas de Ingeniería del Software en la disciplina de pruebas

Actividades: Revisión de referencias bibliográficas correspondientes a la actividad de prueba como libros, manuales, revistas de investigación, tesis de grado, entre otras.

Entregable: Modelo conceptual de la investigación.

Objetivo: Describir la situación de pruebas en la Gerencia de Tecnología y Sistemas de Mercantil Seguros.

Actividades: Elaboración de cuestionario y aplicación del mismo en la Gerencia de Tecnología y Sistemas de Mercantil Seguros.

Entregable: Matriz DOFA del proceso de pruebas.

Objetivo: Identificar las técnicas y herramientas relativas a la disciplina de pruebas que pueden ser aplicados al problema de estudio, sustentado en el marco teórico de esta investigación.

Actividades: Revisión documentación sobre el tema de pruebas, además de los entregables de la investigación como son el Modelo conceptual y la Matriz DOFA

Entregable: Herramientas y técnicas que servirán de base para el desarrollo de la propuesta.

Objetivo: Elaborar una guía metodológica de pruebas de software, proveniente de los estándares y mejores prácticas existentes, previamente seleccionadas.

Actividades: Establecer la descripción general de la guía metodológica, indicar los objetivos que persigue y los procesos que serán considerados.

Entregables: Guía Metodológica

Fase 3: TESIS. Aquí se completan los capítulos restantes, a fin de realizar el análisis de los datos para con ello desarrollar la propuesta, junto con sus conclusiones, recomendaciones y el ensamblaje de la misma. El entregable es la Tesis de grado.

CAPITULO IV

MARCO ORGANIZACIONAL

El presente capítulo se refiera a la institución en la cual se desarrolla el trabajo de investigación, su misión, visión, objetivos, valores y el organigrama. Esta información fue tomada del Portal de Mercantil Seguros. (Seguros, 2012)

4.1. Historia de Mercantil Seguros

Mercantil Seguros (antes Central de Seguros), nace en Diciembre de 1988, materializando la decisión del Consorcio Inversionista Mercantil "CIMA" y del Mercantil Banco Universal S.A.C.A. de ampliar la gama de servicios que para ese momento ofrecían a la colectividad, introduciéndose en el mercado asegurador.

En el año 1998 se asocia con Ætna, la aseguradora de salud más importante de Norteamérica con más de 23 millones de afiliados y 140 años de experiencia. Esta alianza estratégica permitió tener de manera rápida y eficiente la tecnología, recursos y respaldos necesarios para continuar como líderes en el mercado asegurador.

En el año 1999, en la búsqueda de la excelencia organizacional y tecnológica, Mercantil Seguros decide desarrollar un proyecto de implantación de nuevas tecnologías orientado a satisfacer las necesidades de innovación y eficiencia, pudiendo competir en un mercado cada día más exigente. Aumenta su capital a Bs. 6.000.000.000. En el año 2002 nace el Nuevo Mercantil Seguros a raíz de su integración con Seguros Orinoco, empresa con más de 44 años de experiencia en el mercado de seguros, incrementando su capital a 15.796.000.000 y reforzando su red de sucursales y agencias conformada por más de 35 puntos de atención en todo el territorio nacional. Este gran paso la convirtió en una de las compañías de seguros más grandes de Venezuela, creando una plataforma de atención y servicio para intermediarios y asegurados orientada a la excelencia y basados en alta tecnología y un gran equipo humano.

4.2. Misión

Satisfacer las aspiraciones del individuo y la comunidad donde actúa, mediante la prestación de excelentes productos y servicios en diferentes segmentos del mercado, agregando valor a sus accionistas mediante la utilización eficiente de los recursos disponibles.

4.3. Visión

Ser la Empresa de Seguros de referencia, en los mercados que servimos

4.4. Estructura Organizativa

4.4.1. Junta Directiva Mercantil Seguros

Tabla 10. Junta Directiva Mercantil Seguros

Directores Principales	Alberto Benshimol **Presidente** Federico Vollmer A. Luis A. Romero Luis A. Sanabria U. Luis E. Palacios Jonathan Coles W. David Brillembourg C.
Directores Suplentes	María Silvia Rodríguez Carlos Hellmund B. Francisco Monaldi Germán Sánchez M. Alexandra Mendoza V. Luis Pedro España Alberto Sosa S.

4.4.2. Organigrama

A continuación se presenta el organigrama de la empresa y seguidamente el organigrama de la Gerencia de Tecnología y Sistemas de Mercantil Seguros, ya que este último ha sufrido modificaciones que no se observan en el organigrama general de la empresa.

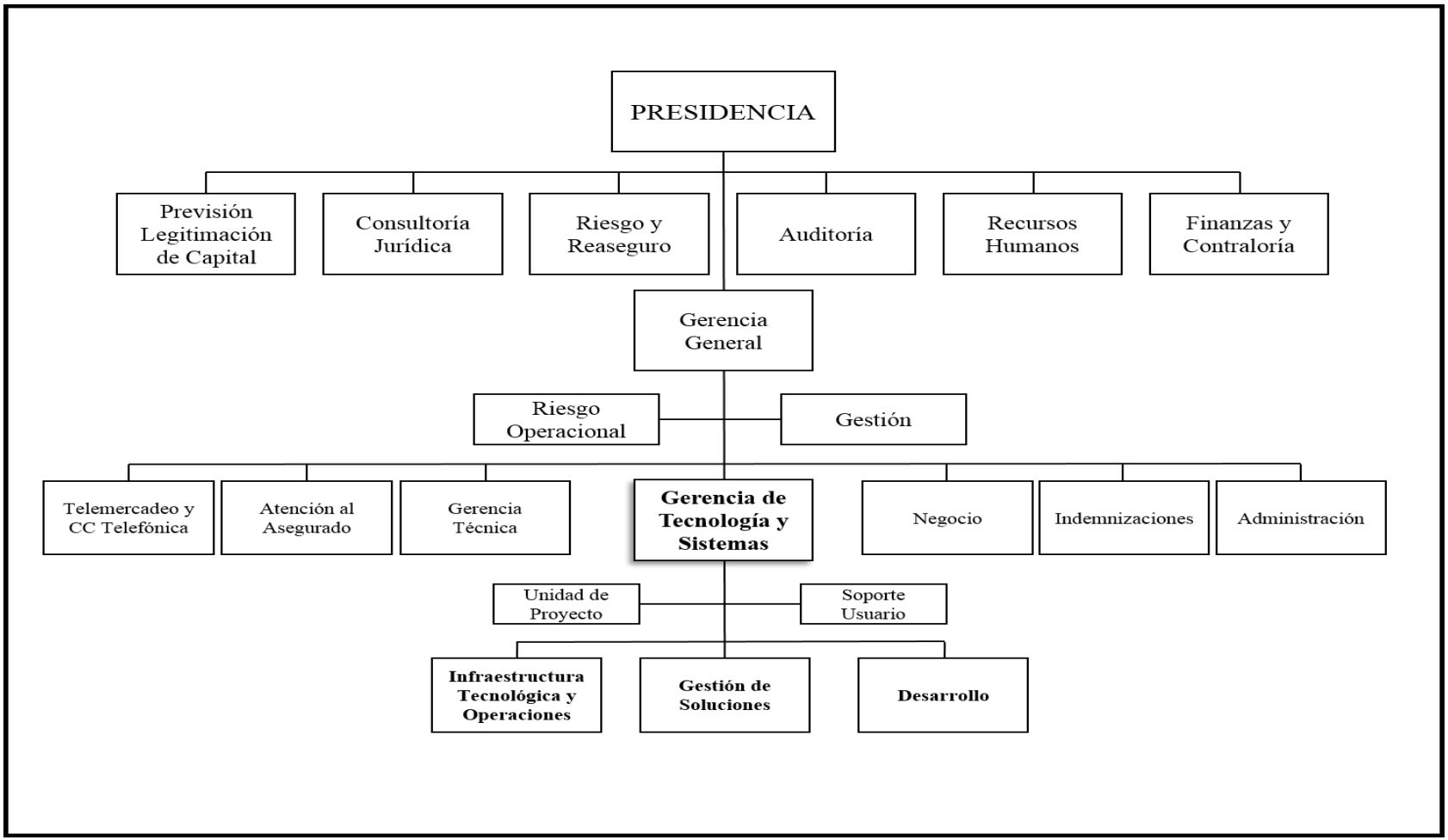


Figura 14. Organigrama Mercantil Seguros (Santana Cecilia, 2018)

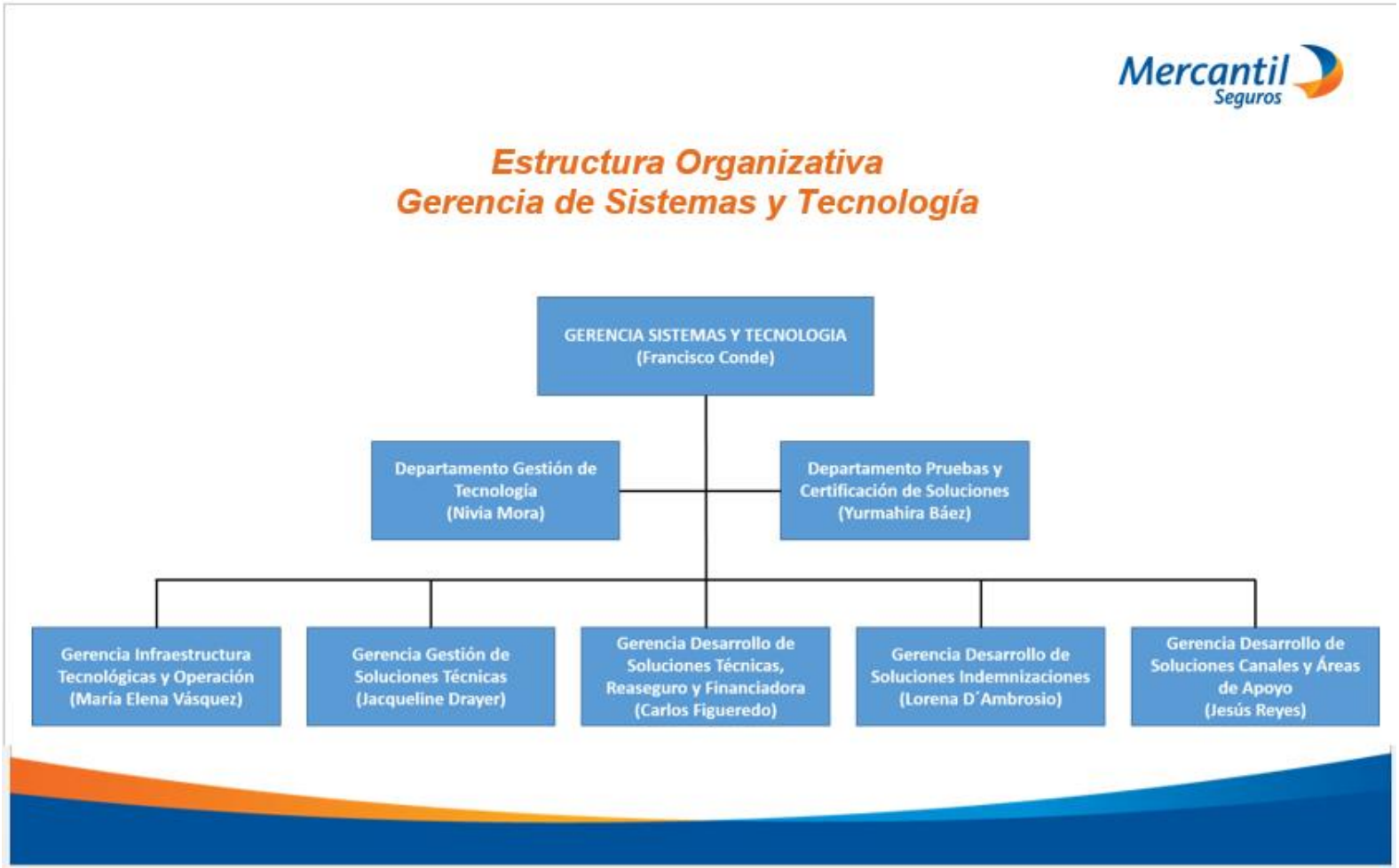


Figura 15. Organigrama Gerencia de Sistemas y Tecnología de Mercantil Seguros (Santana Cecilia, 2018)

CAPITULO V

ASPECTOS ADMINISTRATIVOS DEL PROYECTO

Los aspectos administrativos comprenden un breve capítulo en el cual se expresan los recursos y el tiempo necesario para el desarrollo o ejecución de la investigación (Arias, 2006).

5.1. Recursos Necesarios

- Recursos Materiales: equipos de computación, dispositivos de almacenamiento, material de oficina, como hojas y lapiceros, fuentes de información como libros, revistas, publicaciones electrónicas, Internet, bases de datos, bibliotecas.
- Recursos Humanos: asesor en el desarrollo del trabajo de investigación, con conocimientos sólidos en el área que se encuentre interesado en el seguimiento y culminación del mismo. Así como el personal de Mercantil Seguros.
- Recursos Financieros: cubrir el costo de las unidades de crédito que representa el TEG, incluir costos para material de reproducción y encuadernación.

5.2. Presupuesto estimado

A continuación se muestran los costos a incurrir en el desarrollo de la investigación, expresados en Bolívares (Bs). Los valores mostrados son referenciales.

Tabla 11. Presupuesto tentativo del trabajo de investigación

CONCEPTO	MONTO (Bs)
Pendrive	120.000,00
Horas de navegación por internet (200)	50.000,00
9 crédito de Tesis	150.000,00

CONCEPTO	MONTO (Bs)
Impresión y Encuadernación	250.000,00
Otros Gastos	100.000,00
Total	670.000,00

5.3. Cronograma de actividades

El cronograma, describe en forma progresiva las fases realizadas para asegurar el cumplimiento de las metas y objetivos. Se expresa mediante un gráfico en el cual se especifican las actividades en función del tiempo de ejecución y su porcentaje de avance. El siguiente diagrama de Gantt muestra las fases realizadas para completar la investigación.

Nombre de tarea	% completad	Duración	Comienzo	Fin
Diseñar una Metodología de Pruebas en el Ciclo de Desarrollo del Software en la Gerebncia de Sistemas y Tecnología de Mercantil Seguros, bajo normas, mejores práctivas y el proceso Unificado de Desarrollo de Software (UP)	**100%**	**1109,75 días**	**lun 01/10/12**	**vie 30/12/16**
Proyecto	**100%**	**65,75 días**	**lun 01/10/12**	**lun 31/12/12**
Capitulo 1. Planteamiento del problema	100%	65,88 días	**lun 01/10/12**	**lun 31/12/12**
Capítulo 2. Marco Teórico	100%	65,88 días	**lun 01/10/12**	**lun 31/12/12**
Capítulo 3. Marco Metodológico	100%	65,88 días	**lun 01/10/12**	**lun 31/12/12**
Capítulo 4. Marco organizacional	100%	65,88 días	**lun 01/10/12**	**lun 31/12/12**
Capítulo 5. Aspectos Administrativos	100%	65,88 días	**lun 01/10/12**	**lun 31/12/12**
HITO: Anteproyecto de Investigación	100%	1 día	lun 01/10/12	lun 01/10/12
Desarrollo	**100%**	**42,75 días**	**vie 01/03/13**	**mar 30/04/13**
Elaboración de un Modelo Conceptual que soporte el trabajo de investigación, partiendo de normas, mejores prácticas y el proceso unificado de desarrollo de software en la disciplina de pruebas	100%	21,88 días	vie 01/03/13	sáb 30/03/13
HITO: Modelo Conceptual	100%	1 día	vie 01/03/13	vie 01/03/13
Describir la situación actual de pruebas en la Gerencia de Sistemas y Tecnología de Mercantil Seguros	100%	21,88 días	vie 01/03/13	sáb 30/03/13
HITO: Fortalezas y Debilidades del Proceso actual de Pruebas	100%	1 día	vie 01/03/13	vie 01/03/13
Identificar las técnicas y herramientas relativas a la disciplina de pruebas que puedan ser aplicados al problema de estudio	100%	21,88 días	lun 01/04/13	mar 30/04/13
HITO: Técnicas y herramientas relativas a la disciplina de pruebas	100%	1 día	vie 01/03/13	vie 01/03/13
Elborar una Guía Metodológica de Pruebas de Software, proveniente de las normas, mejores prácticas y del proceso unificado (UP), previamente seleccionado	100%	21,88 días	lun 01/04/13	mar 30/04/13
HITO: Guía Metolodógica	100%	1 día	vie 01/03/13	vie 01/03/13
Tesis	**100%**	21,88 días	jue 01/12/16	vie 30/12/16
Capítulo 6. Desarrollo de la Propuesta	100%	21,88 días	jue 01/12/16	vie 30/12/16
Capítulo 7. Conclusiones y Recomendaciones	100%	21,88 días	jue 01/12/16	vie 30/12/16
HITO: Tesis Especialización	100%	0,88 días	jue 01/12/16	jue 01/12/16

CAPITULO VI

DESARROLLO DE LA PROPUESTA

En este capítulo se desarrollan cada uno de los objetivos planteados por el investigador para el diseño de una metodología de pruebas en el ciclo de desarrollo de software en Mercantil Seguros. A continuación se presentan cada uno de ellos:

Objetivo 1

Elaborar un modelo conceptual que soporte el trabajo de investigación, basado en normas y mejores prácticas de ingeniería del Software en la disciplina de pruebas.

Con el Modelo Conceptual se presentan todos los conceptos, técnicas y modelos considerados a lo largo de la investigación. Se han separado en varios diagramas con la intención de explicar y relacionar de mejor manera, siendo algunos de ellos: casos, tipos, niveles, técnicas de pruebas, tipificación de las pruebas y los modelos de pruebas.

Como indica Edumis Méndez (Méndez, Pérez, & Mendoza, 2007) en la figura número 19 se describen los casos de prueba que están íntimamente relacionados con los requerimientos, pues los casos de uso dan el aporte básico para plantear los respectivos casos. De igual forma, son descritos a través de los procedimientos de pruebas que con sus pasos, condiciones y valores o datos de la prueba, se conjugan para ser candidatos a automatizarse a través de los Scripts. La ejecución de las prueba traen consigo resultados satisfactorios o dignos de atención como lo son las fallas, la cual determinarán finalmente los resultado obtenidos en contraste con los esperados. Por ello los casos se ejecutan en un ambiente de prueba que puede estar conformado por uno o más de los siguientes elementos: software, hardware y base de datos.

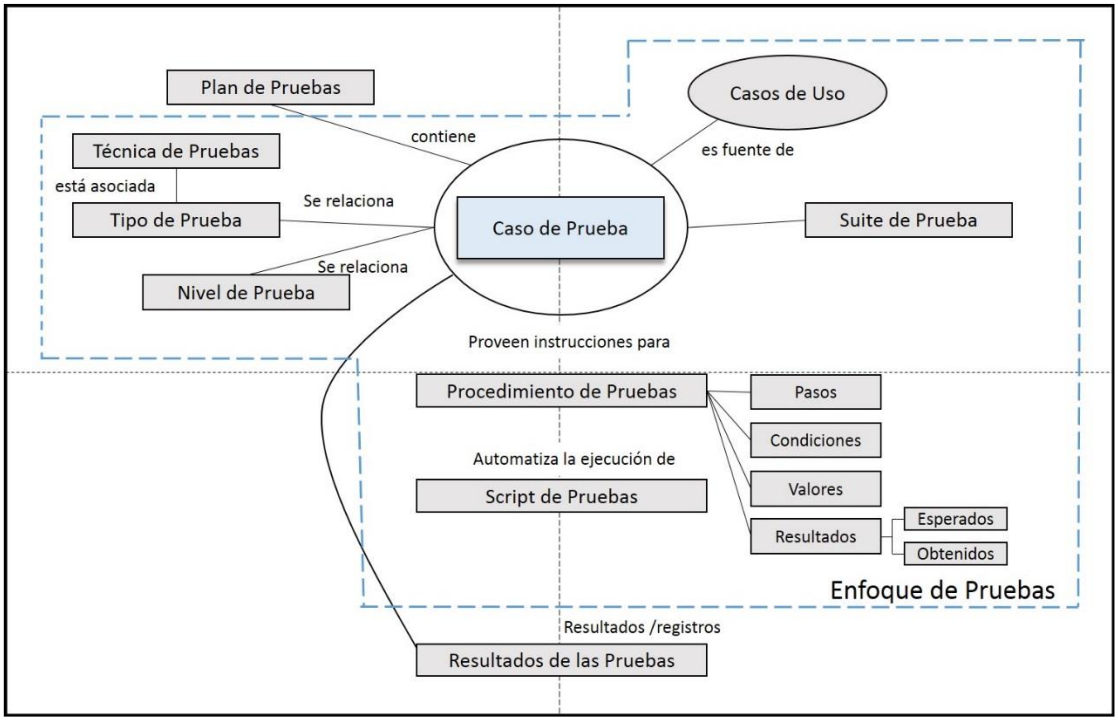


Figura 16 Modelo Conceptual asociado a los casos de prueba (Méndez, Pérez, & Mendoza, 2007)

En virtud que en Mercantil Seguros el sistema de la compañía sufre constantes modificaciones no se considerará el desarrollo de script de pruebas en la Guía Metodológica, ya que el mantenimiento de los mismos impactaría considerablemente en el tiempo de atención por parte del personal del Departamento de Pruebas y Certificación.

Continuando con este bagaje se recrea la figura número 20, donde se presentan ocho (8) tipos de pruebas propuestas por diversos autores, siendo éstos considerados por el investigador como relevante y general entre las diferentes tipologías. Concerniente a los niveles, se presenta la clasificación en función al objeto de la prueba: basados en el desarrollador (pruebas unitarias), con otros módulos (integración) y certificadas (aceptación del usuario). De igual forma existen las técnicas y enfoques de las cuales pueden buscar la verificación (internamente – estructura de implementación) y la validación (externamente – cumplimiento de la funcionalidad) respectivamente.

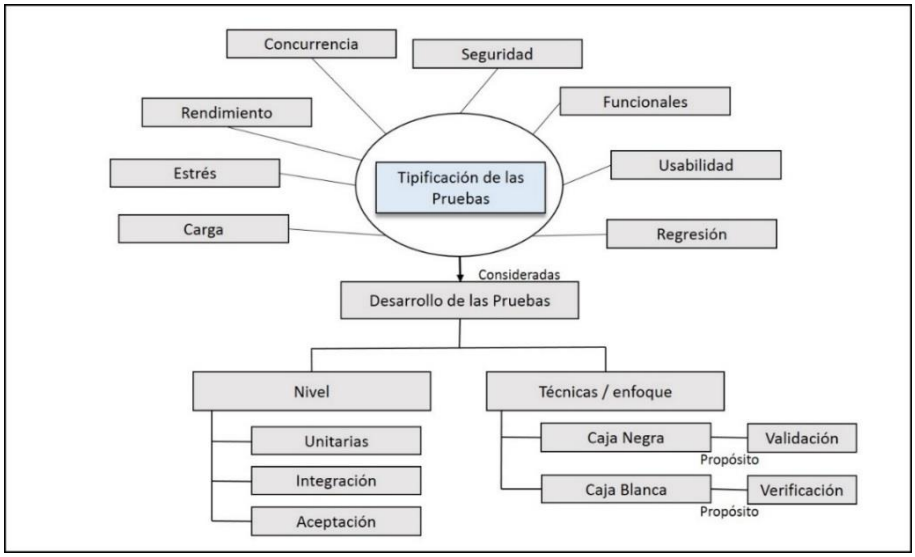


Figura 17 Modelo Conceptual asociado a los tipos, niveles y técnicas de Pruebas

Al articular ambos modelos conceptuales podemos encontrar los elementos que fueron considerados por el autor para llevar adelante esta investigación, partiendo de los casos, que serán aplicados en cada una de las pruebas definidas para el desarrollo del software.

Como puede observarse, en la figura número 21, se muestran los modelos de desarrollo de software mencionados en esta investigación, como son: el espiral, en V, en cascada y UP. A su vez se mencionan los modelos de pruebas, que pueden ser utilizados y de los cuales también se hace referencia en esta investigación, siendo los más destacados: tmm, tmmi, tpi, testpai y tmap.

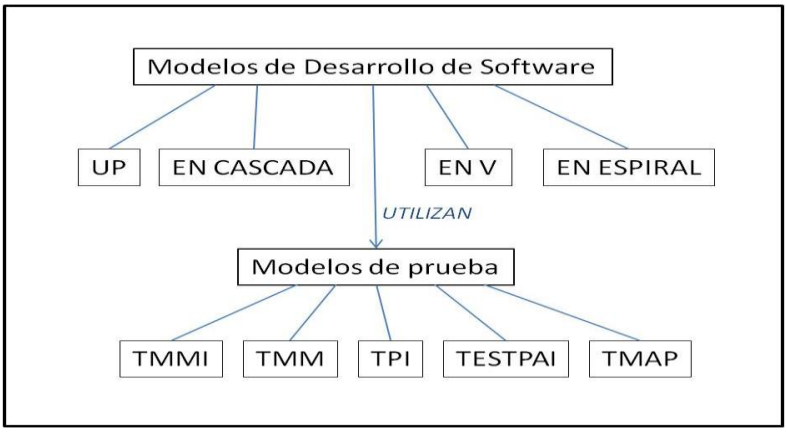


Figura 18 Modelo Conceptual asociado a los Modelos de Desarrollo de Software y de Pruebas

En ese sentido, es necesario indicar que los resultados obtenidos estarán determinados por las características de la organización, no por los objetos a evaluar. Es por ello, que al evaluar el modelo para seleccionar la mejor herramienta de prueba en una organización es muy probable que cada característica varíe a las necesidades y la realidad de la empresa.

Como resultado a la elaboración de un modelo conceptual, En la figura número 22, se presenta la conjunción final de todos los diagramas indicados anteriormente, para el cual se puede observar y contractar la relación simbiótica que tienen con el tema de investigación y las prueba de software.

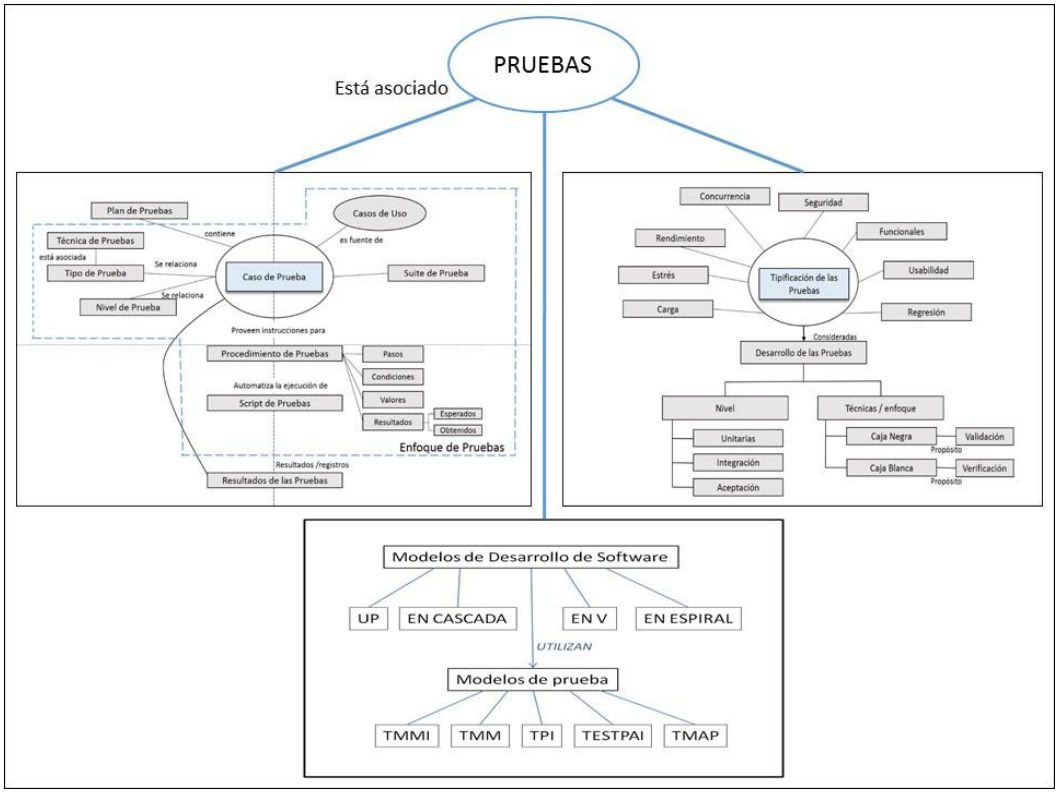


Figura 19 Modelo Conceptual Propuesto

Objetivo 2

Describir la situación actual de pruebas en la Gerencia de Sistemas y Tecnologías de Mercantil Seguros.

Para ello, se preparó una matriz FODA con el fin de obtener conclusiones sobre la forma en que el departamento de Pruebas y Certificación de la Gerencia de Sistemas y Tecnología de Mercantil Seguros será capaz de afrontar los cambios y las turbulencias en el contexto y su ambiente (oportunidades y amenazas) donde a partir de sus fortalezas y debilidades internas.

En el levantamiento de dicha matriz, se entregó a los integrantes del departamento definiciones de cada uno de los factores que intervienen, siendo las siguientes:

Fortalezas: son las capacidades especiales con que cuenta la empresa, y que le permite tener una posición privilegiada frente a la competencia. Recursos que se controlan, capacidades y habilidades que se poseen, actividades que se desarrollan positivamente, etc.

Oportunidades: son aquellos factores que resultan positivos, favorables, explotables, que se deben descubrir en el entorno en el que actúa la empresa, y que permiten obtener ventajas competitivas.

Debilidades: son aquellos factores que provocan una posición desfavorable frente a la competencia, recursos de los que se carece, habilidades que no se poseen, actividades que no se desarrollan positivamente, etc.

Amenazas: son aquellas situaciones que provienen del entorno y que pueden llegar a atentar incluso contra la permanencia de la organización.

Esta actividad se llevó a cabo de la siguiente manera, se entregó una matriz a cada miembro del departamento, para que señalara como máximo cinco (5) enunciados por factor. A razón de la información recabada se efectuó un consolidado o constatación del cual se muestra a continuación:

FORTALEZAS:

F1. Se aplica Metodología para el área de Pruebas y Certificación.

F2. Se cuenta con equipos y aplicaciones necesarias para la actividad de prueba.

F3. Buenas relaciones con las áreas de sistemas y de negocio.

F4. Recurso humano con conocimiento del área de negocio y de los aplicativos utilizados en la compañía.

F5. Recurso humano con alta capacidad analítica y sentido crítico.

F6. Recurso humano motivado, responsable y comprometido con el trabajo.

F7. Existe un excelente ambiente y se fomenta el trabajo en equipo.

F8. Se promueve el aprendizaje y crecimiento profesional.

F9. Se informa oportunamente al personal del departamento sobre los cambios internos o de la Compañía.

F10. El Jefe del área brinda apoyo al personal de área en los proyectos que participa.

F11. Capacidad de redistribuir o re-planificar el recurso humano para proyectos o puntuales en caso de ser necesario

DEBILIDADES

D1. Desconocimiento en aplicación de pruebas para SOA, Movilidad, Usabilidad y BPM.

D2. Se aceptan para pruebas proyectos / puntuales que no cumplen con los requisitos mínimos para su inicio.

D3. Falta de actualización para algunos equipos del departamento.

D4. No hay participación del personal de departamento en todo el ciclo del proyecto / puntual.

D5. No se cuenta con una herramienta que muestre la Gestión del Departamento.

D6. Desconocimiento de nuevas reglas del negocio que puedan estar en un proceso al cual se le aplicará pruebas.

D7. No se cuenta con una Base de Datos del Conocimiento que nos permita detallar rápidamente un proceso sistémico.

D8. Falta de cursos en el área de Aseguramiento de la Calidad.

OPORTUNIDADES

O1. El Desarrollo de aplicaciones bajo SOA, Movilidad y Web, lo cual conlleva a la capacitación del personal.

O2. Herramienta de Medición del Desempeño por área y así manejar indicadores de Gestión.

O3. Entrega de Productos con Calidad a las áreas de negocio.

O4. Posicionamiento del Mercado a través de la mejora en los Productos.

O5. Estándares en la metodología de las diferentes áreas de la Gerencia.

O8. Ambientes de Pruebas para SOA, Movilidad y Web.

O9. Aplicación de encuesta de Satisfacción a los Usuarios Funcionales para mejorar los procesos.

O10. Alinear los Recursos con las demás áreas de Servicio para el cumplimiento de las metas, de acuerdo a los lineamientos manejados.

AMENAZAS

Circunstanciales

AD1. Desviación en la planificación de la ejecución de la prueba.

AD2. Detección de omisiones y/o de incidencias en pruebas (muy tarde).

AD3. Falta de indicadores que nos permitan medir la gestión.

AD4. Estrategias y cobertura Pruebas (incompletas)

Permanentes

A1. El control de versiones se lleva manual, lo cual ha traído como consecuencia pérdida de versiones y repruebas por fallas en producción.

A2. Demasiado tiempo invertido en la resolución de incidencias.

A3. Se presentan constantes desviaciones en cuanto a las fechas de entrega a prueba de los proyectos/puntuales.

A4. Bajo nivel de las pruebas unitarias.

A5. Desmejoramiento en la Calidad de los productos entregados, debido a debilidades en la fase de levantamiento de información y en la fase de diseño.

A6. No se aplica calidad en cada una de las fases del ciclo de desarrollo de los proyectos / puntuales.

A7. No hay un seguimiento oportuno de los proyectos / puntuales para evitar o disminuir las desviaciones que se puedan presentar.

A8. Se formulan por Ingeniería e IT puntuales / proyectos sin documentación y son aceptados para pruebas.

A9. Problemas en los ambientes de prueba causando inestabilidad, lo cual retrasa las pruebas y puede ocasionar re pruebas.

A10. En la Gerencia de Sistema existe desconocimiento de la Metodología de Pruebas que se tiene en el departamento.

A11. En algunas oportunidades existe desconocimiento por parte del Dpto. de Sistema de los procesos que están modificando, con lo cual no se tiene claridad en el alcance de las pruebas.

A12. No se actualiza oportunamente el documento de Especificaciones, acarreando retrabado al momento de realizar el levantamiento de información y escenarios por parte de Pruebas y Certificación.

A13. No hay seguimiento de las incidencias que quedan pendientes por resolver de aquellos proyectos que se instalan en producción.

Una vez construida la matriz se establecieron Estrategias de acción, correspondientes a cada factor. En cada estrategia se identificaron las referencias analizadas que corresponde a los factores.

ESTRATEGIAS:

E1 (D1--D9) Plan de Adiestramiento y Capacitación Integral en todas las áreas necesarias para llevar a cabo los proyectos (QA, SOA, Movilidad, BPM, Usabilidad.)

E2 (D2-A10-A12) Preparar charlas informativas, difusión de información, entre otras, a la Gerencia de Tecnología y Sistemas en cuanto a la metodología utilizada por el Dpto. de Pruebas y Certificación.

E3 (D3-AD2). Plan de actualización de Equipos. (Definir requerimientos mínimos para PC el área)

E4 (D4-AD4-A5-A4-A6) Definir, diseñar, y construir mecanismos de evaluación que permiten medir la calidad del producto en cada una de las fases del desarrollo

E5 (D5-AD5) Evaluación y Adquisición de herramienta para presentar la Gestión de Pruebas y Certificación y sus indicadores.

E6 (D7-AD7-A11- D8). Definir, diseñar y construir mecanismos para que los especialistas de Pruebas participen activamente en la concepción de los productos y/o nuevos procesos, así como también en la actualización de los productos ya existentes.

E8 (A8) Adaptar la metodología de Pruebas y Certificación, para trabajar con los proyectos de Ingeniería e IT, bajo los normas y mejores prácticas.

E9 (A1-A9). Evaluar y adquirir un sistema integral para control de versiones y control de cambios.

E10 (A3-A7). Evaluar y adquirir un sistema integral para la normalización de la planificación, control y seguimiento de proyectos.

E12 (A2).Definir acuerdos de servicios en todas las áreas que intervienen en desarrollos de los sistemas.

Como parte de esta investigación se estará desarrollando la estrategia E8, para lo cual a continuación se identificaron las técnicas y herramientas relativas a la disciplina de pruebas que pueden ser aplicados al proceso de estudio

Objetivo 3

Identificar las técnicas y herramientas relativas a la disciplina de pruebas que pueden ser aplicadas al problema de estudio, sustentado en el marco teórico de esta investigación.

Para esta investigación, se estarán utilizando las técnicas de pruebas dinámicas, siendo a) Basadas en Especificaciones, la cual se concentra en qué hace el software y no en cómo la hace, generando los casos de prueba directamente de la especificación; y b) Basada en Experiencia, son aquellas a las que se recurre cuando no hay una especificación adecuada ni hay tiempo suficiente para ejecutar la estructura completa del paquete de pruebas.

En Mercantil Seguros, la mayoría de los desarrollo cuentan con sus especificaciones, con las cuales se preparan las casos de prueba, tomando para ello las técnicas de partición de equivalencias, análisis de valor frontera, tablas de decisión, transacción de estados y casos de uso. Dependiendo del desarrollo se estará utilizando la técnica que se adecue al mismo. Adicional se utiliza la técnica basada en experiencia, cuando se está preparando la matriz de prueba, se busca al especialista de prueba que tenga más experiencia en el proceso que está inmerso el desarrollo, para que agregue aquellos casos productos de sus experiencias en pruebas anteriores correspondientes al proceso, para que sean incluidos. También una vez realizados los casos de prueba son revisados con el Gestor de Soluciones, persona que tiene toda la información del cambio y que debe tener el conocimiento del proceso involucrado y quien puede brindar información valiosa con su experiencia que puede ser descrita en casos de prueba ya que en la Gerencia no se cuenta con documentación de los procesos.

De igual manera, para los casos de desarrollo en los cuales no hay documentación completa, también utilizamos esta técnica basada en la experiencia, incluyendo al Especialista de Prueba con más experiencia en el proceso que se está desarrollando para que realice los casos de prueba y tenga revisión de los casos de prueba con el Gestor de Soluciones del área correspondiente para que facilite información y con ella terminar de preparar los casos de prueba. Estos son revisados en conjunto para tratar de incorporar todos los casos de prueba necesarios para la revisión del desarrollo.

En el Departamento de Pruebas y Certificación se carece de herramientas automáticas que soporte la gestión, por lo cual se actualizan los formatos utilizados como son: a) Matriz de Casos de Prueba, en la cual se presenta la funcionalidad y aplicación del caso, el detalle del caso de prueba que se debe validar, con sus datos de entrada, las salidas esperadas, un paso a paso para ejecutar el caso, el tipo de prueba que se está aplicando. También tiene campos para asociar el defecto, en el caso que aplique. Maneja una serie de estatus siendo: pendiente, con defecto, corregido (que estuvo como defecto y luego fue satisfactorio) y fallido; b) Matriz de Defectos, en la cual se lleva la información referente a los Defectos que se presentan en la revisión del desarrollo, tiene entre sus campos: abierto y cerrado, se asocia el número del caso en el cual está ubicado, una descripción breve, prioridad de atención y tipo de defecto; c)

Soporte de Pruebas, aquí se lleva la secuencia de las pantallas que se utilizan para validar el caso de prueba, con pequeñas explicaciones que ayuden a entender el resultado que se obtiene y tiene su sección para colocar el defecto y realizar la reprueba, luego de ajustar el defecto asociado al caso de prueba; d) Porcentaje de avance, aquí se lleva el detalle diaria de ejecución de los casos, indicando de antemano por el total de casos a ejecutar las métricas en cuanto a la cantidad de días necesarios para su ejecución como la cantidad de casos de pruebas diarias que deben ser ejecutados satisfactoriamente para cumplir los días de pruebas; e) Informe de Resultado, en el cual se presenta la información del desarrollo de las pruebas para el desarrollo en cuestión, la cual debe ser enviada a las áreas involucradas y es requerida por el área de Control de Cambio para requisito para el pase a producción del desarrollo.

Para más detalle de estos documentos y otros que complementan las herramientas que utilizan se encuentran descritos en la sección de Anexos de este trabajo de investigación.

Objetivo 4

Elaborar una guía metodológica de pruebas de software, basado en normas y mejores prácticas de Ingeniería del Software en la disciplina de pruebas, previamente seleccionadas.

En esta investigación se prepararon una serie de documentos, con los cuales se contemplan estandarizar las actividades que se desarrollan en el departamento y mejorar la presentación de resultados, a continuación el material preparado:

1. **Documento "Guía Metodológica del Proceso de Pruebas y Certificación".**

   Este documento tiene como propósito presentar las normas y procedimientos que forman parte del Proceso de Pruebas, indicando en cada etapa sus premisas, normas y procedimientos. Al final del documento se presenta un glosario con los términos utilizados. Para esta guía se están considerando los requerimientos de tipo Proyecto, SPL's e Incidencias, tanto para el desarrollo realizado por personal interno de Mercantil Seguros como por sus Proveedores. En cuanto a los aplicativos es válido para cualquier tipo que sea objeto de prueba. Esta Guía Metodológica maneja una

sección de anexo con los formatos utilizados, la cual se encuentra en la sección de Anexos de este trabajo de investigación.

Gerencia de Sistemas y Tecnología

Dpto. de Pruebas y Certificación de Soluciones

Proyecto:

Guía Metodológica del Proceso de Pruebas y Certificación

Código: PCS001

(Doc. 1.0)

Julio 2018. Versión 1.0

1. Control del Documento

1.1. Historial de Versiones

Versión	Fecha	Autor	Resumen de cambios

1.2. Tabla de Revisores en *[indicar nombre y fecha de revisión del documento]*

Nombre	Fecha revisión

1.3. Aprobaciones

Este documento requiere las siguientes aprobaciones. Los documentos con las aprobaciones serán guardados en el repositorio del proyecto en *[indicar nombre, título y fecha de aprobación]*

Nombre	Título	Fecha de Aprobación

1.4. Distribución

Este documento se debe distribuir a: *[indicar nombre de las personas a las cuales se les hace llevar el documento]*

Lector	Título

1.5. Ubicación del documento

Este documento se almacena en *[nombre del repositorio y ubicación del documento]*.

1. Tabla de Contenido

2. 1. CONTROL DEL DOCUMENTO 135

1.1. HISTORIAL DE VERSIONES 135

1.2. TABLA DE REVISORES 135

1.3. APROBACIONES 135

1.4. DISTRIBUCIÓN 135

1.5. UBICACIÓN DEL DOCUMENTO 135

3. TABLA DE CONTENIDO 136

4. 3. INTRODUCCIÓN 137

3.1. PROPÓSITO 137

3.2. ALCANCE 137

3.3. PROCESO DE PRUEBAS FUNCIONALES Y SU CERTIFICACIÓN 137

5. 4. NORMAS Y PROCEDIMIENTOS 137

4.1. PLANIFICACIÓN 137

4.2. ANÁLISIS Y DISEÑO 139

4.3. IMPLEMENTACIÓN Y EJECUCIÓN 141

4.4. EVALUACIÓN DE RESULTADOS 143

4.5. ACTIVIDADES DE CIERRE 145

6. 5. CONTROL Y SEGUIMIENTO 146

7. 6. OTRAS PRUEBAS 147

6.1. PRUEBAS UNITARIAS 147

6.2. PRUEBAS DE INTEGRACIÓN 147

8. 7. GLOSARIO DE TÉRMINOS 148

3. Introducción

Un proceso es un conjunto de actividades mutuamente relacionadas o que al interactuar transforman elementos de entrada y lo convierten en resultado. Una prueba es una actividad en la que un sistema o un componente es ejecutado bajo condiciones especificadas, los resultados son observados o registrados, y una evaluación es realizada de un aspecto del sistema o componente. [IEEE Std.610.12-1990]. Las actividades de pruebas buscan el remover los defectos antes de implementar las soluciones de software en el ambiente de producción.
A continuación se presenta el Proceso de Pruebas de Mercantil Seguros, en el cual encontraremos su propósito, alcance, las normas, procedimientos asociados y un glosario de términos.

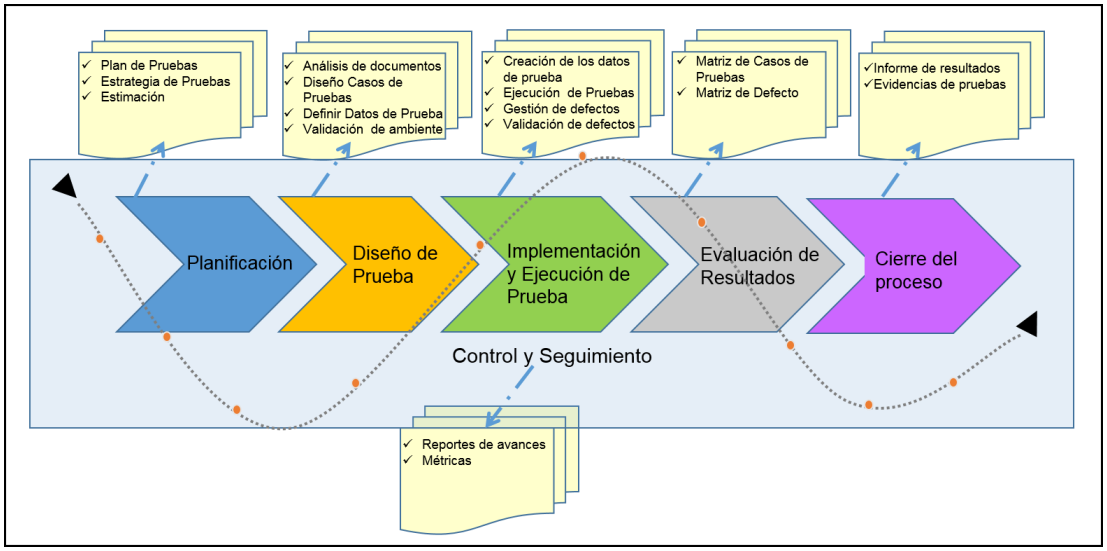
3.1. Propósito

El propósito de este documento es presentar las normas y procedimientos que forman parte del Proceso de Pruebas, indicando de cada etapa sus premisas, normas y procedimientos. Al final del documento se presenta un pequeño glosario de términos.

3.2. Alcance

Este documento abarca los requerimientos de tipo Proyecto, SPL's e Incidencias, tanto desarrollados por Mercantil Seguros como por sus Proveedores. En cuanto a los aplicativos es válido para cualquier tipo que sea objeto de prueba.

3.3. Proceso de Pruebas Funcionales y su Certificación



El proceso de pruebas FUNCIONALES y su CERTIFICACION contemplan las siguientes etapas: (1) Planificación, (2) Diseño de pruebas, (3) Implementación y ejecución, (4) Evaluación de Resultados y (5) Cierre del Proceso. También existe la etapa de Control y Seguimiento que está presente a lo largo de todas las etapas antes descrita. En cada una de estas actividades se encuentran una serie de entregables, los cuales se nombrarán más adelante. A continuación se presentan las normas y procedimientos asociados a cada una de estas etapas.

4. Normas y Procedimientos

4.1. Planificación

Es la etapa en la cual se presentan las primeras actividades para elaborar la estimación del plan de prueba, presentar las estrategias de pruebas y la estimación del tiempo de prueba.

a. Normas

i. La planificación deberá estar aprobada por el área de pruebas y certificación.

ii. La planificación deberá tomar como base las métricas diseñadas para tal fin. (anexo 7.1)

iii. El área de pruebas y certificación son los únicos que pueden realizar ajustes a las métricas y/o actividades asociadas al proceso de pruebas.

iv. El área de pruebas deberá asegurar que el plan del requerimiento (proyecto, SPL o incidencia) cuente con todos los recursos y actividades requeridas. Se debe utilizar como guía el documento "Plan de Pruebas" (anexo)

v. Se establecen los siguientes tiempos para la entregar del plan de pruebas: Proyectos, cinco (5) días; SPL un (1) día e incidentes (4hrs).

vi. El área de Pruebas y Certificación tiene la responsabilidad de alertar oportunamente a las personas participantes en el requerimiento las desviaciones del Plan de Prueba y sus posibles consecuencias.

vii. Se deberá considerar las fechas de los refrescamientos de los ambientes de pruebas y certificación para la planificación de la ejecución y certificación de las prueba, a fin de evitar ser afectados por esta situación.

b. Procedimiento

i. **Objetivo**. Presentar las actividades que se deben llevar a cabo para realizar la planificación de una prueba.

ii. **Alcance**. Esta planificación se presenta en dos momentos: el primero que es para preparar el plan de prueba con la estrategia y estimación de los tiempos de ejecución y el segundo momento que es luego de preparar los casos de prueba se valida la estimación entregada para realizar, si lo amerita, los ajustes correspondientes. Como ya se conocen las fechas de ejecución, se revisan las fechas de refrescamiento de los ambiente. También se presenta el proceso dependiendo del tipo de requerimiento: proyecto, SPL o incidencia.

iii. Responsables

1. Líder de Calidad.
2. Especialista de Calidad.

iv. Criterios de entrada

1. Documento Planteamiento de la necesidad.
2. Documento diseño conceptual.
3. Matriz Casos de Prueba (Anexo)
4. Documentación que soporte el SPL.
5. Documentación que explique la incidencia que se está presentando.

v. Detalle del Procedimiento

1. Tipo de requerimiento: PROYECTO

Responsable	**Actividad**
Área Solicitante	Hace llegar a Pruebas y Certificación el documento "Diseño Conceptual".
Especialista de Pruebas	Recibe el documento "Diseño conceptual" Analiza el documento y genera el "Plan de Pruebas Envía Plan de Pruebas al área solicitante.

Área Solicitante	Recibe Plan de Prueba y actualiza el esfuerzo de pruebas en el plan del proyecto.
Especialista de Pruebas	Una vez preparada la Matriz Casos de prueba se revisa el tiempo de pruebas estimado en el plan de pruebas versus la cantidad de casos de prueba generados y en caso de requerir ajuste en los tiempos previamente estimados se negocia con el área solicitante.
Área Solicitante	Recibe notificación del ajuste de los tiempos de prueba y realiza las negociaciones correspondientes con el área de pruebas y certificación.

2. Tipo de requerimiento: SPL

Responsable	**Actividad**
Área Solicitante	Hace entrega al área de Pruebas y Certificación de toda la información correspondiente al SPL.
Especialista de Pruebas	Recibe la información asociada al SPL Analiza la información y entrega "Plan de Pruebas" Envía Plan de Pruebas al área Solicitante.
Área Solicitante	Recibe Plan de Pruebas y actualiza esfuerzo.

3. Tipo de requerimiento: Incidencia

Responsable	**Actividad**
Área Solicitante	Hace entrega al área de Pruebas y Certificación de toda la información correspondiente a la Incidencia
Especialista de Pruebas	Recibe toda la información la analiza y genera la estimación de los tiempos de prueba. Envía tiempos de prueba al área solicitante.

vi. Criterios de salida

1. Para los proyectos y SPL se entrega el Plan de pruebas con estrategia de prueba y estimación de tiempo.
2. Para las incidencias se entrega información con la estimación del tiempo de pruebas, puede ser vía correo electrónico.

4.2. Análisis y Diseño

Etapa en la cual se realiza el análisis de toda la documentación existen con respecto a la necesidad, requerimiento y sistema para dar inicio a los casos de prueba. También se define cuáles son los datos de prueba necesarios para la ejecución de los casos de prueba diseñados.

a. Normas

i. Para el análisis y diseño de los casos de prueba se deberá contar con el Diseño Conceptual o documentos definitivos para el desarrollo.

ii. Se deberá incorporar aquellos casos de prueba que no son propios del proceso involucrado y aquellos en los cuales se buscan errores.

iii. Se deberá identificar aquellos casos de prueba mínimos necesarios a ejecutar que garanticen la operatividad del desarrollo.

iv. Se llenará el documento Matriz Casos de Prueba.

v. Se deberá identificar en el documento Matriz Casos de Prueba, aquellos a ser utilizado por las Pruebas Unitarias y las Pruebas de Regresión.

vi. Se deberá indicar para cada caso de prueba los datos de prueba necesarios para su ejecución.

b. Procedimiento

i. **Objetivo**. Presentar las actividades que se deben llevar a cabo para realizar el análisis y diseño de los casos de prueba.

ii. **Alcance**. Elaboración de los casos de prueba a ser utilizados para las pruebas unitarias, pruebas de integración y pruebas funcionales.

iii. Responsables

1. Líder de Calidad.
2. Especialista de Calidad.

iv. Criterios de entrada

1. Documento Planteamiento de la necesidad.
2. Documento diseño conceptual.
3. Documento de soporte el desarrollo.

v. Detalle del Procedimiento

Responsable	Actividad
Área Solicitante	Hace llegar a Pruebas y Certificación el documento "Diseño Conceptual".
Especialista de Pruebas	Recibe el documento "Diseño conceptual" y cualquier soporte que complete la documentación del desarrollo.
	Analiza la documentación del desarrollo. Diseña los casos de prueba asociados al desarrollo registrándolos en la matriz "Casos de Pruebas". Identifica los casos a ser utilizados para las pruebas unitarias (si aplica) Identifica los casos a ser utilizados para las pruebas de integración (si aplica) Identifica la prioridad de cada uno de los casos de prueba. Identifica para cada caso de prueba la data requerida para su ejecución. Envía la matriz casos de prueba al área solicitante para su certificación.
Área Solicitante	Recibe Matriz Casos de Prueba para su revisión y certificación. Devuelve la matriz de casos de prueba con las observaciones que correspondan.
Especialista de Pruebas	Recibe la matriz de casos de prueba con sus observaciones. Realiza los ajustes que correspondan. Envía al área solicitante la actualización del tiempo de pruebas, en función a la cantidad de casos de prueba generados

Responsable	Actividad
Área Solicitante	Recibe notificación de los tiempos de prueba y realiza ajustes, en caso de ser requerido realiza negociaciones con el área de pruebas y certificación.

vi. Criterios de salida

1. Matriz de Casos de Prueba definitivo del desarrollo.
2. Información con el tiempo de pruebas real, en función a los casos de prueba preparados.

4.3. Implementación y Ejecución

Etapa en la cual se preparan los datos de prueba necesarios para ejecutar los casos de prueba diseñados y se da la ejecución de estos casos de prueba.

a. Normas

i. Para dar inicio a las pruebas se deberá de contar con las evidencias de los pruebas unitarias y pruebas de integración.

ii. Se deberá validar que se haya instalado el nuevo desarrollo en el ambiente de prueba de manera correcta.

iii. Se debe garantizar contar en el ambiente de pruebas con toda la parametrización necesaria para la prueba.

iv. Las condiciones mínimas requeridas para dar inicio a las pruebas son las siguientes:

1. Se poseen los set de pruebas aprobadas con escenarios claros.
2. El entorno de pruebas es el adecuado para el tipo de pruebas a iniciar.
3. Todos los artefactos requeridos se encuentran disponibles.
4. Se recibe la Versión del Software para pruebas con su correspondiente Reléase y Lista de Chequeo cuando esta aplique.
5. Se reciben las evidencias de las Pruebas Integrales
6. Todos los recursos humanos y técnicos necesarios se encuentran disponibles.

v. Se debe contar con los datos de prueba para comenzar con la ejecución de los casos de prueba.

vi. Se deberán documentar los casos de prueba utilizando el documento Soporte de Pruebas.

vii. Se ejecutarán todo los casos de pruebas que sean factibles en primer ciclo de pruebas, siendo estos todos aquellos que no sean afectados por defectos.

viii. Solicitará corrección de defectos dentro de un ciclo de pruebas, solo en los casos en que la incidencia afecta al 30% o más de los casos. De lo contrario la corrección se hará al final del clico de pruebas y se validará su corrección en el siguiente ciclo de pruebas o Repruebas.

ix. El tiempo de pruebas podrá ser extendido por cambios de alcance. Este tiempo será determinado con el grupo de trabajo.

x. Deberá registrar todo defecto en la Matriz de Defectos y sistema correspondiente.

xi. Deberá informar a todo el equipo el inicio de las pruebas a través del formato Inicio de Pruebas a través de correo electrónico

xii. Deberá informar a todo el equipo de manera diaria el avance de la prueba, a través del formato “Avance de Pruebas” (anexo 8.8) a través de correo electrónico.

xiii. Para presentar el avance diaria de la ejecución deberá preparar el documento Porcentaje (%) de Avance" para incluir el gráfico en el avance diario.

xiv. Deberá comenzar el ciclo de reprueba una vez que hayan sido corregido todos los defectos y se encuentren desplegados en el ambiente de pruebas.

xv. Las pruebas de regresión se realizarán una vez corregidos todos los defectos para garantizar que los defectos corregidos en el proceso de depuración no hayan desencadenado otros tipos de defectos en el sistema

b. Procedimiento

i. Objetivo. Presentar las actividades que se deben realizar en la implementación y ejecución del proceso de pruebas.

ii. Alcance. Comprende la ejecución de los casos de prueba en los ciclos establecidos y la declaración de los defectos encontrados, repruebas y pruebas de regresión.

iii. Responsables

1. Especialista de Calidad.

iv. Criterios de entrada

1. Matriz Casos de Prueba

v. Detalle del procedimiento

Responsable	Actividad
Área de Desarrollo	Solicita actualización del proyecto en el ambiente de pruebas al área de Pruebas y Certificación y de Liberaciones. Entrega soporte de las pruebas realizadas en el ambiente de INTEGRACION.
Especialista de Pruebas	Recibe solicitud de actualización en el ambiente de pruebas del proyecto. Revisa que estén los soportes de las pruebas de Integración en la ruta correspondiente. Autoriza el pase al ambiente de prueba.
Área de Liberaciones	Recibe solicitud de instalación en el ambiente de pruebas del proyecto. Espera la autorización del personal de Pruebas y Certificación. Actualiza el cambio en el ambiente de pruebas.
Especialista de Pruebas	Prepara datos para los casos de prueba. Envía correo electrónico indicando el inicio de las pruebas. Ejecuta los casos de prueba del proyecto por ciclo de prueba. Actualiza la matriz casos de prueba en función a los resultados obtenidos. En caso de encontrar defectos los registra en la matriz de defectos. Envía avance del proyecto diariamente vía correo electrónico.
Área de Desarrollo	Recibe durante la ejecución de las pruebas matriz de defectos diariamente, vía correo electrónico. Realiza los ajustes necesarios para los defectos y los envía al ambiente de integración. Una vez culminado el ciclo inicial de pruebas solicita la actualización de la nueva versión del proyecto con el ajuste a los defectos en el

Responsable	Actividad
	ambiente de pruebas.
Especialista de Pruebas	Recibe solicitud de actualización en el ambiente de pruebas del proyecto. Revisa que estén los soportes de las pruebas de Integración en la ruta correspondiente. Autoriza el pase al ambiente de prueba.
Área de Liberaciones	Recibe solicitud de instalación en el ambiente de pruebas del proyecto. Espera la autorización del personal de Pruebas y Certificación. Actualiza el cambio en el ambiente de pruebas.
Especialista de Pruebas	Ejecuta los casos con defectos Ejecuta los casos de regresión Si el resultado es satisfactorio coordina actividad de pruebas de aceptación con usuario funcional.
Especialista de Pruebas	Ejecución de pruebas de aceptación (UAT) Ejecuta casos de certificación En caso de presentarse defectos u observaciones por parte de los usuarios, los reporta al equipo del proyecto. Prepara documento Avance Sesión de Certificación. Se corrigen los defectos o aclaran las dudas y se notifican al usuario, quien luego de estar conforme autoriza su pase a PRODUCCION.

vi. Criterio de salida

1. Matriz de Casos de Prueba actualizado
2. Matriz de defectos
3. Correo de Inicio de Pruebas
4. Correo de Avance de las pruebas
5. Porcentaje (%) Avance actualizado.
6. Avance Sesión de Certificación (anexo)

4.4. Evaluación de resultados

Etapa en la cual se determina si es posible dar por finalizado un ciclo de pruebas. Para ello se definen una serie de condiciones que permitirán finalizar el proceso de pruebas, comparando los resultados obtenidos contra las condiciones definidas. En esta etapa también se identificará si el proceso de pruebas puede ser suspendido y/o paralizado, debido a aspectos relacionados a la calidad mínima del sistema requerida para el inicio formal de pruebas.

a. Normas

a. Los criterios que pueden suspender las pruebas son los siguientes:

i. Después de la instalación y configuración del sistema, se evidencia problemas o situaciones anormales en cualquiera de sus componentes.

ii. Después de la instalación y configuración del sistema, se evidencia que el ambiente de pruebas no es lo suficientemente estable para la ejecución de las pruebas.

iii. Discrepancia entre la documentación y el sistema.

b. Los criterios que identifican la culminación de un ciclo de prueba:

i. Se ejecutaron todos los casos de pruebas definidos.

ii. Todas las pruebas se ejecutaron de acuerdo a los criterios de evaluación.

iii. Los defectos encontrados en las pruebas fueron corregidos y probados.

b. Procedimiento

i. **Objetivo:** Presentar las actividades que se deben realizar para la evaluación de los resultados en cada ciclo de prueba.

ii. **Alcance:** Análisis de los documentos generados en la etapa de ejecución de las pruebas y presentación de los resultados de la prueba de certificación con usuario funcional.

iii. **Responsables**

1. Especialista de Calidad

iv. **Criterios de entrada**

1. Matriz Casos de Prueba
2. Matriz de defecto
3. Documento Porcentaje (%) de Avance
4. Correo de avance diaria de la prueba

v. **Detalle del procedimiento**

Responsable	Actividad
Especialista de Pruebas	Ejecución ciclo de Prueba -Casos de Prueba Valida que la cantidad de defectos encontrados corresponda a un porcentaje que no supere el porcentaje estimado de defectos. En caso afirmativo suspende la ejecución de las pruebas y re-planifica la actividad de pruebas. En caso que esté por debajo continúa la ejecución hasta culminar todos los casos de prueba o el máximo de casos que permitan los defectos. Se da por finalizada el ciclo de pruebas.
Especialista de Pruebas	Ejecución ciclo de Prueba – Reprueba Valida que todos los defectos fueron corregidos. En caso contrario envía defecto a desarrollo para corrección. Una vez todos los defectos con resultado satisfactorio se finaliza el ciclo de prueba
Especialista de Pruebas	Ejecución ciclo de Prueba – Regresión Valida que todos los casos de prueba seleccionado se ejecuten de manera satisfactoria. En caso contrario envía defecto a desarrollo para su corrección. Una vez todos los casos de prueba con resultado satisfactorio se finalizada el ciclo de prueba.
Especialista de Pruebas	Ejecución de ciclo de Prueba – UAT (Aceptación)

Responsable	Actividad
	Se valida con el usuario funcional que los casos seleccionados se ejecuten de manera satisfactoria. En caso contrario se envía defecto a desarrollo para su corrección. Una vez todos los casos de prueba presentan resultados satisfactorios y el usuario está conforme con su comportamiento se finaliza el ciclo de prueba.

vi. Criterio de salida

1. Matriz de casos actualizado
2. Matriz de defecto actualizado

4.5. Actividades de Cierre

Etapa en la cual se cierran los defectos reportados, se verifica si los entregables planeados han sido entregados y aprobados, finalizar y aprobar las evidencias de las pruebas, se analizan las lecciones aprendidas para aplicar en futuros proyectos.

a. Normas

i. Al finalizar las pruebas se deberá generar el documento Informe de Cierre de Pruebas.

ii. Se deberá de actualizar el documento Resumen de Calidad del Desarrollo.

b. Procedimiento

i. **Objetivo:** Presentar las actividades que se deben realizar para la actividad de cierre del proyecto.

ii. Alcance: Presentación de los resultados obtenidos del proceso de pruebas y certificación.

iii. Responsables

1. Especialista de Pruebas

iv. Criterios de entrada

1. Matriz de Casos de Prueba
2. Matriz de Defectos
3. Porcentaje (%) de Avance
4. Matriz Calidad de Desarrollo

v. Detalle del procedimiento

Responsable	Actividad
Especialista de Pruebas	Ejecución ciclo de Prueba -Casos de Prueba Valida que todos los entregables planteados fueron entregados y aprobados. Se validan la completitud de las evidencias de las pruebas. Se analizan lecciones aprendidas.
Especialista de Pruebas	Ejecución ciclo de Prueba – Reprueba – Regresión - UAT

Responsable	Actividad
	Se cierran todos los defectos reportados. Se valida la completitud de las evidencias de las re pruebas. Se analizan lecciones aprendidas.

vi. Criterio de salida

1. Matriz de Casos de Prueba
2. Matriz de Defectos
3. Informe Cierre de Pruebas

5. Control y Seguimiento

Etapa que se ejecuta durante todo el proceso de prueba, ya que constantemente se está monitoreando el avance en cada proceso de pruebas.

a. Normas

i. Se enviará de manera diaria el avance de las pruebas al equipo del proyecto.

ii. Se enviará al área de Desarrollo la matriz de Defecto para su análisis y corrección correspondiente.

iii. Se solicitará conformidad al usuario final del proceso de certificación del desarrollo y su autorización para el pase a producción.

b. Procedimiento

i. Objetivo: Presentar las actividades que se deben realizar para la actividad de cierre del proyecto.

ii. Alcance: Actividades de seguimiento desde el inicio de las actividades hasta el final del ciclo de pruebas**.**

iii. Responsables

1. Especialista de Pruebas

iv. Criterios de entrada

1. Correo de Inicio de Pruebas
2. Correo de avance de las pruebas
3. Porcentaje (%) de Avance
4. Avance Sesión de Certificación

v. Detalle del procedimiento

Responsable	Actividad
Especialista de Pruebas	Ejecución ciclo de Prueba -Casos de Prueba Al iniciar las pruebas envía correo de Inicio de pruebas al equipo del proyecto. Se envía correo de avance de las pruebas indicando los puntos de atención al equipo del proyecto y aquellas personas que consideren. Se envía correo diario con información de los defectos encontrados al equipo de desarrollo. Se actualiza el porcentaje de avance del proyecto y se incluye la información en el avance diario de las pruebas.

Responsable	Actividad
	Se prepara avance de la Sesión de Certificación con usuario final indicando los acuerdos alcanzados y la conformidad por parte del usuario del desarrollo presentado a la certificación. Se recibe por parte del usuario funcional correo indicando la conformidad con lo presentado en la sesión de certificación y su autorización para el pase a producción del desarrollo.

6. Otras Pruebas

A continuación se presentan otras pruebas que se realizar en Mercantil Seguros y las cuales no necesariamente son realizadas por el área de Pruebas y Certificación.

6.1. Pruebas Unitarias

i. Objetivo:

Probar los componentes específicos del software (módulo o programa) con la finalidad de precisar si éste funciona de acuerdo a las especificaciones, evitando así, la producción de interrupciones o cancelaciones que perjudiquen la fluidez de la prueba de alto nivel.

ii. Definición:

Es una prueba de bajo nivel. En las pruebas unitarias, se analiza la funcionalidad del programa para detectar los comportamientos comprobables discretos que se pueden probar como unidades individuales.

iii. Responsable:

1. Especialista de Desarrollo.

iv. Ambiente:

1. Desarrollo

v. Actividades:

1. Creación de casos de pruebas (matriz de casos de pruebas).
2. Ejecución de la prueba
3. Crear soporte de pruebas
4. Corrección de defectos
5. Elaboración de RFC

vi. Instrumentos:

1. Matriz de casos de pruebas
2. Soporte de prueba
3. RFC

6.2. Pruebas de Integración

i. Definición:

Pruebas realizadas para exponer defectos en las interfaces y en la interacción entre los componentes o sistemas integrados

ii. Objetivo:

Asegurar que las unidades funcionan correctamente cuando trabajan en conjunto; que los cambios generados en cualquier módulo no afecten al resto de la aplicación. El énfasis de esta clase de prueba está en el intercambio de datos entre las operaciones, la verificación del requerimiento y obtener una visión de la estabilidad de la aplicación antes de realizar las pruebas funcionales.

iii. Responsable:

1. Especialista de Desarrollo.

iv. Ambiente:

1. Integración

v. Actividades:

1. Creación de casos de pruebas (matriz de casos de pruebas)
2. Ejecución de la prueba
3. Crear soporte de pruebas
4. Corrección de defectos

vi. Instrumento:

1. Matriz de casos de pruebas
2. Soporte de prueba
3. RFC actualizado.

7. Glosario de Términos

Se presentan un conjunto de términos que se utilizan a lo largo del documento.

7.1. Ciclo de Prueba

Conjunto de actividades del área de pruebas que se realizan para validar que los ajustes realizados en el desarrollo satisfacen los requisitos especificados.

7.2. Defectos

Un desperfecto en un componente o sistema que puede causar que el componente o sistema falle al realizar su función requerida.

7.3. Incidencia

Cualquier evento que ocurre y que necesita investigación.

7.4. Plan de Pruebas

Documento que describe el alcance, método, los recursos y el calendario de las actividades de pruebas previstas.

7.5. Pruebas

El proceso que consiste en todas las actividades del ciclo de vida, tanto estáticos como dinámicas, que tiene que ver con planificación, preparación y evaluación de productos de software y productos relacionados al trabajo para determinar que satisfagan los requisitos especificados, para demostrar que son aptos para el propósito y para detectar defectos.

7.6. Métricas

Una escala de medición y el método utilizado para la medición.

7.7. Re-prueba

Etapa de ejecución para aquellos casos de prueba que han tenido asociados defecto.

CAPÍTULO VII

CONCLUSIONES Y RECOMENDACIONES

En esta sección, son presentadas las ideas principales de cada uno de los capítulos que conforman esta investigación, para plasmar las conclusiones producto de la experiencia obtenida a lo largo del desarrollo de esta investigación, por una parte y por la otra, se exponen las recomendaciones finales como resultado de la hermenéutica planteada, para así brindar bases a futuros trabajo de investigación o mejoras a la Guía Metodológica propuesta.

Conclusiones

Hoy en día, las pruebas constituyen un conjunto de actividades mutuamente relacionadas, que interactúan, trasforman los elementos de entrada, en la que un sistema o un componente es ejecutado bajo condiciones específicas, los resultados son observados, y una evaluación es realizada, de un aspecto del sistema o componente. Para así remover los defectos antes de implementar las soluciones de software en el ambiente de producción.

El ciclo de desarrollo de la propuesta en su (Capítulo V) estuvo concatenado con los objetivos específicos realizados, en cuatro grandes aportes al conocimiento del área de aseguramiento de la calidad del software, los cuales son:

I. Elaborar un modelo conceptual que soporte el trabajo de investigación, basado en normas y mejores prácticas de ingeniería del Software en la disciplina de pruebas.
    - Un modelo conceptual nos ayuda a presentar de una manera gráfica la relación de los diferentes elementos, en este caso asociado a la actividad de prueba.
    - Los casos de prueba están íntimamente relacionado con los requerimientos ya que contienen los casos de uso que serán la base para la prueba a realizarse.
    - Existe gran variedad de tipos de pruebas las cuales tienen que ser consideradas para la actividad de pruebas y se deben seleccionar aquellas que puedan ser

ejecutadas en virtud al desarrollo de software que se presente dentro de la organización y las limitaciones que se puedan tener en cuanto a los equipos necesarios para su realización.

- Se debe considerar los niveles de las pruebas en función al objetivo de las pruebas y de las técnicas y enfoques, a nivel de verificación y validación.

II. Describir la situación actual de pruebas en la Gerencia de Sistemas y Tecnologías de Mercantil Seguros.

- Los miembros del Departamento de Pruebas y Certificación, integrada por Jefe de Departamento, Especialistas de Proyectos y Especialista de Sistemas, manifestaron la necesidad de adaptar la metodología de Pruebas y Certificación, para trabajar con los proyectos de Ingeniería e IT, bajo normas y mejores prácticas.

III. Identificar las técnicas y herramientas relativas a la disciplina de pruebas que pueden ser aplicadas al problema de estudio, sustentado en el marco teórico de esta investigación.

- Las técnicas y herramientas de pruebas permiten diseñar pruebas que validen la lógica interna del desarrollo, descubrir errores en la funcionalidad, el comportamiento y su rendimiento. Además permiten llegar el control de las actividades correspondientes a pruebas.

IV. Elaborar una guía metodológica de pruebas de software, basado en normas y mejores prácticas de Ingeniería del Software en la disciplina de pruebas, previamente seleccionadas.

- El diseño de la Guía metodológica de pruebas permite tener mayor control en las actividades que se están realizando, desde su planificación, ejecución y seguimiento.
- La Guía metodológica presenta los procesos que intervienen a la hora de preparar pruebas para un determinado requerimiento.
- En la Guía metodológica se indican los diferentes formatos que deben ser utilizados al momento de ejecución de las pruebas, información que luego será utilizada para presentar los avances y estadísticas correspondientes.

Recomendaciones

De acuerdo a los resultados obtenidos, se consideran las siguientes recomendaciones para futuros trabajos de investigación, como también profundizar la eficiencia en la toma de decisiones en la organización. En virtud de lo antes descrito, se dan a conocer las recomendaciones:

- Promover la guía metodológica de pruebas, a fin de garantizar la mayor calidad de los productos que se desarrollan, enfatizados en la aplicación del modelo y su innumerable beneficio, a su vez, sensibilizando a la organización sobre la importancia de dar valor a la disciplina de pruebas como ventaja a sus productos.
- Se exhorta a coordinar una capacitación sobre la guía metodológica para el manejo e implementación, donde este adiestramiento permita asimilar los cambios en los procesos, los flujos de información y los beneficios de su aplicación.
- Por esta razón, es recomendable que la aplicación del modelo sea realizada por personal experto en pruebas que tengan muy claro el objeto de evaluación y/o propósito de las pruebas enmarcadas en un proyecto de pruebas, para hacer una acertada selección previa de herramientas que satisfagan los requerimientos explícitamente planteados por la organización para sus mejoras en los procesos.
- Con ello, nos permitiría mantener un servicio de apoyo en el desarrollo óptimo, de tal manera que se tenga garantía de los procesos en el menor tiempo posible, ya que de esto dependerá la disponibilidad del modelo propuesto.
- Los aportes presentados por el autor, no pretenden dar por finalizada la investigación, ya que todo proceso investigativo debe ser sometido a su maduración en el tiempo, a fin de verificar el cumplimiento de la misma, del mismo modo estas reflexiones finales pueden servir como punto de partida para otras investigaciones asociadas al área de Pruebas.

BIBLIOGRAFIA

Amber, S. (3 de 02 de 2007). Entrevista sobre desarrollo agil. (IBM, Entrevistador)

Arias, F. (2006). *El Proyecto de Investigación, Introducción a la metodología científica.* Caracas: Editorial Episteme.

Asamblea Nacional, R. B. (2002). *Ley Orgánica del Sistema Venezolano para la Calidad, Gaceta oficial N° 37.543* . Caracas: Asamblea Nacional.

ASQ, A. S. (20 de Noviembre de 2012). *ASQ.* Obtenido de Glossary: www.asq.org

Astels, D. (2003). *Test - Driven development, a practical guide.* New Jersey: Prentice Hall.

Ballestrini, M. (2002). *Cómo se elabora el proyecto de investigación (6ta edición).* Caracas: BL, Consultores Asociados, Servicio Editorial.

Bernal, C. (2006). *Metodología de la investigación. Para administración, economía, humanidades y ciencias sociales (2da edición).* Mexico: Pearson Educación.

Black, R. (2002). *Gestión del Proceso de Prueba, 2da edición. ISBN 0-471-22398-0.* Wiley.

Boehm, B. (|984). *Veridying and Validating Software Requirements and Design Specifications, IEEE Software volumen 1.*

Braude, E. (2003). *Ingenieria de Software: Una perspectiva orientada a objetos (1era edición)* . Mexico: Alfaomega Grupo Editor.

Bruegge, B., & Dutoit, A. (2002). *Ingenieria del Software Orientado a Objetos (2da edición).* Mexico: Prestice Hall.

Cardona V, C. (2009). *Propuesta Metodológica para la Realización de Pruebas de Software en un Ambiente Productivo.* Medellin: Universidad Nacional de Colombia.

Chrissis, M. B., Konrad, M., & Shrum, S. (2009). *CMMI, Guía para la integración de procesos y la mejora de productos (ISBN: 9788478290963).* Pearson Educación s.a.

Chrissis, M. B., Konrad, M., & Shrum, S. (2009). CMMI, Guía para la integración de procesos y la mejora de productos (ISBN: 9788478290963). En M. B. Chrissis, M. Konrad, & S. Shrum, *CMMI, Guía para la integración de procesos y la mejora de productos (ISBN: 9788478290963).* Pearson Educación.

Corrado Padilla, E. (2001). *Tecnologías de Realidad Virtual: Modelo Edificio Inteligente.* Mexico: Centro de Investigación Científica y de Educación Superior de Ensenada.

Crosby, P. (1980). *Quality is free: The Art of Making Quality certain (1er. ed).* USA: Signet Shakespeare.

España, M. d. (20 de Noviembre de 2012). Métricas V.#. Metodología de Planificación, Desarrollo y Mantenimiento de Sistemas de Información. España, España.

Guajardo, G. (1996). *Administración de la Calidad Total: conceptos y enseñanzas de los grandes maestros de la calidad.* Mexico: Pax.

Hernández Orallo, E. (s.f.). *El Lenguaje Unificado de Modelado (UML).* Recuperado el 15 de noviembre de 2011, de http://www.disca.upv.es/enheror/pdf/ActaUML.PDF

Hernández, R., Fernández, C., & Baptista, M. (2010). *Metolodogía de la investigación (5ta edición).* Mexico: McGraw-Hill.

Hetzel, B. (1998). *La guía completa de pruebas de software, 2da edición ISBN 0-89435-242-3.* Information Sciences Inc.

IEEE. (1990). *610.12 - 1990 Standard Glossary of software engineering terminology.*

IEEE. (1996). *12207 / ISO / IEC 12207 - 1996 Procesos del ciclo de vida del software.*

IEEE. (1997). *1028 - 1997. Estándar para las revisiones de software.*

IEEE. (1998). *829 - 1998 Estándar para la documentación de las pruebas de software.*

IEEE. (2002). *730-2002 Standard for software quality assurance plans.*

IEEE. (2004). *Swebok: Guide to the Software Engineering Body of Knowledge.* USA.

ISO / IEC90003, 2. (2004). *Ingeniería del Software. Directrices para la aplicación de la norma ISO 9001-2000 al software del ordenador.*

ISO IEC 9126-1, 2. (2001). *Ingeniería del software - Calidad del producto del software.*

ISTQB. (09 de enero de 2013). *International Software Testing Qualifications Board*. Obtenido de International Software Testing Qualifications Board: http://www.istqb.org/downloads/syllabi/foundation-level-syllabus.html

ITIL. (15 de Noviembre de 2012). Glosario y abreviaturas de ITIL (Español).

Jacobson, I., Booch, G., & Rumbaugh, J. (2000). *El proceso unificado de desarrollo de softwate. Rational Software Corporation.* Addison - Wesley.

James, P. (1997). *Gestión de la Calidad, Un texto introductorio.* España: Prentice Hall.

Juran, J., Godfrey, A., & Blanton. (1998). *Juran´s Quality Handbook (5ta. ed).* USA: McGraw-Hill Professional.

K., L., & J, L. (2008). *Sistemas de Información Gerencia, Administración de la Empresa Digital.* Mexico: Pearson Educación.

Kaner, C., Bach, J., & Pretichord, B. (2001). *Lecciones aprendidas en Pruebas de Software.* Wiley.

Kaner, C., Falk, J., & Nguyen, H. (1999). *Testing Computer Software, 2da edición ISBN 0471358460.* Wiley.

Kendall, K., & Kendall, J. (2005). *Análisis y Diseño de Sistemas.* México: Pearson.

Kit, E. (1995). *Pruebas de Software en el mundo real: la mejora del proceso, ISBN 0201877562.* Addison Wesley.

Kruchten, P. (2000). *The Rational Unified Process: An Intriduction (2da. ed.).* USA: Adisson-Wesley.

Méndez, E., Pérez, M. A., & Mendoza, L. E. (2007). Aplicación de un Método para Especificar Casos de Prueba de Software en la Administración Pública. *Actas de Talleres de Ingeniería del Software y Bases de Datos, Vol. 1, No. 5*, 47-53. Obtenido de http://www.lisi.usb.ve/publicaciones/13%20pruebas/pruebas_01.pdf

Méndez, E., Pérez, M., & Mendoza, L. (2007). *Aplicación de un método para especificar casos de prueba de software en la Administración Pública. Actas de Talleres de IS Y Base de Datos. Volumen 1.*

Mendez, J. (2006). *Estudios de Metodologías para la implantación de la seguridad en Redes Inalámbricas de área local. Trabajo especial de grado no publicado.* Caracas: Universidad Metropolitana.

Mendoza, M. (2010). *Modelo de Referencia para la Selección de Herramientas de Pruebas como Soporte al Proceso de Desarrollo de Software en PYMES Venezolanas.* Caracas: UCAB.

Morales, J. (20 de diciembre de 2012). *Cacharti. Información, Tecnología y Análisis para las decisiones de TI.* Obtenido de Festion - Gestión de Proyecto - Indice y Estadísticas: www.cacharti.com/?p=308

Moreno, L. (2001). *Gestión de la Calidad y Diseño de las Organizaciones.* España: Prentice Hall.

O'Brien, J. A., & Marakas, G. M. (2006). *Sistemas de información gerencial.* México: McGraw-Hill.

Palella, S. (2006). *Metodología de la Investigación Cuantitativa (2da edición).* Caracas: Fondo editorial de la Universidad Pedagógica Experimental Libertador.

Parada, H. (2010). *Contribución a la Gestión de los Procesos de Pruebas de Software y Servicios. Tesis Doctoral.* Madrid: Universidad Politécnica de Madrid.

Perez L., B. (2006). *Proceso de Testing Funcional Independiente.* Montevideo, Uruguay: PEDECIBA.

Perez, A. (2004). *Guía Metodológica para anteproyectos de investigación.* Caracas: Fondo Editorial de la Universidad Pedagógica Experimental Libertador.

Pfleeger, S. (2002). *Ingeniería del Software: Teoría y Práctica.* Pearson.

PMI, P. M. (2008). *Guia de los fundamentos de la Dirección de Proyectos (Guía del PMBOK) 4ta edición.*

Pol, M., Teunissen, R., & van Veenendaal, E. (2002). *Pruebas de Software, una guia para el enfoque TMap, ISBN: 0-201-74571-2.* Wesley.

Pressman, R. S. (2005). *Ingeniería del Software. Un enfoque práctico* (Septima ed.). México: Mc Graw Hill.

Rational, S. (2003). *Rational Unified Process - Versión 2003.06.15.* IBM Corp.1987-2005.

REA. (10 de diciembre de 2012). *Real academia española*. Obtenido de Sitio web de la Real acandemia española: www.rae.es

Rojas La Roche, T. (2011). Fundamentación de la Organización Virtual como parte integrante de la organización tradicional. *Revista arbitrada del centro de investigación y estudios gerenciales A.C.*, 61.

Rueda, G., & Black, R. (2011). *Fundamento de Pruebas de Software (1era edición).* RBCS.

Sabino, C. (2007). *El proceso de investigación.* Caracas: Panapo de Venezuela.

Sanz Esteban, A. (2011). *Tesis Doctoral, Marco Metodológico para la mejora de las actividades de Validación y Verificación de Productos de Software.* Madrid: Universidad Carlos III de Madrid.

Sanz, A., Saldaña, J., Javier, G., & Gaitero, D. (2008). TestPAI: Un área de proceso de pruebas integrada con CMMI. *REICIS, Revista Española de Innovación, Calidad e Ingeniería del Software*, 8-16.

Scalone, F. (2006). *Estudio Comparativo de los Modelos y Estándares de Calidad del Software. Maestría en Ingeniería de Calidad.* Buenos Aires: Universidad Tecnológica Nacional de Argentina.

Seguros, M. (15 de Diciembre de 2012). *Mercantil Seguro*. Obtenido de Sitio web de Mercantil Seguros: www.segurosmercantil.com

Sogeti. (17 de diciemnbre de 2013). *Ficha de Tmap, Sogeti.* Obtenido de Sito web Sogeti : http://www.es.sogeti.com/Global/Fichas%20comerciales/Ficha_TMap.pdf

Sogeti. (18 de diciembre de 2013). *Los cuatro elementos esenciales, Sogeti.* Obtenido de Sitio web Sogeti: http://www.tmap.net/en/tmap-next/4-essentials

Sommerville, I. (2002). *Ingeniería de Software.* Pearson Educación.

Sommerville, I. (2006). *Software Engineering (8ta. ed.).* USA: Addison-Wesley.

Sotres A., S. G. (2010). La empresa virtual, un nuevo esquema de negocios en la red. *Revista Digital Universitaria Volumen 11, nro 10, UNAM*, 3-5.

Stair, R. M., & Reynolds, G. W. (2010). *Principios de sistemas de información : un enfoque administrativo.* México: Cengage Learning.

Suarez, J., Arias, A., & Fernandez, T. (1996). *Benchmarking de la Función Financiera* . Mc Graw Hill.

SUDESEG, S. d. (18 de diciembre de 2013). *Superitendencia de la Actividad Aseguradora.* Obtenido de Superitendencia de la Actividad Aseguradora: http://www.sudeseg.gob.ve/publico/archivos/estadisticas/esta_merc/merc_10_10_2013.php,

Systems, P. S. (s.f.). *Modelado de Sistemas con UML*. Recuperado el 02 de diciembre de 2011, de http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/doc-modelado-sistemas-uml.pdf

Torossi, G. (16 de Noviembre de 2012). *Provincia del Chaco - Argentina.* Obtenido de Provincia del Chaco - Argentina: http://www.chaco.gov.ar/utn/disenodesistemas/apuntes/oo/ApunteRUP.pdf

Unhelkar, B. (2003). *Process Quality Asurrance for UML - Base Projects (1ra. ed.).* USA: Addison Wesley.

UPEL. (2006). *Manual de Trabajo Especial de Grado de Especialización, Maestria y Doctorado (4ta. edición).* Caracas: Fondo editorial de la Universidad Pedagógica Experimental Libertador.

Velasco Balmaseda, E., & Olaskoaga Larrauri, J. (2005). Modelos de Organización Flexible: Estudio del Sector de la Ma´quina - Herramienta. *Revista de Dirección y Administración de Empresa, número 12*, 187-208.

Whittaker, J. (2000). Que son las Pruebas de Software y por qué son tan difíciles? *IEE Software, Vol 17*, 70-79.

Whitten, J. L., & Bentley, L. D. (2008). *Análisis de Sistemas: diseño y métodos.* México: Mc Graw Hill.

ANEXOS

A continuación los documentos que acompañan cada una de las etapas del proceso de pruebas y certificación, siendo ellos:

Anexo 1	Métricas de Estimación de Tiempos.
Anexo 2	Plan de Pruebas.
Anexo 3	Matriz Casos de Prueba.
Anexo 4	Soporte de Pruebas.
Anexo 5	Matriz de Defectos.
Anexo 6	Resumen de Calidad.
Anexo 7	Porcentaje (%) de Avance del Desarrollo.
Anexo 8	Informe Cierre de Pruebas.
Anexo 9	Correo Inicio de Pruebas.
Anexo 10	Correo Avance de Pruebas.
Anexo 11	Avance Sesión de Certificación.

Anexo 1.- Métricas de Estimación de Tiempos

Estimación Pruebas

Metrica Impacto Bajo (1 a 40 casos)	
Actividad	**Duración**
1. Creación de casos de prueba	1 días
2. Certificación de casos de pruebas	1 día
3. Pruebas Funcionales	
Ejecución (1er ciclo)	3 días
Ejecución (2do ciclo)	1 día
4. Pruebas Certificación Usuario	
Ejecución	1 día
Elaboración de informe	1 día
Total	**9 días**

Metrica Impacto Medio (4 1 a 90 casos)	
Actividad	**Duración**
1. Creación de casos de prueba	3 días
2. Certificación de casos de pruebas	1 día
3. Pruebas Funcionales	
Ejecución (1er ciclo)	6 días
Ejecución (2do ciclo)	2 día
4. Pruebas Certificación Usuario	
Ejecución	2 día
Elaboración de informe	1 día
Total	**15 días**

Metrica Impacto Alto (91 a 150 casos)	
Actividad	**Duración**
1. Creación de casos de prueba	5 días
2. Certificación de casos de pruebas	2 día
3. Pruebas Funcionales	
Ejecución (1er ciclo)	15 días
Ejecución (2do ciclo)	5 día
4. Pruebas Certificación Usuario	
Ejecución	3 día
Elaboración de informe	2 día
Total	**32 días**

Anexo 2.- Plan de Pruebas

Plan de Pruebas

Preparado para

Proyecto ID –xx-xxx- Descriptivo

Versión:	*[1.0]*
Ultimo cambio del documento:	*[dd/mm/aaaa]*
Autor:	*XXXXXXXXXXXXX*
Nombre del documento:	*Plan de Pruebas V1.0.doc*
Ubicación del documento:	*[Nombre repositorio y ubicación del documento]*

Referencias

Título del document	Archivo (Hipervínculo)
Documento de Diseño Conceptual	
Documento de Diseño Técnico	

1 Control del Documento

1.1 Historial de Versiones

Versión	Fecha	Autor	Resumen de cambios

1.2 Tabla de Revisores

Nombre	Fecha revisión
	[dd/mm/yyyy]

1.3 Aprobaciones

Este documento requiere las siguientes aprobaciones. Los documentos con las aprobaciones serán guardados en el repositorio del proyecto en *[indicar persona que aprobará el documento]*.

Nombre	Título	Fecha de Aprobación
[Nombre Aprobador]	*[Título /Rol Aprobador]*	*[dd/mm/yyyy]*

1.4 Distribución

Este documento se debe distribuir a: *[indicar personas a los cuales se les distribuirá el documento]*.

Lector (Nombre/Rol)	Sector

1.5 Ubicación del documento

Este documento se almacena en *[nombre del repositorio y ubicación del documento]*.

2 Introducción

(Se coloca una pequeña presentación de la actividad que se está llevando a cabo en referencia a un proyecto / requerimiento)

2.1 Propósito

(Se coloca lo que persigue con el plan de prueba una pequeña presentación de la actividad que se está llevando a cabo en referencia a un proyecto / requerimiento)

3 Información del Proyecto

3.1 Número y Nombre del Proyecto:

Identificador del Proyecto	Nombre del Proyecto

3.2 Alcance del Proyecto:

(Se indica el alcance que tendrán las pruebas para este proyecto / requerimiento)

4 Objetivos del Plan de Prueba

(Se indica cuales objetivos son los que persigue el plan de pruebas, a continuación se mencionan varios, validar si corresponden todos o algunos para su proyecto / requerimiento)

5 Estrategia de Pruebas

(A continuación se indica la información en cuanto a la estimación de la planificación de la prueba (estimación), el diseño de los casos de prueba y los ciclos de pruebas a ejecutar (generalmente 2) como se espera ejecuta, evaluación y cierre, seguimiento y control)

5.1 Planificación

A continuación se presenta la planificación en cuanto a estimación de esfuerzo para las actividades de prueba: (Plan de Pruebas)

El proceso de planificación se ajusta al comenzar cada ciclo debido a posibles:

a. Atrasos en el desarrollo.
b. Modificaciones en los requerimientos iniciales.
c. Cambios en el alcance del producto.
d. Calidad del producto.

5.2 Análisis y Diseño de casos de prueba

Para el diseño de las pruebas, se tendrán en cuenta aspectos que permitirán encontrar defectos en el período de desarrollo del software:

a. **Alcance**: El alcance de las pruebas estará dado por el marco del Proyecto que se encuentra en desarrollo, está compuesta por:
   - Documento Conceptual
   - Documento de Diseño Técnico.

b. **Ciclo de la Prueba**: Las actividades de la prueba se realizarán para una determinada versión del producto, sobre la cual se ejecutan las pruebas y se reportan los defectos encontrados. A esto lo llamamos ciclo de prueba. Para este proyecto se definen dos (2) ciclos de pruebas.
   - En el primer ciclo se validarán la mayor cantidad de casos de pruebas hasta que algún defecto no permita continuar con la actividad.
   - El segundo ciclo comienza una vez culminado el primer ciclo, iniciando con la validación de los casos que presentan corrección de defectos, luego con aquellos casos que quedaron pendientes del ciclo anterior impactados por los defectos y por último los casos de regresión; con este último se trata de garantizar que las correcciones realizadas no afectaron la funcionalidad ya validada.

5.3 Implementación y Ejecución: Se prepara la batería de datos para la ejecución de los casos de prueba. Se estará comenzando la prueba con la entrega completa del desarrollo y se tiene disponible un recurso para las pruebas.

5.4 Evaluación de Resultados: En cada culminación de ciclo de prueba se estarán comparando los resultados obtenidos contra las condiciones definidas con la finalidad de determinar si puede finalizar dicho ciclo o por el contrario se deben realizar ajustes a la planificación, suspensión y/o paralización de las pruebas

5.5 Actividades de Cierre: Se estarán cerrando los defectos reportados, se verifica que los entregables planeados han sido entregados y aprobados, también finalizados y aprobadas las evidencias de las pruebas. Se prepara el Informe de Cierre de Prueba y se solicita la autorización del Usuario Final para el pase a producción del desarrollo.

5.6 Control y Seguimiento: Se enviará de manera diaria el avance de las pruebas y los defectos encontrados con sus soportes. También se llevarán a cabo reuniones para presentar los casos de prueba y su certificación, para aclarar dudas en cuento a los defectos que se puedan estar presentando y para la presentación del informe final.

6 Uso de Herramientas

(Se colocan las herramientas generalmente utilizadas, pueden variar dependiendo del proyecto / requerimiento)

Las herramientas que se utilizarán a lo largo del ciclo de vida de las pruebas serán las siguientes:

- **a.** Plan de Pruebas
- **b.** Matriz Casos de Prueba
- **c.** Matriz de Defectos
- **d.** Porcentaje de Avance del Proyecto
- **e.** Reporte Calidad de Desarrollo
- **f.** Informe de Cierre de Pruebas

7 Criterios de Prueba

(Esta información debería de mantenerse estándar)

A continuación se indican las pautas para dar inicio, suspensión o finalización a la actividad de pruebas.

7.1 Criterio de Entrada

A continuación se señalan las condiciones mínimas que se deben presentar para iniciar la ejecución de las pruebas:

- **a.** Se poseen los set de pruebas aprobadas con escenarios claros.
- **b.** El entorno de pruebas es el adecuado para el tipo de pruebas a iniciar.
- **c.** Todos los artefactos requeridos se encuentran disponibles.
- **d.** Se recibe la Versión del Software para pruebas con su correspondiente Reléase y Lista de Chequeo cuando esta aplique.
- **e.** Se reciben las evidencias de las Pruebas Integrales
- **f.** Todos los recursos humanos y técnicos necesarios se encuentran disponibles.

7.2 Criterio de suspensión

Los criterios de suspensión impiden la iniciación y/o continuación de las pruebas ante cualquier situación de improvisto que hace que la ejecución de las pruebas no logre grados satisfactorios de probabilidad de éxito.

- **a.** Después de la instalación y configuración del sistema, se evidencia problemas o situaciones anormales en cualquiera de sus componentes.
- **b.** Después de la instalación y configuración del sistema, se evidencia que el ambiente de pruebas no es lo suficientemente estable para la ejecución de las pruebas.
- **c.** Discrepancia entre la documentación y el sistema.

7.3 Criterio de salida

A continuación se señalan los criterios de terminación de las pruebas a ejecutar.

- **a.** Se ejecutaron todos los casos de pruebas definidos
- **b.** Todas las pruebas se ejecutaron de acuerdo a los criterios de evaluación.

c. Los defectos encontrados en las pruebas fueron corregidos y probados.

8 Tipos de Pruebas

(Esta información podría variar dependiendo del desarrollo)

Para llevar a cabo este requerimiento se estarán aplicando las siguientes tipos de pruebas:

- **Pruebas Unitarias**: Tienen como objetivo verificar la funcionalidad y estructura de cada componente individualmente del sistema una vez que ha sido codificado. Es una prueba técnica que es ejecutada por el Equipo de Desarrollo.
- **Pruebas de Integración:** Se realiza con el objeto de asegurar que las unidades funcionan correctamente cuando trabajan en conjunto; que los cambios generados en cualquier módulo no afecten al resto de la aplicación. Es una prueba técnica que es ejecutada por el Equipo de Desarrollo.
- **Prueba Funcional:** Pretenden verificar que la especificación solicitada a nivel de sistema fueron desarrollado según lo solicitado. Se enfocan en garantizar que se cubran los requerimientos funcionales de la aplicación. Esta prueba es aplicada por el área de Pruebas y Certificación.
- **Prueba de Regresión:** Comprueban los efectos de los cambios realizados en los requerimientos, una vez aplicado los ajustes a los defectos encontrados.
- **Pruebas de Aceptación**: Tiene como objetivo validar que la solución desarrollada cumpla con el funcionamiento esperado y permita al usuario de dicho sistema su aceptación, desde el punto de vista de su funcionalidad y de su rendimiento. Estas pruebas son realizadas por el cliente en coordinación con el área de Pruebas y Certificación.

9 Entregables de Pruebas

(Esta información podría varias dependiendo del desarrollo)

De acuerdo al tipo de pruebas ejecutadas puede que el entregable del mismo sea diferente, en el siguiente cuadro se señalan los diferentes entregables por tipo de prueba.

Tipo de Pruebas	Entregables	Responsables
Pruebas Unitarias	• Soporte con evidencia de las Pruebas	Líder de Desarrollo Especialista de Desarrollo
Pruebas de Integración	• Se entregará un documento con evidencias de pruebas	Líder de Desarrollo Especialista de Desarrollo
Pruebas Funcionales	• Matriz de Casos de Prueba	Líder de Calidad

Tipo de Pruebas	Entregables	Responsables
	• Matriz de defectos • Correo de avance diario del proyecto • Evidencias de la ejecución de los casos de pruebas. • Informe de resultados.	Especialista de Calidad
Pruebas de Integración	• Soporte con evidencia de las pruebas	Líder de Desarrollo Especialista de Desarrollo
Pruebas de Aceptación	• Correo de cierre de culminación de pruebas. • Informe final de pruebas.	Líder de Calidad Especialista de Calidad

FIN DEL DOCUMENTO

(Toda la información en los párrafos en color azul es solamente de ayuda y debe ser eliminada antes entregar al usuario final)

Anexo 3.- Matriz Casos de Prueba

Mercantil Seguros

Gerencia de Tecnología y Sistemas

Dpto. de Pruebas y Certificación de Soluciones

Matriz de Casos de Prueba

PROYECTO		Lider de Pruebas		Fecha inicio		Elaborado por	
Ambiente de		Gestor de Solución		Fecha fin		Participantes	
Líder de Desarrollo		Plan de Pruebas		Documento			

Nro. De Caso (Id de Prueba)	Fecha Ejecuci	Funcionalidad	Aplicación	Descripción	Pasos	Datos de Entrada / Precondiciones	Resultado Esperado	Resultado Obtenido	Status	Defecto Asociada	Nivel / Priorid del Caso	Tipo de Prueba	Observaciones
C-001													
C-002													
C-003													
C-004													
C-005													
C-006													
C-007													
C-008													
C-009													
C-010													
C-011													
C-012													
C-013													
C-014													
C-015													
C-016													
C-017													
C-018													
C-019													
C-020													
C-021													
C-022													
C-023													
C-024													
C-025													
C-026													
C-027													
C-028													
C-029													
C-030													
C-031													
C-032													

Anexo 4.- Soporte de pruebas

Evidencias de Pruebas			
Proyecto ● Requer. ○		Aplicación	
Ambiente de Pruebas		Fecha inicio	
Líder de Desarrollo		Fecha fin	
Líder de Pruebas		Documento	
Gestor de Solución		Elaborado por	
Plan de Pruebas		Participantes	

Datos del Caso de Prueba				
Nro. Caso de Prueba	Descripción del Caso	Resultado Esperado	Dato utilizado para la prueba	Estatus

Secuencia de Pantallas

Fecha:
Secuencia de pantallas:

Descripción del Defecto

Fecha:

Anexo de la Reprueba Nro. 01

Fecha:
Secuencia de pantallas:

Anexo 5.- Matriz de Defectos

Mercantil Seguros

Gerencia de Tecnología y Sistemas

Dpto. de Pruebas y Certificación de Soluciones

Matriz de Defectos

ID del Proyecto:			Nombre del proyecto:					Clasificación del Defecto					Fechas Asociadas al Defecto			Días Reporte del Defecto		
Nro. del Defecto	Link con Ubicación del defecto	Nro del Caso de Prueba	Severidad del defecto	Prioridad de atención del defecto	Cantidad de casos afectados por el defecto	Descripción del defecto	Persona que reporta el defecto	Tipo de Defecto	Fase en la que se identifica el defecto	Nivel de pruebas donde se identifica el defecto	Estatus del defecto	Sub Estatus del defecto	Fecha Reporte del Defecto	Fecha Corrección del Defecto QD, QE	Fecha Corrección de Error de QA y de Ambiente	Cant. De días desde el reporte hasta la corrección del defecto	Cant. De días desde el reporte hasta la corrección del error Qa y de	Observaciones
									Pruebas	Integración/Sistema						0	0	
									Pruebas	Integración/Sistema						0	0	
									Pruebas	Integración/Sistema						0	0	
									Pruebas	Integración/Sistema						0	0	
																0	0	
																0	0	
																0	0	
																0	0	
																0	0	
																0	0	
																0	0	
																0	0	
																0	0	
																0	0	
																0	0	
																0	0	
																0	0	
																0	0	

Anexo 6.- Resumen de Calidad

Mercantil Seguros

Gerencia de Tecnología y Sistemas
Gerencia de Desarrollo
Depto. De Pruebas y Certificación

INFORMACIÓN GENERAL DEL PROYECTO	
Identificación del Proyecto	Empresa que Desarrolla
0	0

INFORMACIÓN DE LOS CASOS DE PRUEBA			
Cant. Casos Diseñados	Cant. Casos Probados	Cant. Casos No Aplican	% Avance
0	0	0	#¡DIV/0!

INFORMACIÓN DE LAS INCIDENCIAS			
Cant. Defectos Encontradas	Cant. Casos Afectados por Defectos	Impacto % de los Defectos Relacionadas con la Calidad del Desarrollo	% de Día Resolución de Defectos
0	0	#¡DIV/0!	0

INFORMACIÓN DE LA PLANIFICACIÓN					
Fecha Inicio Estimada de las Pruebas	Fecha Fin estimada de las Pruabs	Duración estimada días	Fecha Inicio Real de las Prueba	Fecha Fin Real de las Pruebas	Días de Desvío del Plan Inicial
00/01/1900	00/01/1900	0	00/01/1900	00/01/1900	0

Anexo 7.- Porcentaje (%) de Avance de Desarrollo

Estimación del % de Avance de las Pruebas

% Estimado de Regresión	15,00%
% Estimado de Incidencias	30,00%
Ejecución de casos Diario	16
Ciclos de pruebas Regresión	1
Casos Diseñados:	211

Cantidad Casos Reprueba:	64
Cantidad Casos de Regresión:	32

1. Debe manejarse por separado el avance de ejecución de las pruebas Integrales, Regresión e incidencias.
2. En la planificación debe considerar el tiempo de repruebas; la estimación de los casos debe ser un 30% de los casos diseñados para las pruebas integrales.
3. En la planificación debe considerar el tiempo de Regresión , la estimación de los casos debe ser un 15% de los casos diseñados para las pruebas integrales y adicional hay que considerar cuantos ciclos de regresión se ejecutaran.

1

Planificación				Ejecutado										Pendientes por Ejecución	
Duración Planificada (Cant. Días Hábiles)	Cantidad Total Casos de Pruebas	Cantidad de Casos de Pruebas Exitosos Promedio diario a Ejecutar	Porcentaje de Avance Promedio Diario Estimado	Cant. Casos Pruebas Ejecutados: 1/2 Jornada 29/10/14		Cant. Casos Pruebas Ejecutados: 1 Jornada 30/10/14		Cant. Casos Pruebas Ejecutados: 1/2 Jornada 31/10/14		Cant. Casos Pruebas Ejecutados: 1 Jornada 03/11/14		Cant. Casos Pruebas con estatus "No Aplican"	Porcentaje de Avance a la fecha 06/11/2015	Cant. Casos Pruebas Pendientes por Ejecutar	Porcentaje de Casos Pruebas Pendientes por Ejecutar
				E	F	E	F	E	F	E	F				
13	211	16	8%	0	0	0	0	0	0	0	0	0			
Porcentaje de Avance				0%		0%		0%		0%			0%	211	100%
Desviación del avance planificado				16		16		16		16					

2

Planificación Reprueba				Ejecutado										Pendientes por Ejecución	
Duración Planificada (Cant. Días Hábiles)	Cantidad Total Casos de Pruebas	Cantidad de Casos de Pruebas Exitosos Promedio diario a Ejecutar	Porcentaje de Avance Promedio Diario Estimado	Cant. Casos Pruebas Ejecutados: Jornada 1		Cant. Casos Pruebas Ejecutados: Jornada 2		Cant. Casos Pruebas Ejecutados: Jornada 3		Cant. Casos Pruebas Ejecutados: Jornada 4		Cant. Casos Pruebas con estatus "No Aplican"	Porcentaje de Avance a la fecha 06/11/2015	Cant. Casos Pruebas Pendientes por Ejecutar	Porcentaje de Casos Pruebas Pendientes por Ejecutar
				E	F	E	F	E	F	E	F				
4	64	16	25%	7	0	4	1	1		1		1			
Porcentaje de Avance				11%		6%		2%		2%			25%	48	75%
Desviación del avance planificado				9		12		15		15					

3

Planificación Regresión				Ejecutado										Pendientes por Ejecución	
Duración Planificada (Cant. Días Hábiles)	Cantidad Total Casos de Pruebas	Cantidad de Casos de Pruebas Exitosos Promedio diario a Ejecutar	Porcentaje de Avance Promedio Diario Estimado	Cant. Casos Pruebas Ejecutados: Jornada 1		Cant. Casos Pruebas Ejecutados: Jornada 2		Cant. Casos Pruebas Ejecutados: Jornada 3		Cant. Casos Pruebas Ejecutados: Jornada 4		Cant. Casos Pruebas con estatus "No Aplican"	Porcentaje de Avance a la fecha 06/11/2015	Cant. Casos Pruebas Pendientes por Ejecutar	Porcentaje de Casos Pruebas Pendientes por Ejecutar
				E	F	E	F	E	F	E	F				
2	32	16	50%	7	0	4	1	1		1		1			
Porcentaje de Avance				22%		13%		3%		3%			44%	18	56%
Desviación del avance planificado				9		12		15		15					

Anexo 8.- Informe Cierre de Pruebas

Informe de Resultados de Pruebas
Requerimiento: XX-XXX-XXXXXXXXX

Preparado por

Departamento de Pruebas y Certificación de Soluciones

Versión:	
Ultimo cambio del documento:	
Autor(es):	
Nombre del documento:	
Ubicación del documento:	***[Nombre repositorio y ubicación del documento]***

1 Control del Documento

Historial de Versiones

(Se deben de colocar las diferentes versiones del documento, indicando la fecha, personas y un breve resumen del cambio)

Versión	Fecha	Autor	Resumen de cambios

1.1 Tabla de Revisores

(Como revisores deben estar los líderes de proyecto, gestor de soluciones, jefe de pruebas u otro que se considere)

Nombre	Fecha revisión
(Nombre del revisor)	*[dd/mm/yyyy]*

1.2 Distribución

Este documento se debe distribuir a:

Lector (Nombre/Rol)	Gerencia / Depto

1.3 Ubicación del documento

Este documento se almacena en (*nombre del repositorio y ubicación del documento).*

Tabla de Contenidos

1 Control del Documento 181

Historial de Versiones 181
1.1 Tabla de Revisores 181
1.2 Distribución 181
1.3 Ubicación del documento 181

2 Introducción 183

2.1 Propósito 183
2.2 Referencias 183
2.3 Notación 183

3 Alcance 183

4 Información General 183

5 Estrategia de Pruebas 184

5.1 Metodología 184
5.2 Tipos de pruebas y herramientas 184
5.3 Herramientas para las pruebas 184

6 Información del Ambiente de Pruebas 184

7 Datos de Prueba Utilizados 185

8 Resultado de las Pruebas 185

9 Estatus de Defectos de las Pruebas 185

10 Resumen 186

11 Indicadores de Calidad del Desarrollo 186

11.1 Indicador del Tiempo de Ejecución de la Prueba 187
11.2 Indicador de Efectividad de los Casos de Prueba 187
11.3 Tiempo Promedio de Solución de Defectos 187

2 Introducción

Este documento explica el resumen del cierre de las actividades de pruebas con los resultados respectivos del XXXXXXXX.

2.1 Propósito

El propósito del informe de resultados de pruebas es proveer la información sobre el estatus de la ejecución de las pruebas y los defectos reportados sobre el Proyecto XXXXXXXXXXXXXX.

2.2 Referencias

A continuación se enumeran la documentación referenciada:

Título del Documento	Archivo

2.3 Notación

DEF: Documento de Especificación Funcional
DET: Documento de Especificación Técnica
PPR: Plan de Pruebas

3 Alcance

Está delimitado a presentar los resultados de las prueba ejecutadas para los requerimientos de XXXXXXXXXXX. En este caso abarca las siguientes funcionalidades.

REQ	Funcionalidad /Caso de Uso

4 Información General

Duración de las pruebas

Para esta ejecución se tenía estimado una duración de XX días.

Fecha de Inicio	Fecha planificada de Culminación	Fecha real de Culminación	Cantidad de Casos de Pruebas a Ejecutar
Dd/mm/aaaa	Dd/mm/aaaa	Dd/mm/aaaa	XX

Para la ejecución de estas pruebas funcionales se encuentran diseñados un total de XX **Casos de pruebas correspondientes al Proyecto / Requerimiento XXXXXX.** (No coloque detalle de la planificación, hay otra sección para colocar esa información)

5 Estrategia de Pruebas

5.1 Metodología

Enfoque de pruebas:

El objetivo de estas pruebas es validar que el sistema realice de forma correcta las funcionalidades solicitadas para XXXXXXXX.

5.2 Tipos de pruebas y herramientas

Pruebas Integradas Funcionales

El objetivo de estas pruebas es validar el comportamiento esperado según indicado en los documentos funcionales y diseño conceptual definido para el requerimiento de XXXXXXX.

5.3 Herramientas para las pruebas

Estas pruebas fueron realizadas con las siguientes herramientas:

- **Word**: Herramienta para el registro de los casos de prueba del recorrido funcional.
- **Excel**: Herramienta para el registro de los defectos detectadas en las pruebas.

6 Información del Ambiente de Pruebas

El ambiente de pruebas utilizado para las pruebas fue el ambiente de (XXX) y la certificación con el usuario se realizó en el ambiente XXX. A continuación el control de los pases que se realizaron en el ambiente: **(En caso de no llevar el control de los pases al ambiente no se coloca información en esta tabla, solamente se indica el ambiente)**

Requerimiento	Versión Entregada	Fecha Entrega	Versión Instalada	Fecha Instalación
REQ XXXXX	XXXXXX	dd/mm/aaaa	XXXXXXX	dd/mm/aaaa

7 Datos de Prueba Utilizados

(De manera genérica, se indica por cada funcionalidad los datos de entrada que fueron considerados para los casos de prueba)

Descripción Funcionalidad	Datos de Entrada

8 Resultado de las Pruebas

Durante el tiempo de ejecución de las pruebas funcionales del aplicativo "XXXXXX", en las funcionalidades señaladas en el alcance del documento se obtuvieron los siguientes resultados

Casos de Pruebas	Cantidad
Casos de pruebas Exitoso	04
Casos de pruebas Pendiente	00
Casos de pruebas Fallido	00
Casos de pruebas No aplican	02
TOTAL	06

Planificación				Ejecutado														Pendientes por Ejecución	
Duración Planificada (Cant. Días Hábiles)	Cantidad Total Casos de Pruebas	Cantidad de Casos de Pruebas Exitosos Promedio diario a Ejecutar	Porcentaje de Avance Promedio Diario Estimado	Cant. Casos Pruebas Ejecutados: 1/2 Jornada 10-02-2016		Cant. Casos Pruebas Ejecutados: 1 Jornada 11-02-2016		Cant. Casos Pruebas Ejecutados: 1Jornada 12-02-2016		Cant. Casos Pruebas Ejecutados: 1 Jornada 15-02-2016		Cant. Casos Pruebas Ejecutados: 1 Jornada 16-02-2016		Cant. Casos Pruebas Ejecutados: 1 Jornada 17-02-2016		Cant. Casos Pruebas con estatus "No Aplican"	Porcentaje de Avance a la fecha 17/02/2016	Cant. Casos Pruebas Pendientes por Ejecutar	Porcentaje de Casos Pruebas Pendientes por Ejecutar
				E	F	E	F	E	F	E	F	E	F	E	F				
1	6	6	100%	1	0	1	1	0	0	1	0	0	0	1	0	2			
Porcentaje de Avance				17%		17%		0%		17%		0%		17%			100%	0	0%
Desviación del avance planificado				5		5		6		5		6		5					

9 Estatus de Defectos de las Pruebas

Defectos	Cantidad
Defectos Reportados	01
Defectos Resueltos y certificaos	01
Defectos pendientes	00
TOTAL DE DEFECTOS:	01

Estos defectos se distribuyen de la siguiente manera:

Defecto	Cantidad
Defectos por Error de QA	**0**
Defectos por Omisión de Desarrollo	**1**
Defectos por Omisión de Especificación	**0**
Defectos por problema de Ambiente	**0**
TOTAL DE DEFECTOS:	**1**

10 Resumen

(En este punto se deben indicar los acuerdos surgidos durante las pruebas o la certificación, así como cualquier otra información que sea relevante)

En la ejecución de estas pruebas se lograron ejecutar xxx casos de pruebas de forma exitosa, se certificaron xxx defectos, por lo que se dan por finalizadas las pruebas funcionales del Req XXXX.

Se realizó la Certificación con los usuarios, quedando conformes para su pase a Producción.

11 Indicadores de Calidad del Desarrollo

Mercantil Seguros

Gerencia de Tecnología y Sistemas
Dpto. De Pruebas y Certificación

INFORMACIÓN GENERAL DEL PROYECTO	
Identificación del Proyecto	Empresa que Desarrolla
400-158	Global Fusion

INFORMACIÓN DE LOS CASOS DE PRUEBA			
Cant. Casos Diseñados	Cant. Casos Probados	Cant. Casos No Aplican	% Avance
18	18	0	100%

INFORMACIÓN DE LOS DEFECTOS			
Cant. Defectos Encontradas	Cant. Casos Afectados por Defectos	Impacto % de los Defectos Relacionadas con la Calidad del Desarrollo	Promedio de Día Resolución de Defectos
2	2	11	2

(Agregar cuadro resumen de indicadores)

A continuación se presentarán los resultados correspondientes a los indicadores de: tiempo de ejecución, Defectos que No aplican, efectividad de los casos de prueba y el tiempo promedio de solución de defectos

11.1 Indicador del Tiempo de Ejecución de la Prueba

Este indicador nos presentará la desviación que se pueda presentar entre el tiempo de pruebas planificado y el tiempo de prueba real.

Información de Planificación							
Fecha Inicio Estimada de las Pruebas	Fecha Fin estimada de las Pruabas	Ejecución estimada en días	Fecha Inicio Real de las Prueba	Fecha Fin Real de las Pruebas	Ejecución real en días	Días de Desviación en Ejecución	Días de Desvío del Plan Inicial
02/05/2017	15/05/2017	5	02/05/2017	22/06/2017	6	1	20

(Se coloca el detalle de las desviaciones que se presentaron en la planificación, tanto de la duración estimada vs. La duración real y la fecha fin estimada versus la fecha fin real)

11.2 Indicador de Efectividad de los Casos de Prueba

Este indicador nos permite verificar la efectividad de nuestros casos de prueba a la hora de encontrar errores. Los valores esperados dependerán fuertemente del tipo de desarrollo, sin embargo, son esperados números cercanos a 1.

Este indicador se calcula de la siguiente forma:
ECP = CPEE / CPE
Donde:
ECP = Efectividad de los Casos de Pruebas
CPEE = Es la cantidad de casos de prueba ejecutados con error
CPE = Es cantidad de casos ejecutados

Para este Proyecto / Requerimiento los números son:

CPEE= XX
CPE= XX

Por lo que la Efectividad de Casos de Prueba (ECP) es de: **(X,XX)** lo cual nos presenta que fueron efectivos los casos planteados para la detección de errores en el desarrollo.

11.3 Tiempo Promedio de Solución de Defectos

Este indicador nos permite identificar cuando el tiempo promedio de solución de las defectos reportadas sobrepasa ciertos límites pre-establecidos en acuerdos o SLA, por lo que es importante monitorear este indicador.

Este indicador se calcula de la siguiente forma:
DPS = (∑ DTR) / CIR

Donde:
DPS = Días Promedio de Solución
DTR = Cantidad de días transcurridos desde el reporte del defecto hasta la entrega con la corrección.
CIR= Cantidad de Defectos reportadas.

Para este Proyecto / Requerimiento los números son:

DPS= XX
∑DTR= XX

Por lo que el Tiempo promedio de solución de los defectos reportadas fue de (XX) días.

FIN DEL DOCUMENTO

Anexo 9. Correo de inicio de pruebas

INICIO DE PRUEBAS

Fecha:	/ /

1. Alcance de la Prueba
(Indique aquí las expectativas del equipo solicitante sobre la prueba, mencione la cantidad de casos de prueba y tiempo estimado para las pruebas y re pruebas)

2. Estrategia de Prueba
(Indique la estrategia planteada para la ejecución de las pruebas))

3. Actividades realizadas a la fecha
(Mencione las actividades asociadas al proceso de pruebas y el % de completitud correspondientes)

Anexo 10.- Correo avance de pruebas

AVANCE DIARIO DE PRUEBAS

Fecha:	/ /

4. Ejecución General de la Prueba	
TOTAL CASOS DE PRUEBA	
Cantidad de casos de prueba Exitosos	
Cantidad de casos de prueba Fallidos	
Cantidad de casos de prueba afectados por defectos	
Cantidad de casos de prueba que No aplican	
Cantidad de casos de prueba Pendientes	
Cantidad de casos de prueba en re-prueba	

5. Puntos de Atención	
Nro.	**Descriptivo**

6. Defectos	
TOTAL DEFECTOS	
Cantidad de defectos pendientes de resolución	
Cantidad de defectos que No aplican	
Cantidad de defectos Cerradas	
Cantidad de defectos Pospuestos	
Cantidad de casos de prueba Pendientes	
Cantidad de defectos pendientes por Validación QA	

7. Métricas de Avance de la ejecución del Proyecto

Planificación				Ejecutado										Pendientes por Ejecución	
Duración Planificada (Cant. Días Hábiles)	Cantidad Total Casos de Pruebas	Cantidad de Casos de Pruebas Exitosos Promedio diario a Ejecutar	Porcentaje de Avance Promedio Diario Estimado	Cant. Casos Pruebas Ejecutados: 1/2 Jornada 29/10/14		Cant. Casos Pruebas Ejecutados: 1 Jornada 30/10/14		Cant. Casos Pruebas Ejecutados: 1/2 Jornada 31/10/14		Cant. Casos Pruebas Ejecutados: 1 Jornada 03/11/14		Cant. Casos Pruebas con estatus "No Aplican"	Porcentaje de Avance a la fecha 06/11/2015	Cant. Casos Pruebas Pendientes por Ejecutar	Porcentaje de Casos Pruebas Pendientes por Ejecutar
				E	F	E	F	E	F	E	F				
13	211	16	8%	0	0	0	0	0	0	0	0	0			
Porcentaje de Avance				0%		0%		0%		0%			0%	211	100%
Desviación del avance planificado				16		16		16		16					

Anexo 11.- Avance Sesión de Certificación

SESION DE CERTIFICACION

Fecha:	/ /

PARTICIPANTES	
Nombre	**Unidad administrativa**

CASOS DE CERTIFICACION		
Nro.	**Descriptivo**	**Resultado Obtenido**

PUNTOS DE ATENCION		
Nro.	**Descriptivo**	**Responsable**

OBSERVACIONES

Elaborado por:	