

UNIVERSIDAD PARA LA COOPERACION INTERNACIONAL  
(UCI)

NOMBRE DEL PROYECTO

PROPUESTA DE METODOLOGÍA DE LA GESTIÓN DE LA CALIDAD PARA LOS  
PROYECTOS DE DESARROLLO SOFTWARE EN LA EMPRESA GBSYS S.A.

NOMBRE DEL ESTUDIANTE

MARIA ANGÉLICA VILLALOBOS ACUÑA

PROYECTO FINAL DE GRADUACION PRESENTADO COMO REQUISITO  
PARCIAL PARA OPTAR POR EL TITULO DE MASTER EN ADMINISTRACION  
DE PROYECTOS

San José, Costa Rica

Agosto, 2018

UNIVERSIDAD PARA LA COOPERACION INTERNACIONAL  
(UCI)

Este Proyecto Final de Graduación fue aprobado por la Universidad como  
Requisito parcial para optar al grado de Máster en Administración de Proyectos

---

Jorge Trejos  
PROFESOR TUTOR

---

Paula Villalta  
LECTOR No.1

---

Luis Emilio Ramírez  
LECTOR No.2

---

María Angélica Villalobos Acuña  
SUSTENTANTE

## **DEDICATORIA**

**Cada mérito académico es gracias a mis padres Flor y Guillermo, sus enseñanzas son las que perdurarán para siempre.**

## **AGRADECIMIENTOS**

Mis agradecimientos a mi familia por sus palabras de motivación y apoyo. A mis compañeros de clase con los cuales tuve el placer de conocer y trabajar. A mis profesores, lectores y tutor por el tiempo invertidos y sus recomendaciones.

## INDICE

HOJA DE APROBACION	ii
DEDICATORIA	iii
AGRADECIMIENTO	iv
INDICE	v
INDICE ILUSTRACIONES	vii
INDICE CUADROS	viii
INDICE DE ACRÓNIMOS Y ABREVIACIONES	
ix	
RESUMEN EJECUTIVO	x
1 INTRODUCCION .....	1
Antecedentes .....	1
1.1. Problemática. ....	1
1.2. Justificación del problema .....	2
1.3. Objetivo general .....	3
1.4. Objetivos específicos.....	3
2 MARCO TEORICO .....	4
2.1 Marco institucional .....	4
2.1.1 Antecedentes de la Institución .....	4
2.1.2 Misión y visión. ....	4
2.1.3 Estructura organizativa.....	5
2.1.4 Productos que ofrece .....	6
2.2 Teoría de Administración de Proyectos .....	6
2.2.1 Proyecto.....	6
2.2.2 Administración de Proyectos.....	7
2.2.3 Ciclo de vida de un proyecto.....	8
2.2.3.1 Ciclo de vida de un producto de Software.....	9
2.2.3.2 Características de los ciclos de vida de Software .....	10
2.2.3.3 Fases del proyecto .....	11
2.2.4 Procesos en la Administración de Proyectos .....	11

2.2.5	Áreas del Conocimiento de la Administración de Proyectos .....	12
2.3	Gestión de la calidad .....	15
2.3.1	Calidad y atributos de calidad de Software .....	16
2.3.2	Proceso de pruebas de software .....	17
2.3.3	Modelo en V (Modelo de desarrollo secuencia).....	21
2.3.4	Niveles de prueba.....	22
2.3.5	Tipos de pruebas .....	24
2.3.6	Casos de prueba .....	25
2.4	Metodología Ágiles .....	26
2.4.1	Scrum.....	27
3	MARCO METODOLOGICO. ....	31
3.1	Fuentes de información .....	31
3.1.1	Fuentes Primarias .....	31
3.1.2	Fuentes Secundarias .....	31
3.2	Métodos de Investigación.....	35
3.3	Herramientas.....	38
3.3.1	Tormentas de ideas.....	39
3.3.2	Análisis FODA. ....	39
3.3.3	Juicio de Expertos .....	39
3.3.4	Reuniones .....	39
3.3.5	Entrevistas .....	40
3.3.6	Diagrama de Flujos .....	40
3.3.7	Estudio Comparativos .....	40
3.4	Supuestos y Restricciones. ....	41
3.5	Entregables.....	44
4	DESARROLLO.....	46
4.1.	Objetivo 1. Determinar la situación actual y deficiencias de la empresa GBSYS.SA en el área de gestión de la calidad.....	46
4.1.1	Roles y tareas.....	46
4.1.2	Procesos, pruebas y herramientas .....	48
4.1.3	Análisis de la gestión de calidad actual.....	59

4.2	Objetivo 2. Realizar la una comparación de la situación actual de empresa en gestión de calidad de software respecto a los procesos y técnicas de calidad descritas en la norma para software IEEE 829, el estándar ISO/IEC 9126 y en el marco de conocimiento Foundation Level Syllabus de ISTQB...	63
4.2.1	Comparación de procesos .....	63
4.2.2	Comparación entre tipos y técnicas de prueba .....	67
4.2.3	Comparación entre documentación de prueba .....	69
4.2.4	Comparación de herramienta de gestión. ....	73
4.3	Objetivo 3. Determinar las oportunidades de mejora en la gestión de la calidad de software tomando como fuente el análisis realizado. ....	74
4.4	Objetivo 4. Diseñar una guía de los procesos y técnicas para la gestión de la calidad de software que se ajusten a las necesidades de la empresa considerando el análisis elaborado .....	78
4.4.1	Gestión de calidad .....	79
4.5	Objetivo 5. Crear un plan de implementación para los involucrados para transmitir el conocimiento sobre la propuesta de metodología en gestión de calidad de software y la apliquen en sus proyectos. ....	103
4.5.1	Objetivos del plan de implementación .....	104
4.5.2	Plan de Implementación .....	104
4.5.3	Capacitaciones .....	107
1	CONCLUSIONES .....	110
2	RECOMENDACIONES .....	111
3	BIBLIOGRAFIA .....	112
4	ANEXOS .....	114
	Anexo 1: ACTA DEL PFG .....	114
	Anexo 2: EDT .....	120
	Anexo 3: CRONOGRAMA .....	121
	Anexo 4: Entrevista sobre el proceso de gestión de calidad de la empresa.....	123
	GBSYS S.A. ....	123
	Anexo 5: Encuestas de sobre el proceso de pruebas, técnicas y herramientas utilizadas por el equipo de calidad de software .....	125

## ÍNDICE DE FIGURAS

Figura 1. Estructura organizacional GBSYS.SA. ....	5
Figura 2. Ciclo de vida de Software.....	9
Figura 3. Ciclos de vida de proyectos de software y sus características.....	10
Figura 4. Correspondencia entre grupos de procesos y áreas de conocimiento de la dirección de proyectos de la Guía del PMbok.....	14
Figura 5. Modelo V .....	21
Figura 6. Procesos de calidad de software en el ciclo de vida del proyecto en GBSYS.SA.....	49
Figura 7. Niveles de pruebas en el ciclo de desarrollo de software en GBSYS.SA .....	50
Figura 8. Plantilla actual de caso de prueba de la empresa GBSYS.SA.....	52
Figura 9. Lista de casos de prueba en la herramienta Testlink.....	54
Figura 10. Registro de casos de prueba en la herramienta Testlink.....	55
Figura 11. Registro de ejecución de planes de prueba de en la herramienta Testlink.....	56
Figura 12. Informes y métricas generados por la herramienta Testlink.....	56
Figura 13. Lista de incidencias en la herramienta Mantis Bug Tracker .....	57
Figura 14. Registro de incidencias en la herramienta Mantis Bug Tracker .....	58
Figura 15. Resumen y métricas de las incidencias en la herramienta Mantis Bug Tracker.....	59
Figura 16. Diagrama de causa y efecto del proceso de gestión de calidad de GBSYS.SA.....	75
Figura 17. Plan de pruebas en la herramienta Testlink .....	88
Figura 18. Ingreso del Build asociado al plan de pruebas en el Testlink .....	89
Figura 19. Actividades de los procesos de la propuesta de metodología de gestión de calidad para proyectos de desarrollo software para GBSYS.SA .....	93
Figura 20. Caso de Prueba en la herramienta Testlink .....	95
Figura 21. Ciclo de vida de una incidencia en el Mantis BT.....	101
Figura 22. EDT del seminario del proyecto de graduación. ....	120
Figura 23. Cronograma del seminario del proyecto de graduación. ....	121



Figura 24. Diagrama de Gantt del Seminario de graduación. ....122

## ÍNDICE DE CUADROS

Cuadro 1 Procesos de la administración de proyecto. ....	11
Cuadro 2 Actividades del proceso de pruebas .....	18
Cuadro 3 Niveles de pruebas .....	22
Cuadro 4 Tipos de pruebas .....	24
Cuadro 5 Niveles de un caso de prueba .....	25
Cuadro 6 Roles y eventos de Scrum.....	27
Cuadro 7. Fuentes de Información Utilizadas .....	32
Cuadro 8. Tipos de Observaciones .....	35
Cuadro 9. Métodos de Investigación Utilizadas .....	36
Cuadro 10. Herramientas Utilizadas.....	40
Cuadro 11. Supuestos y Restricciones .....	42
Cuadro 12. Entregables .....	44
Cuadro 13. Roles y tareas del equipo de control y aseguramiento de la calidad de la empresa GBSYS en la actualidad. ....	47
Cuadro 14. Datos recolectados de entrevistas al equipo de trabajo de GBSYS.SA .....	59
Cuadro 15. Comparación de los procesos y actividades de calidad expuesta en el marco del conocimiento ISTQB y los procesos de calidad realizados por GBSYS.SA.....	63
Cuadro 16. Comparación de las técnicas de diseño de pruebas expuesta en el marco del conocimiento ISTQB y las técnicas de diseño realizadas por GBSYS.SA .....	68
Cuadro 17. Comparación de documentación de pruebas de software expuesta en la norma IEEE 829 y los documentos realizados por GBSYS.SA .....	70
Cuadro 18. Comparación entre las herramientas para pruebas expuesta en el marco del conocimiento ISTQB y las técnicas de diseños realizadas por GBSYS.SA.....	73

Cuadro 19. Causas del problema en la calidad de software y oportunidades de mejora en los procesos de gestión de calidad de software en la empresa GBSYS.SA.....	75
Cuadro 20. Grupos de procesos de la gestión de proyecto y actividades del área de conocimiento de gestión de calidad .....	80
Cuadro 21. Propuesta de plan maestro de pruebas para la empresa GBSYS.SA	85
Cuadro 22. Plantilla base para estándar de interfaz gráfica.....	90
Cuadro 23. Plantilla base para estándar de base de datos .....	91
Cuadro 24. Plantilla de caso de prueba para GBSYS.SA .....	94
Cuadro 25. Selección de particiones de equivalencia .....	96
Cuadro 26. Propuesta para tabla de decisiones .....	97
Cuadro 27. Lista de verificación para validar la usabilidad de una aplicación de software .....	98
Cuadro 28. Métricas para pruebas de rendimiento.....	99
Cuadro 29. Informe de seguimiento de pruebas para GBSYS.SA.....	102
Cuadro 30. Registro de lecciones aprendidas para el proceso de pruebas .....	103
Cuadro 31. Plan de Implementación de la propuesta de metodología gestión de calidad para GBSYS.SA. ....	105
Cuadro 32. Plan de contenidos para capacitación de herramientas de automatización para GBSYS.SA .....	107
Cuadro 33. Plan de contenidos para capacitación del proceso de calidad y técnicas de diseño de casos de prueba para GBSYS.SA .....	108
Cuadro 34. Plan de contenidos para capacitación de la metodología propuesta de gestión de calidad de software para GBSYS.SA. ....	109

## **INDICE DE ACRÓNIMOS Y ABREVIACIONES**

GBSYS: Global Business System

IDE: Integrated Development Environment

ISTQB: International Software Testing Qualifications Board

PMBOK: Project Management Body of Knowledge (Fundamentos para la Dirección de Proyectos).

PMI: Project Management Institute, Inc.

PFG: Proyecto Final de Graduación.

QA: Quality assurance

UCI: Universidad para la Cooperación Internacional.

WBS: Work break down structure / Estructura de desglose de trabajo (EDT).

## **RESUMEN EJECUTIVO**

Global Business System (GBSYS S.A.) inició operaciones el 16 de julio de 1987, agrupando las actividades particulares de consultoría de un grupo de profesionales en computación, para dedicarse al desarrollo de sistemas de información en ambientes de bases de datos, contando para tal efecto con la colaboración de un selecto grupo de profesionales en diferentes especialidades de la informática. Actualmente la empresa no cuenta una metodología donde se detalle los procesos, procedimientos, herramientas y técnicas para el control y aseguramiento de calidad lo que origina algunos problemas en el desarrollo del producto debido a revisiones exhaustivas, atrasos en los vistos buenos del departamento de QA, reiteración de errores entre proyectos, aumentos de costos por una errónea distribución de los integrantes de los equipos.

A pesar del esfuerzo de implementar ciertos conceptos de calidad, todavía no se ha realizado una metodología que alinea los procesos realizados en cada proyecto, uno de los objetivos de este proyecto es realizar una investigación sobre las mejores prácticas en el área de desarrollo de software para crear una metodología que sea la base de los procesos y procedimientos relacionados a la gestión de la calidad para la empresa GBSYS.SA. Con esta propuesta se pretende capacitar al equipo de QA e involucrados directos de cómo realizar su planeaciones, diseño, ejecuciones y procesos de mejora continua para agilizar el ciclo de desarrollo de los proyectos de software de la cartera de proyectos de la empresa.

El objetivo general de este proyecto es desarrollar una propuesta de una metodología para la gestión de la calidad que pueda ser aplicada a los proyectos de desarrollo de software, de tal manera que se estandaricen las prácticas y procesos asociados al control y aseguramiento de la calidad de los productos desarrollados en la empresa. Los objetivos específicos son los siguientes: determinar la situación actual y deficiencias de la empresa GBSYS.SA en el área de gestión de calidad, realizar una comparación de la situación actual de la

empresa respecto a procesos y técnicas de calidad descritas en normas, estándares y marcos de conocimiento para calidad de software, determinar las oportunidades de mejora, diseñar una guía de procesos y técnicas para la gestión de calidad y crear un plan de implementación para los involucrados para transmitir el conocimiento sobre la propuesta de metodología.

La metodología utilizada para la investigación fue de observación y analíticasintética ya que se utilizaron los métodos para observar el proceso actual de gestión de proyectos y calidad de la empresa, obtenida de la información de las entrevistas con los involucrados y posteriormente fue analizada para realizar una comparación entre las prácticas investigadas de fuentes secundarias y procesos realizados en la empresa, con el objetivo de desarrollar una metodología de gestión de calidad que integra un conjunto de procesos y procedimientos de calidad de software.

## 1 INTRODUCCION

### **Antecedentes**

Global Business System (GBSYS S.A.) inició operaciones el 16 de julio de 1987, agrupando las actividades particulares de consultoría de un grupo de profesionales en computación, para dedicarse al desarrollo de sistemas de información en ambientes de bases de datos, contando para tal efecto con la colaboración de un selecto grupo de profesionales en diferentes especialidades de la informática.

Hoy en día es una empresa especializada en Administración de Bases de Datos ORACLE y SQL Server además del desarrollo, mantenimiento y migración de aplicaciones en plataformas Developer de Oracle, Java y .Net. Adicionalmente se cuenta con una línea de distribución de productos de software especializados incluyendo productos propios y otros ofrecidos por casas extranjeras reconocidas mundialmente. También se ofrecen servicios de tercerización (outsourcing) ya sea en soporte técnico para administración de sistemas y bases de datos, así como para desarrollo de aplicaciones.

Actualmente en la empresa labora alrededor de 80 empleados distribuidos en 4 departamentos: ventas, proyectos, conectividad, infraestructurasoporte y calidad y aseguramiento integrados por 7 personas.

El fin de este proyecto es realizar una propuesta de metodología que funcione como una guía de trabajo para los equipos de QA's.

### **1.1. Problemática.**

Actualmente la empresa no cuenta una metodología donde se detalle los procesos, procedimientos, herramientas y técnicas para el control y aseguramiento de calidad lo que desencadena los siguientes aspectos negativos:

- Revisiones exhaustivas por poca planeación de las pruebas lo que origina un aumento en el tiempo de las pruebas.
- Atrasos en los vistos buenos del departamento de QA por no terminar pruebas en paralelo con el desarrollo del proyecto.
- Existen tipos de pruebas que no se ejecutan por falta de capacitación teórica y técnica, lo cual desencadena realizar cambios a último momento o insatisfacción del cliente.
- Se repiten los errores asociados al producto entre proyectos por falta de un procedimiento de aseguramiento de la calidad.
- Falta de conocimientos técnicos relacionados a técnicas de diseño y herramientas de automatización por parte de los integrantes de equipo de aseguramiento y control, lo que provoca un aumento de los costos y el tiempo de prueba.

## **1.2. Justificación del problema**

A principios del 2017, la empresa GBSYS.SA inauguró un departamento de aseguramiento de la calidad (QA) el cual está compuesto de 7 profesionales de calidad. A pesar de los esfuerzo de implementar cierto conceptos de calidad, todavía no se ha realizado una metodología que alinee los procesos realizados en cada proyecto, uno de los objetivos de este proyecto es realizar una investigación sobre las mejores práctica en el área de desarrollo de software para crear una metodología que sea la base de los procesos y procedimientos relacionados a la gestión de la calidad de software. Con esta propuesta se pretende capacitar al equipo de QA e involucrados directos de cómo realizar sus planeaciones, diseños, ejecuciones y procesos de mejora continua para agilizar el ciclo de desarrollo de los proyectos de software de la cartera de proyectos de la empresa.

Beneficios esperados:

- Mejorar los procesos de planeación, diseño y ejecución de calidad de software.



- Reducir los tiempos de revisiones de sistemas por parte de los equipos de QA.
- Disminuir el número de incidencias reportadas por el usuario final.
- Aumentar la confiabilidad de los productos entregados al usuario final.
- Satisfacción del cliente final al obtener productos sin errores.

### **1.3. Objetivo general**

Desarrollar una propuesta de una metodología para la gestión de la calidad, de tal manera que se estandaricen las prácticas y procesos asociados al control de la calidad en los productos de software desarrollados por la empresa.

### **1.4. Objetivos específicos.**

1. Determinar la situación actual y deficiencias de la empresa GBSYS.SA en el área de gestión de la calidad.
2. Realizar una comparación de la situación actual de empresa en gestión de calidad de software respecto a los procesos y técnicas de calidad descritas en la norma para software IEEE 829, el estándar ISO/IEC 9126 y en el Foundation Level Syllabus de ISTQB.
3. Determinar las oportunidades de mejora en la gestión de la calidad de software tomando como fuente el análisis realizado.
4. Diseñar una guía de los procesos y técnicas para la gestión de la calidad de software que se ajusten a las necesidades de la empresa considerando el análisis elaborado.
5. Crear un plan de implementación para los involucrados del proceso de calidad y aseguramiento de software para transmitir el conocimiento sobre la propuesta de metodología en gestión de calidad de software, con el fin de que sea aplicada en sus proyectos.

## **2 MARCO TEORICO**

### **2.1 Marco institucional**

#### **2.1.1 Antecedentes de la Institución**

Esta propuesta se va elaborar en la empresa nacional GBSYS.SA, la cual está ubicada en Curridabat de Costa Rica. GBSYS fue fundada el 1987 por el ingeniero en sistema Gerardo Brenes Trejos el cual desempeñaba funciones como administrador de base de datos. Los primeros servicios de GBSYS.SA fue relacionados al soporte de base de datos, posteriormente se introdujo en el desarrollo de software a la medida y tercerización de servicios. En el 2017 inauguró su departamento de calidad de software con 7 integrantes los cuales son los encargados de controlar y asegurar los productos de software elaborados por el departamento de proyectos, el cual posee una actual cartera de proyectos de 11 proyectos en ejecución. Por particularidades de algunos proyectos, no todos los proyectos que se ejecutan son revisados por los equipos de aseguramiento y calidad. La primera experiencia con el equipo ha sido positiva ya que se ha podido solventar algunas situaciones aprendidas en otros proyectos relacionados a la confiabilidad de los productos entregados, sin embargo, no existe una metodología definida para realizar el proceso, lo cual genera algunos retrasos en las aprobaciones, con este proyecto se pretende estandarizar los procedimientos e incursionar con nuevos procesos novedosos.

#### **2.1.2 Misión y visión.**

GBSYS (2015) indica que su misión es “ofrecer productos y servicios de la más alta calidad en tecnología informática que mejoren la eficacia, eficiencia y rentabilidad de nuestros clientes” y su visión es “ser reconocidos mundialmente como una corporación especializada en tecnología informática.”

La propuesta para este proyecto va relacionada directamente con la misión de la empresa, ya que para ofrecer los mejores productos y servicios es necesario tener

las políticas y procedimientos para garantizar la calidad de los productos entregados al cliente, si bien la empresa ha tratado de implementar tareas relacionadas a calidad dentro el desarrollo de los mismos, falta una estandarización de procesos.

La calidad de los productos es un foco de atención para atraer clientes nacionales y extranjero, si una empresa es reconocida por brindar servicios y productos que producen satisfacción al usuario final esto genera prestigio, lo cual va relacionado a su visión.

### 2.1.3 Estructura organizativa

GBSYS.SA posee la siguiente estructura organizacional:

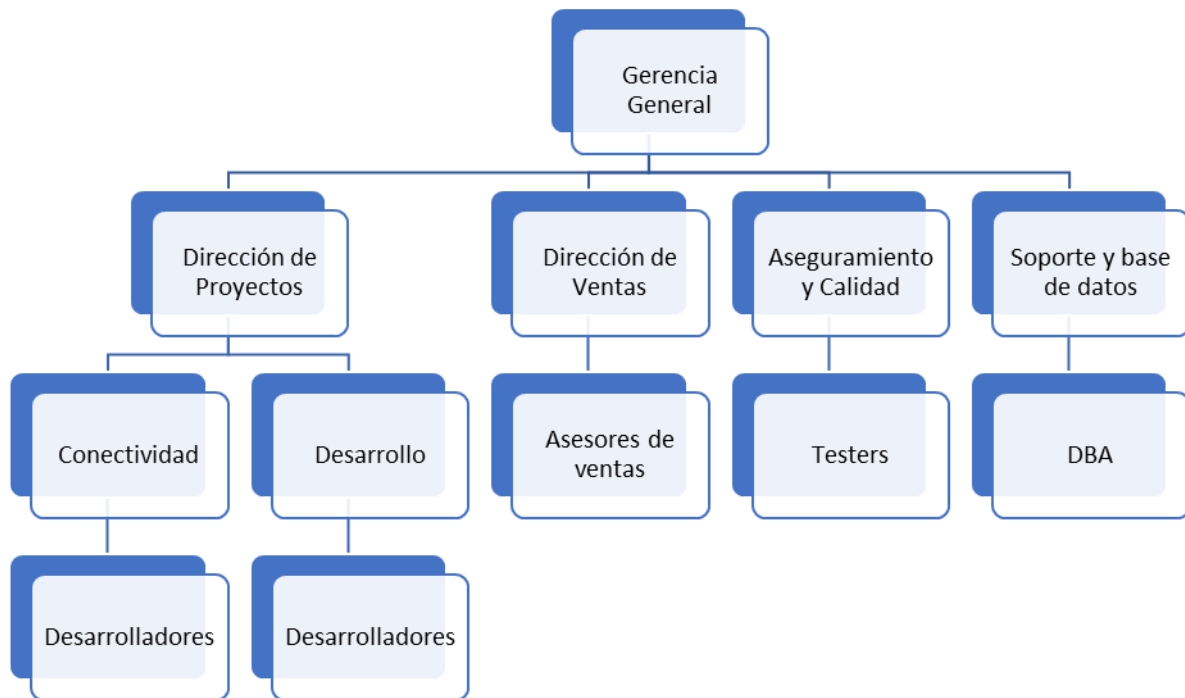


Figura 1. Estructura organizacional GBSYS.SA.

Fuente: Elaboración propia

#### **2.1.4 Productos que ofrece**

GBSYS.SA ofrece los siguientes productos:

- Administración de la infraestructura tecnológica: Ofrece análisis y monitoreo de equipos de redes y bases de datos principalmente de productos Oracle.
- Desarrollo de aplicaciones informáticas: Ofrece productos de software a la medida, migración de sistemas y soluciones relacionadas a la conectividad de sistemas.
- Tercerización: Ofrece servicios de tercerización de profesionales capacitados en el área de administración de sistemas, diseño de aplicaciones y programación.
- Consultoría en tecnología de información: Ofrece servicios de planeación, diseño e implementación de sistemas web.

## **2.2 Teoría de Administración de Proyectos**

### **2.2.1 Proyecto**

En el área de administración de proyectos, PMI (2013) define al proyecto como “un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único”. (p.3). Por otra parte, (Gray y Larson) afirman

“Un proyecto es un esfuerzo complejo, no rutinario, limitado por el tiempo, el presupuesto, los recursos y las especificaciones de desempeño y que se diseña para cumplir las necesidades del cliente” (p.5)

El concepto de proyecto se refiere a un planeamiento de tareas, actividades para cumplir un propósito. La ejecución del plan de un proyecto genera la obtención de un resultado el cual puede ser un producto o servicio.

Dependiendo del objetivo que se desea alcanzar, el proyecto puede variar su complejidad y características. No existe una formula exacta para gestionarlos solo

guías de buenas prácticas, experiencia y factores ambientales relacionados pueden ayudar a realizar una mejor gestión de ellos.

### **2.2.2 Administración de Proyectos**

La gestión de proyectos es el arte de la planeación de un proyecto para una ejecución eficiente, PMI (2013) afirma que “la dirección de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades del proyecto para cumplir con los requisitos del mismo” (p.5).

El proyecto de software tiene un conjunto de características que puede dificultar su gestión, en la extensión para software el PMI (2013) indica las siguientes características de un proyecto de software:

- El software es un producto intangible y maleable. El desarrollo de software se caracteriza por ser un proceso de aprendizaje donde se obtiene conocimiento en el transcurso de todo el proyecto.
- Los elementos claves que hacen que los proyectos de software sean desafiantes tiene que ver tanto con el producto y proyecto, la asignación no lineal de los recursos en el transcurso de proyecto y la incertidumbre inicial. Esto va relacionado a que depende de conocimiento que va adquiriendo los recursos en el desarrollo de proyecto y la dificultad que existe en medir el desempeño.
- Es muy probable que los requisitos cambien en el transcurso del proyecto a medida que se adquiere conocimiento del mismo.
- El software es un capital intelectual ya que es el principal activo de las organizaciones que desarrollan estos productos.
- La comunicación entre equipos que interactúan en proyecto a menudo es difícil por los procesos de globalización existe una gran parte de involucrados del proyecto que trabajan e interactúan virtualmente.

- Uno de los factores con más dificultad en los proyectos de software es la planeación inicial y la estimación de las actividades ya que van ligados a la eficiencia y efectividad de los programadores.
- El proceso de prueba de software debe planearse cuidadosamente para abarcar todas las rutas lógicas y entornos que necesitan ser probados, en muchas ocasiones se torna exhaustivo y no práctico. Sumado a esto es que la cuantificación o medida de la calidad es complicada debido a la naturaleza intangible del producto.
- Las herramientas y procesos de software están constantemente actualizándose.

### **2.2.3 Ciclo de vida de un proyecto**

El ciclo de vida de un proyecto “es la serie de fases por las que atraviesa un proyecto desde su inicio hasta su cierre”. PMI (2013, p. 38). Sobre la función de ciclo de vida, el mismo autor indica que “proporciona el marco de referencia básico para dirigir el proyecto, independiente del trabajo específico que se realiza”. PMI (2013, p. 38).

Los ciclos de vida de un proyecto varían dependiendo de la característica del mismo, no es igual el ciclo de vida de un proyecto de construcción que para un proyecto de software, según Gray, & Larson (2009)

“Por lo general, el ciclo de vida del proyecto atraviesa, en forma secuencial, cuatro etapas: definición, planeación, ejecución y entrega. El punto de partida se inicia en el momento en que arranca el proyecto. Los esfuerzos comienzan poco a poco, pero llegan a un punto máximo y luego caen hasta la entrega del proyecto al cliente” (p.7).

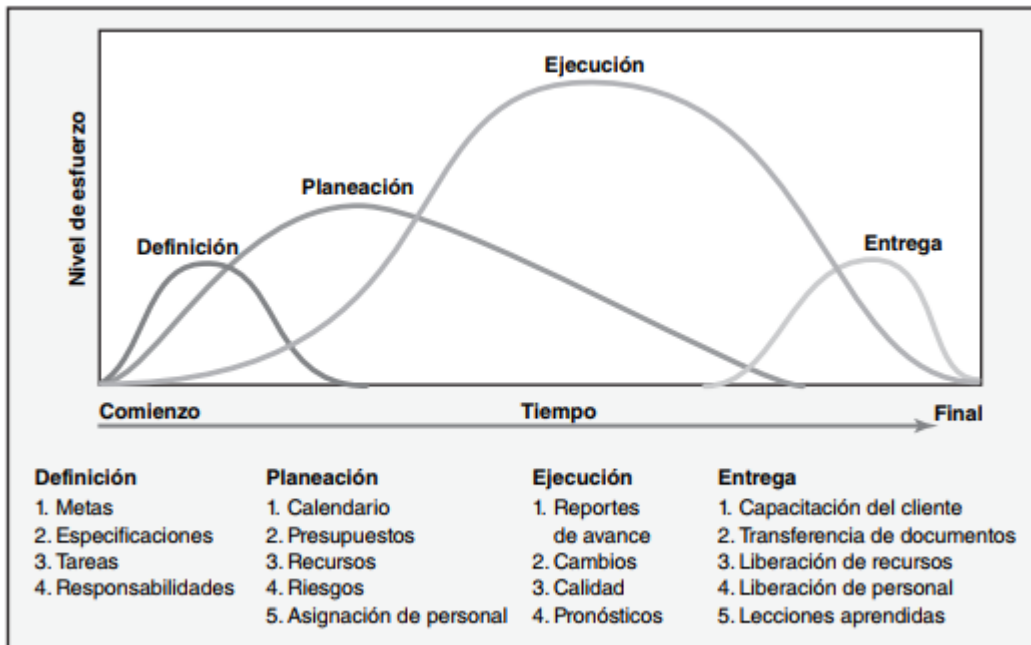


Figura 2. Ciclo de vida de Software

Fuente: Administración de proyectos. Gray & Larson, 2009

### 2.2.3.1 Ciclo de vida de un producto de Software

Los proyectos de software tienen la particularidad que puede desarrollarse en diferentes modelos de ciclo de vida, mayormente conocidos como ciclo de vida de software. El PMI (2013), en su extensión indica lo siguiente:

“Los ciclos de vida del proyecto de software y los ciclos de vida del producto de software son conceptos distintos. El ciclo de vida de un producto de software incluye un ciclo de vida inicial del proyecto de software, pero también incluye los procesos de implementación, soporte, mantenimiento, evolución, reemplazo y retiro de un producto de software. La mejora y la adaptación del software entregado inicialmente pueden implicar varios ciclos de vida del proyecto más allá de la inicial.”

Los enfoques de los ciclos de vida pueden cambiar entre ciclos predictivos y adaptativos. De acuerdo a la Extensión de Software de PMI (2013, p.26) clasifica

los ciclos de vida para los proyectos de software según el manejo de los requerimientos, costos, riesgos y participación de los involucrados como altamente predictivos, predictivos, adaptativos y altamente adaptativos.

• **Predictivo      Adaptativo**

Altamente Predictivo	Predictivo      Adaptativo	Altamente Adaptativo
<ul style="list-style-type: none"> <li>• Requerimientos son especificados durante el inicio y planificación.</li> <li>• El riesgo y el costo son controlados por planificación detallada basada en un análisis profundo de los requerimientos y limitaciones previo al desarrollo.</li> <li>• Los principales interesados se involucran en hitos calendarizados.</li> </ul>	<ul style="list-style-type: none"> <li>• Requerimientos son elaborados en intervalos periodicos durante el desarrollo del software.</li> <li>• El riesgo y el costo son controlados por una planificación progresiva basada en la especificación oportuna de requerimientos y limitaciones durante el desarrollo.</li> <li>• Los principales interesados se involucran en intervalos específicos.</li> </ul>	<ul style="list-style-type: none"> <li>• Requerimientos son elaborados en intervalos frecuentes durante el desarrollo del software.</li> <li>• El riesgo y el costo son controlados conforme surgen los requerimientos y limitaciones durante el desarrollo.</li> <li>• Los principales interesados se involucran continuamente.</li> </ul>

Figura 3. Ciclos de vida de proyectos de software y sus características

Fuente. Extensión de Software de la Guía de PMbok, PMI 2013.

### 2.2.3.2 Características de los ciclos de vida de Software

La creación de un producto de software requiere una variedad de procesos en el ciclo de vida. Según ISO/IEC/IEEE Standard 12207(citado por PMI,2013), el desarrollo de software tiene los siguientes procesos.

- Análisis: Proceso análisis de requerimientos de Software.
- Arquitectura: Proceso de diseño de la arquitectura de Software.
- Diseño: Proceso de diseño detallado de Software.
- Construcción: Proceso de construcción del Software.
- Integración: Proceso de Integración de Software.
- Pruebas: Proceso de Calidad y Pruebas de Software.



### 2.2.3.3 Fases del proyecto

Las fases de un proyecto es un conjunto de actividades que culmina con la finalización de un hito o varios hitos. (PMI,2013). Dependiendo de las necesidades de la organización o de las características de los proyectos, los equipos de trabajo pueden dividir el proyecto en uno o más fases, estas adaptaciones tienen impacto en los ciclos de vida de producto.

### 2.2.4 Procesos en la Administración de Proyectos

Para alcanzar un producto, resultado o servicio predefinido hay que realizar un conjunto de acciones y actividades relacionadas entre sí, esto es lo que llamamos un proceso y se caracteriza por sus entradas, herramientas y técnicas que se pueden aplicar y las salidas que se obtienen. PMI (2013). A continuación, se explica lo procesos de la administración de proyectos según el PMbok.

*Cuadro 1 Procesos de la administración de proyecto.*

Proceso	Descripción
<b>Grupo de Procesos de Inicio</b>	Procesos realizados para definir un nuevo proyecto o nueva fase de un proyecto existente al obtener la autorización para iniciar el proyecto o fase.
<b>Grupo de Procesos de Planificación</b>	Aquellos procesos requeridos para establecer el alcance del proyecto, refinar los objetivos y definir el curso de acción requerido para alcanzar los objetivos propuestos del proyecto.
<b>Grupo de Procesos de Ejecución</b>	Aquellos procesos realizados para completar el trabajo definido en el plan para la dirección del proyecto a fin de satisfacer las especificaciones del

Proceso	Descripción
	mismo.
<b>Grupo de Procesos de Monitoreo y Control</b>	Aquellos procesos requeridos para rastrear, revisar y regular el progreso y el desempeño del proyecto, para identificar áreas en las que el plan requiera cambios y para iniciar los cambios correspondientes.
<b>Grupo de Procesos de Cierre</b>	Aquellos procesos realizados para finalizar todas las actividades a través de todos los Grupos de Procesos, con el fin de cerrar formalmente el proyecto o una fase del mismo.

Fuente Guía de PMbok, PMI (2013)

### 2.2.5 Áreas del Conocimiento de la Administración de Proyectos

Un área de conocimiento representa un conjunto de conceptos, términos y actividades que constituyen un campo profesional, campo de administración de proyecto o área de especialización”. PMI (2013)

Los equipos de proyecto deben hacer uso de las áreas de la manera que consideren más adecuada respecto al proyecto específico que se vaya a implementar. A continuación, se describen las 10 áreas del conocimiento:

- **Gestión de la integración:** Incluye los procesos y actividades necesarios para identificar, definir, combinar, unificar y coordinar los procesos y actividades de administración de proyectos dentro de los grupos de procesos de gestión de proyectos. PMI (2013, p.63).
- **Gestión del alcance:** Incluye los procesos para garantizar que el proyecto incluya todo el trabajo requerido para completar el proyecto exitosamente. Gestionar el alcance del proyecto se enfoca primordialmente en definir y

controlar qué se incluye y qué no se incluye en el proyecto. PMI (2013, p.105).

- **Gestión del tiempo:** Incluye los procesos necesarios para gestionar la finalización dentro del plazo del proyecto. PMI (2013, p. 141).
- **Gestión del costo:** Incluye los procesos involucrados en el manejo de presupuesto, financiamiento y control de costos de manera que el proyecto se complete con el presupuesto aprobado. PMI (2013, p. 193).
- **Gestión de la calidad:** Incluye los procesos y actividades de la organización ejecutora que determina las políticas de calidad, sus objetivos y responsabilidades de forma que el proyecto satisfaga las necesidades para las cuales fue planeado PMI (2013, p.227).
- **Gestión de los recursos humanos:** Incluye los procesos que organizan, gestionan y conducen al equipo del proyecto. El equipo del proyecto está compuesto por las personas a las que se han asignado roles y responsabilidades para completar el proyecto. PMI (2013, p. 255).
- **Gestión de las comunicaciones:** Incluye los procesos que son requeridos para garantizar la planificación, recolección, distribución, administración, almacenamiento, control y monitoreo de la información para que comunicación sea la adecuada dentro del proyecto. PMI (2013, p.287).
- **Gestión del riesgo:** Incluye los procesos para conducir la gestión de la planificación, identificación, análisis, respuesta y control de los riesgos del proyecto. PMI (2013, p.309).
- **Gestión de las adquisiciones:** Incluye los procesos para adquirir productos, servicios o resultados que se requieran fuera del equipo de proyecto. PMI (2013, p.355).
- **Gestión de los interesados:** Incluye los procesos requeridos para identificar las personas, grupos u organizaciones que pueden afectar o ser afectados por el proyecto, analizar sus expectativas e impacto en el proyecto. PMI (2013, p.391).

La figura 4 indica las áreas de conocimiento según PMI y los grupos de procesos involucrados en cada una de ellas.

Áreas de Conocimiento	Grupos de Procesos de la Dirección de Proyectos				
	Grupo de Procesos de Inicio	Grupo de Procesos de Planificación	Grupo de Procesos de Ejecución	Grupo de Procesos de Monitoreo y Control	Grupo de Procesos de Cierre
<b>4. Gestión de la Integración del Proyecto</b>	4.1 Desarrollar el Acta de Constitución del Proyecto	4.2 Desarrollar el Plan para la Dirección del Proyecto	4.3 Dirigir y Gestionar el Trabajo del Proyecto	4.4 Monitorear y Controlar el Trabajo del Proyecto 4.5 Realizar el Control Integrado de Cambios	4.6 Cerrar Proyecto o Fase
<b>5. Gestión del Alcance del Proyecto</b>		5.1 Planificar la Gestión del Alcance 5.2 Recopilar Requisitos 5.3 Definir el Alcance 5.4 Crear la EDI/WBS		5.5 Validar el Alcance 5.6 Controlar el Alcance	
<b>6. Gestión del Tiempo del Proyecto</b>		6.1 Planificar la Gestión del Cronograma 6.2 Definir las Actividades 6.3 Secuenciar las Actividades 6.4 Estimar los Recursos de las Actividades 6.5 Estimar la Duración de las Actividades 6.6 Desarrollar el Cronograma		6.7 Controlar el Cronograma	
<b>7. Gestión de los Costes del Proyecto</b>		7.1 Planificar la Gestión de los Costos 7.2 Estimar los Costos 7.3 Determinar el Presupuesto		7.4 Controlar los Costos	
<b>8. Gestión de la Calidad del Proyecto</b>		8.1 Planificar la Gestión de la Calidad	8.2 Realizar el Aseguramiento de Calidad	8.3 Controlar la Calidad	
<b>9. Gestión de los Recursos Humanos del Proyecto</b>		9.1 Planificar la Gestión de los Recursos Humanos	9.2 Adquirir el Equipo del Proyecto 9.3 Desarrollar el Equipo del Proyecto 9.4 Dirigir el Equipo del Proyecto		
<b>10. Gestión de las Comunicaciones del Proyecto</b>		10.1 Planificar la Gestión de las Comunicaciones	10.2 Gestionar las Comunicaciones	10.3 Controlar las Comunicaciones	
<b>11. Gestión de los Riesgos del Proyecto</b>		11.1 Planificar la Gestión de los Riesgos 11.2 Identificar los Riesgos 11.3 Realizar el Análisis Cualitativo de Riesgos 11.4 Realizar el Análisis Cuantitativo de Riesgos 11.5 Planificar la Respuesta a los Riesgos		11.6 Controlar los Riesgos	
<b>12. Gestión de las Adquisiciones del Proyecto</b>		12.1 Planificar la Gestión de las Adquisiciones	12.2 Efectuar las Adquisiciones	12.3 Controlar las Adquisiciones	12.4 Cerrar las Adquisiciones
<b>13. Gestión de los Interesados del Proyecto</b>	13.1 Identificar a los Interesados	13.2 Planificar la Gestión de los Interesados	13.3 Gestionar la Participación de los Interesados	13.4 Controlar la Participación de los Interesados	

Figura 4. Correspondencia entre grupos de procesos y áreas de conocimiento de la dirección de proyectos de la Guía del PMbok.

Fuente: Guía PMbok, PMI 2013

El objetivo de esta propuesta es realizar un conjunto de prácticas, técnicas, procedimientos y normas usadas para quienes trabajan en el departamento de control y aseguramiento de la calidad software. En un proyecto de desarrollo de software, el proceso de gestión de calidad involucra actividades que se realizan en paralelo con los procesos de ciclo de vida de software y existe una particularidad que dichas actividades son independientes. Es decir, el proceso de calidad podría ser visto como un pequeño proyecto independiente que conlleva subproceso de planeación, diseño, ejecución, seguimiento y cierre. Es por ello que el tema principal de este documento es limitándose en la gestión de la calidad con el fin de profundizar en el área debido a las características que posee un proyecto de desarrollo de software.

### **2.3 Gestión de la calidad**

Los marcos de trabajo para el control y aseguramiento de la calidad de software involucran una serie de conceptos que han ganado popularidad transcurso de los años. Al ser el producto de software un resultado no tangible y asociado a la creatividad de los programadores tiende a presentar altas probabilidades de error humano, por ello al involucrar procesos de aseguramiento y control de calidad con el conjunto de procesos de desarrollo disminuye el riesgo de los proyectos.

De acuerdo al PMbok la gestión de calidad incluye los siguientes procesos:

- **Planificar la Gestión de la Calidad:** Es el proceso de identificar los requisitos y/o estándares de calidad para el proyecto y sus entregables, así como de documentar cómo el proyecto demostrará el cumplimiento con los mismos. PMI (2013, p.217)
- **Realizar el Aseguramiento de Calidad:** Es el proceso que consiste en auditar los requisitos de calidad y los resultados de las mediciones de control de calidad, para asegurar que se utilicen las normas de calidad y las definiciones operacionales adecuadas. PMI (2013, p.217)

- **Controlar la Calidad:** Es el proceso por el que se monitorea y se registran los resultados de la ejecución de las actividades de control de calidad, a fin de evaluar el desempeño y recomendar los cambios necesarios. PMI (2013, p.217)

Cabe destacar que dentro de la calidad de software existen actividades tanto de control y como para aseguramiento. Según la extensión para software de la guía PMbok de PMI (2013) indica que el “aseguramiento de la calidad de software (SQA por sus siglas en inglés) es un proceso continuo que audita otros procesos de software para garantizar que los mismos está siendo seguidos. SQA también determina el grado en que se obtienen los resultados deseados del control de la calidad de software”. Por otro lado, el Control de Calidad de Software (SQC) según PMI (2013) se refiere “a la aplicación de métodos, herramientas y técnicas para garantizar que los productos de software (incluyendo, pero no limitado al código de software) satisfagan los requisitos de calidad para un producto de software en desarrollo o modificación”. Es decir, mientras el aseguramiento se enfoca a verificación de los procesos para así prevenir defectos y mejorar el rendimiento, el control se dirige a la validación de producto final, encontrar defectos y corregirlos.

### **2.3.1 Calidad y atributos de calidad de Software**

El concepto de calidad según PMI (2013) es “el grado en que un conjunto de características inherentes cumple con los requisitos”. La definición de IEEE (2010) sobre calidad es similar y define como “el grado en que un producto de software satisface las necesidades indicadas e implícitas cuando se usa bajo condiciones específica”. De acuerdo con la extensión de software del PMbok de PMI (2013), la complejidad de la calidad de software ha dado lugar a varios modelos y los mismo pueden incrementar la definición e incluir calidad del proceso, producto, uso, datos y código. Hace varios años atrás, probablemente el concepto de calidad estaba limitada a satisfacer las necesidades del cliente, pero a medida que aumenta la

competitividad del mercado y los sistemas se hacen más complejos, empezaron a surgir otras necesidades y las mismas son el punto de partida para la definición de una serie de atributos de calidad que no se limita únicamente a cumplir los requisitos funcionales.

En la extensión para software de la guía PMbok de PMI (2013) indica que “los atributos de calidad para un producto de software en particular pueden incluir elementos de una lista muy larga, que incluye atributos que van desde la accesibilidad, adaptabilidad, analizabilidad, disponibilidad, compatibilidad y complejidad, a la capacidad de supervivencia, la capacidad de prueba, la comprensibilidad y la facilidad de uso”.

Según el Estándar Internacional ISO 9126 la calidad de software se clasifica en los siguientes atributos:

- Funcionalidad
- Fiabilidad
- Usabilidad
- Eficiencia
- Mantenibilidad
- Portabilidad
- Calidad de uso

### **2.3.2 Proceso de pruebas de software**

De acuerdo al Internacional Software Testing Qualifications Board (2011) “las pruebas pueden medir la calidad de un software en términos de los defectos detectados por lo que respecta a los requisitos y características funcionales y no funcionales (tales como fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad)”. El proceso de pruebas incluye subprocesos no solo de control de calidad sino también de aseguramiento, debido a que está en constante evolución, aprende de la experiencia de otros proyectos y genera insumos para la mejora continua no solo del proceso de pruebas sino también para las actividades de desarrollo.

### 2.3.2.1 Actividades del proceso de pruebas

De acuerdo al Internacional Software Testing Qualifications Board (2011), “la parte más visible del proceso de pruebas es la ejecución. No obstante, para ser efectivos y eficientes, el plan de pruebas también debe indicar el tiempo necesario para planificar las pruebas, diseñar los casos de prueba, preparar la ejecución y evaluar resultandos.” A continuación, se detalla las actividades del proceso de pruebas:

*Cuadro 2 Actividades del proceso de pruebas*

Actividades	Tareas
Planificación de pruebas y control	<ul style="list-style-type: none"> <li>• Definir los objetivos de la prueba y la especificación de las actividades</li> <li>• Comparar constantemente el progreso real con el plan previsto (Control de pruebas)</li> <li>• Informar sobre el estado de las pruebas</li> </ul>
Análisis y diseño de pruebas	<ul style="list-style-type: none"> <li>• Revisar la base de pruebas (requerimientos, nivel de integridad de software, informes de análisis de riesgos, arquitectura, diseño y especificaciones de interfaz)</li> <li>• Evaluar la testabilidad de la base de pruebas y de los objetos de prueba</li> <li>• Identificar y priorizar las condiciones de prueba.</li> <li>• Diseñar y priorizar los casos de</li> </ul>



Actividades	Tareas
	<p>pruebas de alto nivel.</p> <ul style="list-style-type: none"> <li>• Identificar los datos de pruebas necesarios para soportar las condiciones de prueba y los casos de prueba.</li> <li>• Diseñar la configuración del entorno de pruebas e identificar cualquier estructura y herramientas necesarias.</li> <li>• Crear una trazabilidad bidireccional entre la base de pruebas y los casos de prueba.</li> </ul>
Implementación y ejecución de pruebas	<ul style="list-style-type: none"> <li>• Finalizar, implementar y priorizar los casos de prueba.</li> <li>• Desarrollar y priorizar los procedimientos de prueba, crear datos de prueba, preparar arneses de prueba y redactar los scripts de las pruebas automatizadas.</li> <li>• Crear los juegos de prueba a partir de procedimientos de prueba para lograr una ejecución de pruebas eficiente.</li> <li>• Verificar que el entorno de pruebas ha sido correctamente configurado.</li> <li>• Verificar y actualizar una trazabilidad entre la base de</li> </ul>

Actividades	Tareas
	<p>pruebas y los casos de pruebas.</p> <ul style="list-style-type: none"> <li>• Ejecutar los procedimientos de prueba manualmente o recurriendo a herramientas de ejecución.</li> <li>• Registrar los incidentes.</li> <li>• Repetir las actividades de pruebas como resultado de una medida adoptada para cada incidente.</li> </ul>
Evaluación de criterios de los criterios de salidas e informes	<ul style="list-style-type: none"> <li>• Comprobar los registros de pruebas con los criterios de salida previstos en la planificación de pruebas.</li> <li>• Elaborar un resumen de las pruebas para las partes interesadas.</li> </ul>
Actividades de cierre	<ul style="list-style-type: none"> <li>• Comprobar cuáles de los productos entregables previstos ha sido efectivamente entregados.</li> <li>• Cerrar los informes de incidencias o aportar modificaciones a aquellos que sigan abiertos.</li> <li>• Documentar la aceptación del sistema.</li> <li>• Finalizar y archivar los productos</li> </ul>

Actividades	Tareas
	de soporte de pruebas. <ul style="list-style-type: none"> <li>• Analizar las acciones aprendidas.</li> <li>• Utilizar la información recompilada para mejorar la madurez de las pruebas.</li> </ul>

Fuente: Software Testing Qualifications (2011).

### 2.3.3 Modelo en V (Modelo de desarrollo secuencia).

El modelo en V es similar al modelo de cascada, pero con la variante que se integra al modelo de desarrollo un paralelo de actividades de pruebas.

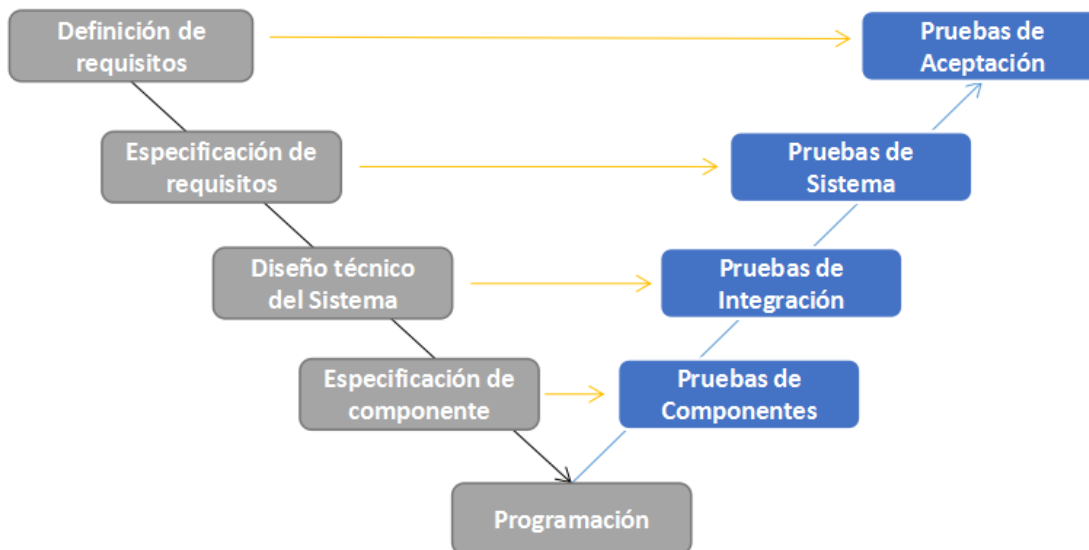


Figura 5. Modelo V

Fuente: Advanced Software Testing Vol.3. Guide to the ISTQB Advance Certification as an Advance Technical Test Analyst, Black&Michell 2008

En la figura 5 se puede visualizar que cada uno de las fases de desarrollo de un producto de software lleva un paralelo con una actividad de pruebas, es decir,

desde que inicia la definición del producto de software al inicio de un proyecto también inicia el proceso de pruebas.

### 2.3.4 Niveles de prueba

En el modelo en V cada nivel de desarrollo tiene un correspondiente nivel de pruebas, de acuerdo al Software Testing Qualifications (2011), cada nivel de prueba cumple con un enfoque diferente dentro del proceso de desarrollo de software, por ende, cada nivel posee base de pruebas (productos de trabajo a que se hace referencia para derivar los casos de prueba), objetos de pruebas (que se está probando) y alcance.

*Cuadro 3 Niveles de pruebas*

Nivel	Descripción	Base de Prueba	Objetos de Prueba
Prueba de componentes	Las pruebas de componentes (también conocidas como pruebas de unidad, módulo o programa) tiene por objeto localizar defectos en y comprobar el funcionamiento de módulos de software, objetos, clases, etc. Que puede probarse por separado.	Requisitos de componentes Diseño de detalle Código	Componentes Programas Conversión de datos / programas de migración.
Pruebas de	Las pruebas de	Diseño de	Implementación de

Nivel	Descripción	Base de Prueba	Objetos de Prueba
Integración.	integración se ocupan de probar las interfaces entre componentes, las interacciones con distintas partes de un mismo sistema.	software y sistema Arquitectura Flujos de trabajo Casos de uso	base de datos de subsistemas Infraestructura Interfaces
Pruebas de Sistema	Las pruebas de sistema se refieren al comportamiento de todo un sistema/producto. El alcance de las pruebas debe estar claramente indicado en el plan maestro de pruebas.	Especificación de requisitos de software Casos de uso Especificaciones funcionales Informes de análisis de riesgos	Manuales de sistema, usuario y funcionamiento Configuración del sistema
Pruebas de aceptación	Realizadas por el usuario final, el objetivo de estas pruebas es crear confianza en el sistema, partes del sistema o características específicas no funcionales.	Requisitos de usuario Requisitos de sistema Casos de uso Proceso de negocio Informes de análisis de riesgos	Procesos de negocio en el sistema completamente integrado. Procedimiento de usuario. Formularios. Informes.

Fuente: Software Testing Qualifications (2011).

### 2.3.5 Tipos de pruebas

Para emerger dentro del área de la calidad de software hay que conocer sobre los tipos de prueba que se pueden realizar en cada uno de los niveles de pruebas, de acuerdo a Software Testing Qualifications (2011), cada tipo de prueba cumple un objetivo específico (función a realizar por el software, una característica de calidad no funciona, la estructura o arquitectura de software o sistema, confirmar si se han solucionado defectos reportados o localizar cambios no intencionados).

*Cuadro 4 Tipos de pruebas*

Tipos de Pruebas	Objetivo
Pruebas funcionales	La prueba funcional se basa en las funciones y prestaciones y en su interoperabilidad con los sistemas específicos. Puede llevarse a cabo en cualquier nivel de pruebas.
Pruebas no funcionales	Las pruebas no funcionales, pero sin limitarse a ello, son pruebas de rendimiento, pruebas de carga, pruebas de estrés, pruebas de usabilidad, pruebas de mantenibilidad, pruebas de fiabilidad y pruebas de portabilidad. Pueden ejecutarse a todos los niveles de pruebas.
Pruebas de estructurales	Las técnicas estructurales (caja blanca) ayudan a medir la exhaustividad de las pruebas mediante la evaluación de la cobertura de un tipo de estructura. Pueden utilizarse a cualquier nivel de pruebas.

Pruebas de repetición y regresión	Las pruebas de regresión es la prueba reiterada de un programada ya probado, después de haber sido modificado, con vista a localizar defectos surgidos.
-----------------------------------	---

Fuente: Software Testing Qualifications (2011).

### 2.3.6 Casos de prueba

Los casos de prueba es una herramienta de revisión creado en el proceso de diseño de las pruebas, su objetivo es revisar un objeto de prueba. De acuerdo al estándar IEEE 829 la especificación de los casos de prueba es un documento que detalla los objetivos, entradas, acciones de prueba, resultados esperados y precondiciones de la ejecución para un elemento de prueba. La siguiente tabla explica cada una de las partes de un caso de prueba.

*Cuadro 5 Niveles de un caso de prueba*

Nivel	Descripción
Precondiciones	Describe el entorno de prueba necesario para la configuración de la prueba, la ejecución y la grabación de resultados. Esta sección es comúnmente documentada por escenario o grupo de escenarios.
Valores de entrada	Especifica cada entrada requerida para ejecutar cada caso de prueba. Puede ser valores o tablas de base de datos.
Resultados esperados	Especifica todas las salidas y el comportamiento esperado (por ejemplo, tiempo de respuesta) requerido para los elementos de prueba. Proporcionar el (los) valor (es) exacto (s) para cada

Nivel	Descripción
	salida requerida y comportamiento esperado.
Requerimientos especiales	Cualquier restricción especial en los procedimientos de prueba del nivel que ejecutan este caso de prueba como postcondiciones y / o procesamiento.
Dependencias	Identificadores de casos de prueba que se deben ejecutar antes de este caso de prueba. .
Identificador	Describe el identificador único que cada caso de prueba necesita para poder distinguirlo de todos los demás casos de prueba. Una herramienta automatizada puede controlar la generación de los identificadores.

Fuente: IEEE Standard for Software and System Test Documentation, IEEE (2008)

## 2.4 Metodología Ágiles

Las metodologías y procesos ágiles nacieron para solventar la necesidad de un alcance cambiante en el desarrollo de proyecto. Tradicionalmente con guías para administración de proyecto como la que ofrece PMbok, es necesario definir el alcance desde el primer grupo de procesos de inicio y el cambio de alcance desencadena una serie de procesos relacionados a control y monitoreo que conllevar una serie de documentación que dependiendo de las características de proyecto provoca retrasos. De ahí nacieron las metodologías ágiles como SCRUM siendo las más popular seguida por Extreme Programming.

La empresa patrocinadora de este proyecto desarrolla los proyectos siguiendo los principios de la guía de trabajo de Scrum. Los equipos de aseguramiento y control



calidad actúan como un equipo independiente al equipo de desarrollo de los proyectos y se acoplan a un marco de trabajo ágiles llamado Scrum.

### 2.4.1 Scrum

Scrum es un proceso para la administración y control que disminuye la complejidad para enfocarse en la construcción del producto adaptándose la necesidad del negocio. Scrum compuesto por equipos Scrum y sus roles y eventos, artefactos y normas asociadas. Cada componente dentro del marco de trabajo tiene un fin específico y es fundamental para el éxito de Scrum y para su uso. Scrum.Org (2009).

*Cuadro 6 Roles y eventos de Scrum.*

Rol / Evento	Función
Rol: Scrum Master	El Scrum Master es el líder del equipo de Scrum. El Scrum Master ayuda a aquellos ajenos al equipo a comprender cuáles de sus interacciones con el equipo son útiles y cuáles no. El Scrum Master ayuda a todos a cambiar estas interacciones para maximizar el valor creado por el equipo de Scrum.
Rol: Dueño del producto	El propietario del producto es responsable de maximizar el valor del producto resultante del trabajo del equipo de desarrollo. El propietario del producto es una persona, no un comité. El propietario del producto puede representar los deseos de un comité en la lista de pedidos del producto, pero aquellos que deseen cambiar la prioridad de un elemento de la cartera

Rol / Evento	Función
	de productos deben dirigirse al propietario del producto.
Rol: Equipo Scrum	El Equipo de desarrollo está formado por profesionales que realizan el trabajo de entregar un Incremento potencialmente liberable del producto "Hecho" al final de cada Sprint. Se requiere un incremento "Hecho" en la Revisión de Sprint. Solo los miembros del Equipo de Desarrollo crean el Incremento. Los equipos de desarrollo están estructurados y facultados por la organización para organizar y gestionar su propio trabajo. La sinergia resultante optimiza la eficiencia y eficacia general del equipo de desarrollo.
Evento: Sprint	El corazón de Scrum es un Sprint, un lapso de tiempo de un mes o menos durante el cual se crea un Incremento de producto el cual utilizable y potencialmente liberable.
Evento: Planeación del Sprint	El trabajo que se realizará en el Sprint está planificado en la planeación del Sprint. Este plan es creado por el trabajo colaborativo de todo el equipo de Scrum. La planificación de Sprint está sincronizada en el tiempo hasta un

Rol / Evento	Función
	máximo de ocho horas para un Sprint de un mes.
Evento: Reunión Diaria de Scrum	Una reunión diaria de Scrum es un evento de 15 minutos para el Equipo de Desarrollo. El DS se lleva a cabo todos los días del Sprint. En él, el equipo de desarrollo planea trabajar durante las próximas 24 horas. Esto optimiza la colaboración y el rendimiento del equipo inspeccionando el trabajo desde la última reunión y pronosticando el próximo trabajo de Sprint.
Evento: Revisión de Sprint	Una Revisión de Sprint se lleva a cabo al final del Sprint para inspeccionar el Incremento y adaptar el Retraso del Producto si es necesario. Durante la Revisión de Sprint, el Equipo de Scrum y las partes interesadas colaboran sobre lo que se hizo en el Sprint. En función de eso y de cualquier cambio en la acumulación de productos durante el Sprint, los asistentes colaboran en las siguientes cosas que podrían hacerse para optimizar el valor.
Evento: Retrospectiva de Scrum.	La Retrospectiva de Sprint es una oportunidad para que equipo Scrum se inspeccione a sí mismo y cree un plan para que se implementen mejoras durante el próximo Sprint. La

Rol / Evento	Función
	Retrospectiva Sprint ocurre después de la Revisión Sprint y antes de la próxima Planificación Sprint.

Fuente: Guía Scrum. Schwaber & Sutherland (2017)

### **3 MARCO METODOLOGICO.**

#### **3.1 Fuentes de información**

“Las fuentes de información tienen como objetivos principales buscar, localizar y difundir el origen de la información contenida en cualquier soporte físico, no exclusivamente en formato libro, aunque sus productos más elaborados y representativos sean los repertorios.” (Martín, 1995)

##### **3.1.1 Fuentes Primarias**

Las fuentes primarias son aquellas que proporcionan datos de primera mano. Es decir, es el propio investigador el que informa sobre los resultados de su investigación (Mari 2002). La información obtenida de este tipo de fuente es el insumo para la investigación.

En el proyecto se utilizarán fuentes primarias para investigar la situación actual de la empresa con relación al área de calidad, para obtener información se debe realizar entrevistas a los encargados de los departamentos para recompilar información histórica que brinde un panorama de los procesos que se realizan en el presente para evaluar que se debe cambiar, sustituir o mejorar.

##### **3.1.2 Fuentes Secundarias**

Se refiere a toda información que se encuentra disponible en el momento en que surge la necesidad de su utilización. Con mucha frecuencia los investigadores encuentran datos, informes, páginas web o metodologías que resultan útiles para sacar adelante sus proyectos. (Abascal & Grande, 2009, p.60). Para el proyecto se van utilizar las siguientes fuentes secundarias:

- Páginas de internet: Es la herramienta donde es simple obtener información de páginas confiables. Se va a consultar sobre procesos de calidad en los productos de software.
- Libros sobre administración de proyectos y calidad: La literatura debe ser consultada para analizar sobre la integración de los procesos de calidad con los demás procesos de administración de software. Las fuentes

secundarias que van a ser más utilizadas en este proyecto son la extensión de PMbok para proyectos de software y el manual para la certificación de calidad de software ISTQB.

*Cuadro 7. Fuentes de Información Utilizadas*

Objetivos	Fuentes de información	
	Primarias	Secundarias
1. Determinar la situación actual de la empresa en la gestión de la calidad en la empresa.	<ul style="list-style-type: none"> <li>Entrevista a los integrantes de equipo de QA para obtener información histórica sobre procesos de calidad.</li> </ul>	<ul style="list-style-type: none"> <li>Extensión del PMbok para proyectos de software</li> </ul>
2. Realizar un análisis de la situación actual de la empresa en gestión de calidad de software respecto a los procesos y técnicas de calidad descritas en la norma para software IEEE	<ul style="list-style-type: none"> <li>Activos de la empresa, datos históricos.</li> </ul>	<ul style="list-style-type: none"> <li>Extensión del PMbok para proyectos de software</li> <li>Silabo de ISTQB CTFL, IEEE 829, ISO/IEC 9126</li> <li>Información de procesos de calidad obtenidos de otras fuentes por internet.</li> </ul>

Objetivos	Fuentes de información	
	Primarias	Secundarias
829, el estándar ISO/IEC 9126 y el Silabo Foundation Level de ISTQB.		
3. Determinar las oportunidades de mejora en la gestión de la calidad de software tomando como fuente el análisis realizado.	<ul style="list-style-type: none"> <li>• Criterio Experto del director de proyectos</li> </ul>	<ul style="list-style-type: none"> <li>• Extensión del PMbok para proyectos de software</li> <li>• Silabo de ISTQB CTFL, IEEE 829, ISO/IEC 9126</li> <li>• Información de procesos de calidad obtenidos de otras fuentes por internet.</li> </ul>
4. Diseñar una guía de de los procesos y técnicas para la gestión de la calidad de software que se ajusten a las necesidades	<ul style="list-style-type: none"> <li>• Criterio Experto del director de proyectos de la empresa.</li> <li>• Activos de la organización tales como plantillas de procedimientos de gestión de</li> </ul>	<ul style="list-style-type: none"> <li>• Extensión del PMbok para proyectos de software</li> <li>• Silabo de ISTQB CTFL, IEEE 829, ISO/IEC 9126</li> </ul>

Objetivos	Fuentes de información	
	Primarias	Secundarias
de empresa considerando el análisis elaborado.	proyectos.	
5. Crear un plan de capacitación para los involucrados para transmitir en conocimiento sobre la propuesta de metodología en gestión de calidad de software y la apliquen en sus proyectos.	<ul style="list-style-type: none"> <li>• Reunión con el gerente de proyectos</li> <li>• Activos de la empresa para obtener plantillas de presentaciones y guía de capacitaciones.</li> </ul>	<ul style="list-style-type: none"> <li>• Silabo de ISTQB CTFL, IEEE 829, ISO/IEC 9126</li> <li>• Extensión del PMbok para proyectos de software</li> <li>• Información de procesos de calidad obtenidos de otras fuentes por internet</li> </ul>

Fuente: Elaboración propia.



### 3.2 Métodos de Investigación

Según (Eyssautier, 2006) los métodos de investigación se definen “como un procedimiento o conjunto de procedimientos que sirven de instrumento para alcanzar los fines de la investigación, un procedimiento general basado en principios lógicos que pueden ser comunes a varias ciencias”

#### 3.2.1 Método Observación

De acuerdo a Eyssautier (2006) la observación “es la acción de mirar detenidamente una cosa para asimilar en detalle la naturaleza investigada, su conjunto de datos, hechos y fenómenos”. En cuadro 8, se muestra los tipos de observaciones según el mismo autor.

*Cuadro 8. Tipos de Observaciones*

Tipo	Descripción
Observación directa	La observación directa consiste en interrelaciones de manera directa con el medio y con la gente que lo forman para realizar los estudios de campo.
Observación indirecta	La observación indirecta consiste en tomar nota de un hecho que sucede ante los ojos de un observador entrenado, midiendo el comportamiento externo del individuo en sociedad.
Observación por entrevista	La observación por entrevista es un intercambio conversacional en forma oral, entre dos personas, con la finalidad de obtener información, datos o hechos. El método de entrevista puede ser informal, estructurado o no estructurado.

Fuente: Metodología de la Investigación. Desarrollo de la inteligencia. Eyssautier, 2006

### 3.2.2 Método Analíticosintético

“El método analítico sintético descompone una unidad en sus elementos más simples, examina cada uno de ellos por separado, volviendo a agrupar las partes para considerarlas en conjunto” (Eyssautier, 2006, p. 98). En la investigación de inicial de este proyecto es necesario analizar cada componente sujeto de estudio por separado como las buenas prácticas de calidad para luego juntar los componentes y realizar una propuesta sobre lo investigado.

En el cuadro 9 se describe cómo se van a utilizar los métodos los de investigación en cada objetivo específico.

*Cuadro 9. Métodos de Investigación Utilizadas*

Objetivos	Métodos de investigación	
	Observación	Método Analíticosintético
1. Determinar la situación actual y deficiencias de la empresa GBSYS.SA en el área de gestión de la calidad.	El método será utilizado para observar el proceso para la gestión de proyectos y calidad actual de la empresa con la información obtenida de las entrevistas.	Se analiza información investigada de procesos sobre calidad de productos de software y se analiza la información histórica de la empresa en el tema de calidad.
2. Realizar la una comparación de la situación actual de empresa en gestión de calidad de software respecto a los	Mediante la reunión con los integrantes del equipo de calidad, se observar sobre procesos en los marcos de trabajo, estándares y normas	Se analiza para realizar una comparación entre las prácticas estudiadas y procesos realizados en la empresa con el objetivo de desarrollar los procesos que van a formar parte de

Objetivos	Métodos de investigación	
	Observación	Método Analítico sintético
procesos y técnicas de calidad descritas en la norma para software IEEE 829, el estándar ISO/IEC 9126 y en el Foundation Level Syllabus de ISTQB.	internacionales de calidad y se realiza una comparación con lo investigado.	la propuesta como un conjunto de procesos.
3. Determinar las oportunidades de mejora en la gestión de la calidad de software tomando como fuente el análisis realizado.	Resultado del análisis comparativo se identifica las debilidades, fortalezas.	Se identifica las oportunidades de mejora tomando como fuente las debilidades y fortalezas de los procesos de gestión de calidad realizados actualmente.
4. Diseñar una guía de los procesos y técnicas para la gestión de la calidad de software que se ajusten a las necesidades de la empresa considerando el	Se realiza una observación de las plantillas utilizadas actualmente por la empresa, para realizar cambios y crear nuevas.	Se analiza los criterios dados por el juicio experto de otros miembros encargados de proyecto de la empresa y el material de investigación para diseñar nuevas plantillas con las sugerencias propuestas.

Objetivos	Métodos de investigación	
	Observación	Método Analítico sintético
análisis elaborado		
5. Crear un plan de capacitación para los involucrados para transmitir en conocimiento sobre la propuesta de metodología en gestión de calidad de software y la apliquen en sus proyectos.	Por medio de reunión y observación de guías históricas de capacitaciones se realiza un plan de capacitación para los involucrados	Se analizan los temas y métodos de aprendizaje para impartir la metodología de la propuesta a los involucrados.

Fuente: Elaboración propia.

### 3.3 Herramientas.

De acuerdo a PMI (2013), “una herramienta es algo tangible, como una plantilla o programa de software, utilizado al realizar una actividad, para producir un producto o resultado” (p. 548).

Las herramientas facilitan y contribuyen a conseguir los objetivos de un proyecto. Dentro de la guía PMbok en la explicación de los apartados se detalla una serie de herramientas correspondiente a un grupo de procesos. Aunque explícitamente no vamos a utilizar estas herramientas para realizar un proceso de gestión, las misma pueden colaborar para cumplir los objetivos específicos de este proyecto, a continuación, se explica brevemente las herramientas que se van a utilizar en este proyecto.

### **3.3.1 Tormentas de ideas.**

Según PMI (2013) la tormenta de ideas es

“una técnica que se utiliza para generar y recopilar múltiples ideas relacionadas con los requisitos del proyecto y del producto. Si bien la tormenta de ideas en sí misma no incluye votaciones o establecimiento de prioridades, a menudo se emplea con otras técnicas grupales de creatividad que sí lo hacen” (p.115.)

### **3.3.2 Análisis FODA.**

Según PMI (2013), el análisis FODA es

“una técnica que examina el proyecto desde cada uno de los aspectos FODA (fortalezas, oportunidades, debilidades y amenazas) para aumentar el espectro de riesgos identificados, incluidos los riesgos generados internamente. La técnica comienza con la identificación de las fortalezas y debilidades de la organización, centrándose ya sea en el proyecto, en la organización o en el negocio en general”. (p.326).

### **3.3.3 Juicio de Expertos**

PMI (2013) afirma que

“Los expertos con la experiencia adecuada, adquirida en proyectos o áreas de negocio similares, pueden identificar los riesgos directamente. El director del proyecto debe identificar a dichos expertos e invitarlos a considerar todos los aspectos del proyecto, y a sugerir los posibles riesgos basándose en sus experiencias previas y en sus áreas de especialización”. (p.327).

### **3.3.4 Reuniones**

Según PMI (2013) las reuniones se utilizan

“para discutir y abordar los asuntos pertinentes del proyecto durante la dirección y gestión del trabajo del proyecto. Los asistentes a las reuniones pueden incluir al director del proyecto, al equipo del proyecto y a los interesados adecuados, involucrados o afectados por los asuntos tratados”. (p.84)

### 3.3.5 Entrevistas

Según PMI (2013) indica que una entrevista es “una manera formal o informal de obtener información de los interesados, a través de un diálogo directo con ellos. Se lleva a cabo habitualmente realizando preguntas, preparadas o espontáneas y registrando las respuestas.” (p.114).

### 3.3.6 Diagrama de Flujos

Según PMI (2013) indica que los diagramas de flujos “también denominados mapas de procesos, porque muestran la secuencia de pasos y las posibilidades de ramificaciones que existen en un proceso que transforma una o más entradas en una o más salidas.” (p.236).

### 3.3.7 Estudio Comparativos

Según PMI (2013) indica que “los estudios comparativos implican comparar prácticas reales o planificadas del proyecto con las de proyectos comparables para identificar las mejores prácticas, generar ideas de mejora y proporcionar una base para medir el desempeño.” (p.239).

*Cuadro 10. Herramientas Utilizadas*

Objetivos	Herramientas
1. Determinar la situación actual y deficiencias de la empresa GBSYS.SA en el área de gestión de la calidad.	<ul style="list-style-type: none"> <li>• Revisión de documentos</li> <li>• Investigación Bibliográfica</li> <li>• Entrevistas</li> <li>• Diagrama de flujo</li> </ul>
2. Realizar la una comparación de la situación actual de empresa en gestión de calidad de software respecto a los procesos y técnicas de calidad descritas en	<ul style="list-style-type: none"> <li>• Entrevistas</li> <li>• Estudio Comparativos</li> <li>• Investigación Bibliográfica</li> </ul>

Objetivos	Herramientas
la norma para software IEEE 829, el estándar ISO/IEC 9126 y en el Foundation Level Syllabus de ISTQB.	
3. Determinar las oportunidades de mejora en la gestión de la calidad de software tomando como fuente el análisis realizado.	<ul style="list-style-type: none"> <li>• Diagrama causaefecto</li> <li>• Análisis Foda</li> <li>• Investigación Bibliográfica</li> <li>• Entrevistas</li> </ul>
4. Diseñar una guía de los procesos y técnicas para la gestión de la calidad de software que se ajusten a las necesidades de la empresa considerando el análisis elaborado	<ul style="list-style-type: none"> <li>• Revisión de documentos por ejemplo plantillas</li> <li>• Investigación Bibliográfica</li> <li>• Juicio de Expertos</li> </ul>
5. Crear un plan de capacitación para los involucrados para transmitir en conocimiento sobre la propuesta de metodología en gestión de calidad de software y la apliquen en sus proyectos.	<ul style="list-style-type: none"> <li>• Revisión de documentos</li> <li>• Juicio de Expertos</li> <li>• Análisis y procesamiento de información</li> </ul>

Fuente elaboración propia

### 3.4 Supuestos y Restricciones.

Los supuestos son factores que ocurren dentro del proceso de planificación que se consideran verdaderos, reales o seguros sin pruebas ni demostraciones. Las

restricciones son limitaciones que puede presentarse en la ejecución de algún proceso (PMI, 2013).

*Cuadro 11. Supuestos y Restricciones*

<b>Objetivos</b>	<b>Supuestos</b>	<b>Restricciones</b>
<p>1. Determinar la situación actual y deficiencias de la empresa GBSYS.SA en el área de gestión de la calidad.</p>	<ul style="list-style-type: none"> <li>• Existe fuentes de información válidas y confiables para investigar sobre los procesos de calidad de software.</li> <li>• Existe interés de la gerencia de la empresa en realizar la investigación.</li> <li>• La información histórica es de fácil acceso y no restringida.</li> </ul>	<p>Existe poca disponibilidad de los integrantes de equipo de aseguramiento y control de calidad y del gerente de proyectos para realizar entrevistas.</p>
<p>2. Realizar la una comparación de la situación actual de empresa en gestión de calidad de software respecto a los procesos y técnicas de calidad descritas en la norma para software IEEE 829, el estándar ISO/IEC</p>	<ul style="list-style-type: none"> <li>• Existe disponibilidad del encargado de proyectos para revisar los avances de los procesos y dar su criterio y recomendaciones</li> </ul>	<ul style="list-style-type: none"> <li>• Los procesos y procedimientos a desarrollar deben integrarse a la metodología ya establecida para la gestión de proyectos de la empresa.</li> </ul>



Objetivos	Supuestos	Restricciones
9126 y en el Foundation Level Syllabus de ISTQB.		
3. Determinar las oportunidades de mejora en la gestión de la calidad de software tomando como fuente el análisis realizado.	<ul style="list-style-type: none"> <li>• Existe fuentes de información válidas y confiables para investigar sobre los procesos de calidad de software.</li> </ul>	<ul style="list-style-type: none"> <li>• Los procesos y procedimientos a desarrollar deben integrarse a la metodología ya establecida para la gestión de proyectos de la empresa.</li> </ul>
4. Diseñar una guía de de los procesos y técnicas para la gestión de la calidad de software que se ajusten a las necesidades de la empresa considerando el análisis elaborado	<ul style="list-style-type: none"> <li>• La información histórica y plantillas utilizadas actualmente es de fácil acceso y no restringida.</li> </ul>	<ul style="list-style-type: none"> <li>• Existe poca disponibilidad de los compañeros de la empresa que realiza funciones de dirección de proyectos para obtener opinión sobre el diseño de las plantillas.</li> </ul>
5. Crear un plan de capacitación para los involucrados para transmitir en conocimiento sobre la	<ul style="list-style-type: none"> <li>• Se cuenta con la autorización de gerencia para realizar la capacitación a los</li> </ul>	<ul style="list-style-type: none"> <li>• Los días para realizar la capacitación queda a disponibilidad de</li> </ul>

Objetivos	Supuestos	Restricciones
propuesta de metodología en gestión de calidad de software y la apliquen en sus proyectos.	involucrados directos. • Se cuenta con el presupuesto y herramientas para realizar la capacitación.	los interesados.

Fuente elaboración propia

### 3.5 Entregables.

Para este proyecto los entregables van relacionados al resultado tangible de los objetivos específicos. Según la definición del PMI (2013) un entregable es:

“Cualquier producto, resultado o capacidad de prestar un servicio único y verificable que debe producirse para terminar un proceso, una fase o un proyecto. Los entregables son componentes tangibles completados para alcanzar los objetivos del proyecto y pueden incluir elementos del plan para la dirección del proyecto.” (p. 84).

En el cuadro 12 se detallan los entregables de los objetivos específicos del proyecto.

Cuadro 12. Entregables

Objetivos	Entregables
1. Determinar la situación actual y deficiencias de la empresa GBSYS.SA en el área de gestión de la calidad.	Descripción de roles, procesos, herramientas utilizados por la empresa GBSYS.SA que colaboran a la gestión de calidad.  Análisis de las deficiencias actuales en el área de gestión de calidad.
2. Realizar la una comparación de la situación actual de empresa en	Cuadros comparativos de los procesos, técnicas de diseño, pruebas,

Objetivos	Entregables
gestión de calidad de software respecto a los procesos y técnicas de calidad descritas en la norma para software IEEE 829, el estándar ISO/IEC 9126 y en el Foundation Level Syllabus de ISTQB.	documentación y herramientas utilizadas por la empresa GBSYS y técnicas de calidad descritas en las normas, estándares y marcos metodológicos conocidos para calidad de software
3. Determinar las oportunidades de mejora en la gestión de la calidad de software tomando como fuente el análisis realizado.	Lista de necesidades y oportunidades de mejora en los procesos de gestión de calidad utilizados por la empresa
4. Diseñar una guía de de los procesos y técnicas para la gestión de la calidad de software que se ajusten a las necesidades de la empresa considerando el análisis elaborado	Propuesta de metodología para la gestión de calidad software para la empresa GBSYS.SA
5. Crear un plan de capacitación para los involucrados para transmitir en conocimiento sobre la propuesta de metodología en gestión de calidad de software y la apliquen en sus proyectos.	Plan de capacitación para los interesados para retroalimentar sobre la propuesta de la metodología de gestión de calidad.

Fuente: Elaboración propia

## **4 DESARROLLO**

Este capítulo se divide en varias secciones para presentar los resultados obtenidos de la investigación realizada. Los resultados se plantean en cinco secciones, en la primera sección se presenta un detalle de la situación actual de la empresa en gestión de calidad, la segunda sección es una comparación de la situación de la empresa en los procesos investigados y abarcados en el marco teórico de este documento. La tercera sección se presenta los puntos de mejora identificados del análisis de la situación actual, la cuarta sección es una propuesta de procesos y actividades relacionadas a la gestión de la calidad. La última sección es un plan de capacitación para implementar la propuesta de metodología de gestión de calidad en un plazo de un año.

### **4.1. Objetivo 1. Determinar la situación actual y deficiencias de la empresa GBSYS.SA en el área de gestión de la calidad.**

Para evaluar el escenario de trabajo del equipo humano de calidad, se detalla los roles, procesos, y herramientas utilizadas actualmente en la empresa GBSYS.SA asociados con el área de calidad de software.

#### **4.1.1 Roles y tareas**

El equipo de aseguramiento y control de calidad de la empresa GBSYS.SA está integrado por 7 profesionales. A continuación, se detalla los roles identificados y las tareas que realizan:

*Cuadro 13. Roles y tareas del equipo de control y aseguramiento de la calidad de la empresa GBSYS en la actualidad.*

Rol	Tareas
Probador ( Tester)	<ul style="list-style-type: none"> <li>• Analizar requerimientos</li> <li>• Seleccionar los casos de uso</li> <li>• Ejecutar los casos de uso</li> <li>• Reportar incidencias a los desarrolladores.</li> </ul>
Líder de calidad	<ul style="list-style-type: none"> <li>• Servir de puente entre el equipo de QA y el administrador de proyectos.</li> <li>• Dar seguimiento de tareas y asignaciones.</li> <li>• Dar seguimiento del registro de incidencias en el proyecto.</li> <li>• Ayudar a resolver impedimentos en el equipo.</li> <li>• Decidir cuándo terminar el proceso de prueba en acuerdo con el administrador del proyecto.</li> <li>• Asignar los recursos de QA para cada proyecto.</li> </ul>

Fuente: Elaboración propia

#### 4.1.2 Procesos, pruebas y herramientas

La siguiente sección detalla los procesos, pruebas y técnicas de la empresa GBSYS.SA utilizadas para el proceso de gestión de la calidad. Los mismos se identificaron mediante entrevistas realizadas a integrantes del equipo de QA y el coordinador de proyectos de la empresa.

##### 4.1.2.1 Procesos de calidad en el ciclo de vida los proyectos

Los procesos relacionados a calidad de producto de software están presentes en el transcurso de la vida del proyecto. Por medio de una recopilación de información, se identificaron los procesos que realiza la empresa relacionados con área de calidad, los mismos se ejemplifican en la figura 6 y se detallan a continuación:

- **Fase de planificación:** En esta fase se incluye dentro del plan de proyecto, los niveles de prueba (unitarias, integrales y aceptación) que deben ejecutarse en el producto de software. También se indica los tipos de pruebas no funcionales que deben realizarse al terminar el producto. En el cronograma del proyecto se estiman las fechas de diseño y ejecución de pruebas.
- **Fase de diseño:** El diseño de las pruebas conlleva las actividades de crear los estándares visuales, seleccionar y realizar la especificación de los casos de prueba.
- **Fase de Ejecución:** En la fase de diseño y ejecución se realizan procesos de pruebas en paralelo con algunos procesos del ciclo de vida de desarrollo. Después de realizar el diseño, se continua con la ejecución en la cual se ejecutan los casos de prueba, registran incidentes, y se cierran los incidentes. Puede suceder que, al finalizar el proceso de ejecución, el equipo de QA brinde los insumos al administrador de proyectos para realizar pruebas de aceptación con el usuario.

- **Fase de monitoreo y control:** El control de los procesos de pruebas es realizado por el Líder de calidad el cual informa al administrador de proyecto el avance de las pruebas por medio de reuniones no formales.

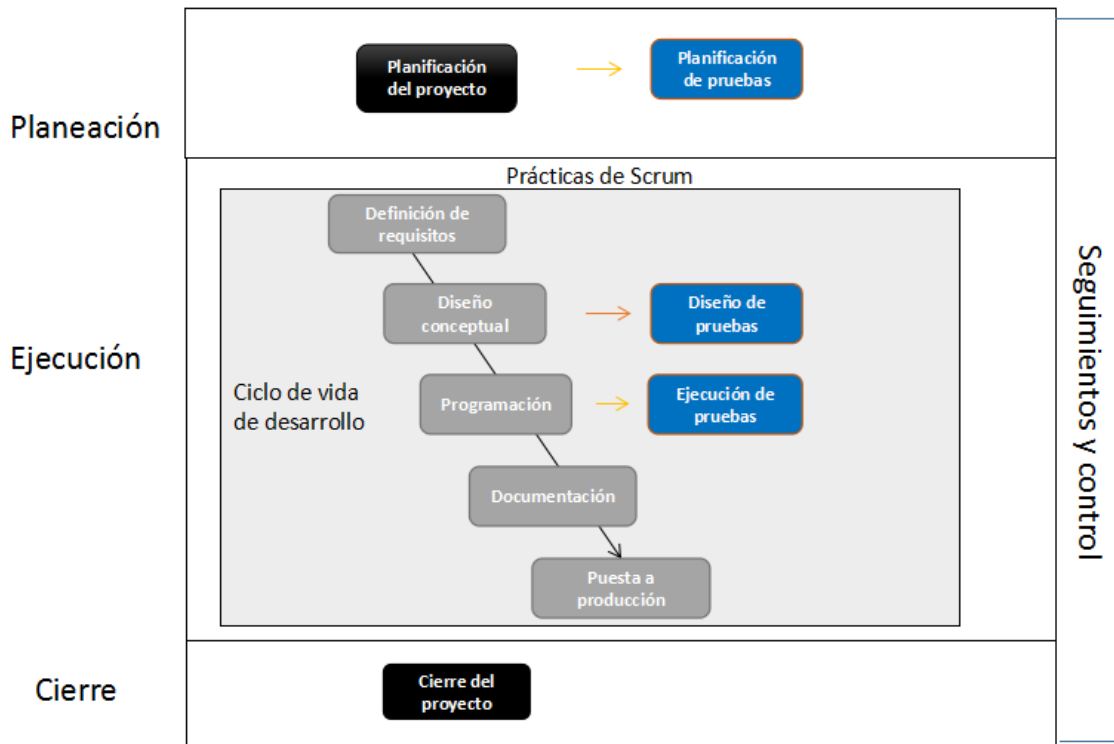


Figura 6. Procesos de calidad de software en el ciclo de vida del proyecto en GBSYS.SA

Fuente: Elaboración propia.

#### 4.1.2.2 Niveles de pruebas en el ciclo de vida de software

El ciclo de desarrollo en la empresa GBSYS.SA se caracteriza por ser un modelo en cascada al que se le incluye tres niveles de pruebas los cuales se pueden visualizar en la figura 7.

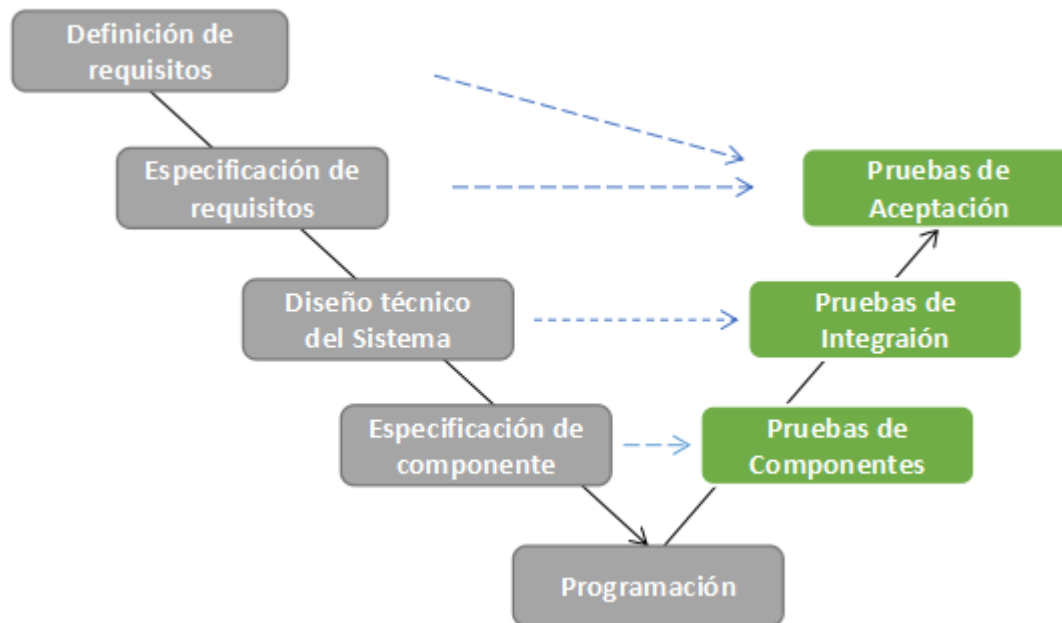


Figura 7. Niveles de pruebas en el ciclo de desarrollo de software en GBSYS.SA

Fuente: Elaboración propia.

. A continuación, se explica cada uno de los niveles de prueba utilizados en el ciclo de desarrollo por la empresa:

- **Pruebas de componentes:** También conocidas como pruebas unitarias, se ejecutan para probar una funcionalidad de manera individual, puede ser un método o acción ejecutada por el programa. Por lo general son ejecutadas de forma manual por un probador.
- **Pruebas de integración:** Las pruebas de integración se ejecutan al finalizar el desarrollo para probar todas las interacciones entre los módulos. Es muy frecuente realizar diagramas flujos de procesos para la confección de los casos de pruebas.
- **Pruebas de aceptación:** Este tipo de pruebas son ejecutadas por el usuario para validar los requerimientos funcionales en el sistema desarrollado.



#### 4.1.2.3 Tipos de pruebas de software en GBSYS.SA

A continuación, se identifican los tipos de pruebas utilizados por el equipo de QA en la ejecución de las pruebas de software.

- **Pruebas funcionales:** El objetivo de este tipo de pruebas es comprobar que el sistema funciona de acuerdo a los requerimientos solicitados por cliente. El equipo de QA utiliza la técnica caja negra, la cual se caracteriza por revisar una funcionalidad desde un punto externo sin tener el conocimiento de cómo está construido. La técnica de caja negra que utilizan es basada en los casos de uso, es decir, los casos de prueba se obtienen de la especificación descrita en el caso de uso.
- **Pruebas no funcionales:** Se utilizan para probar criterios no relacionados con la funcionalidad. En la mayoría de los casos, el encargado de diseñar y ejecutar las pruebas no funcionales es el líder de QA o un desarrollador que no pertenece al equipo y son realizadas a nivel de sistema.
- **Pruebas de regresión/confirmación:** Son el tipo de pruebas ejecutadas para probar si los cambios realizados por arreglar un defecto en el software pudieron afectar alguna funcionalidad asociada o si existe cambios en los requerimientos que desencadena que el equipo tenga que volver a probar las funcionalidades afectadas. Este tipo de pruebas se ejecutan en cualquier nivel de pruebas.

#### 4.1.2.4 Casos de prueba

Los casos de prueba es un documento utilizado para las revisiones que contiene el nombre de la prueba, descripción de la prueba, pasos a seguir y resultados esperados. El equipo de QA utiliza la técnica de casos de prueba basados en casos de uso, la cual se caracteriza en crear un conjunto de casos de pruebas (suites de pruebas) que tienen como objetivos probar pequeñas funcionalidades descritas en un caso de uso. En la figura 8, se detalla la plantilla de caso de prueba que utiliza la empresa GBSYS.SA, la cual consta de la siguiente partes

- **Nombre de caso de prueba:** Nombre breve de la funcionalidad que se va a probar
- **Descripción del caso de prueba:** Descripción detallada del caso de prueba.
- **Pasos a seguir:** Secuencia de pasos al realizar para ejecutar la prueba.
- **Resultados esperados:** Comportamiento del sistema después de haber ejecutado los pasos.
- **Precondiciones:** Estado inicial del sistema antes de ejecutar el caso de prueba.

<p><b>Nombre del caso de prueba</b></p> <p>Resumen</p> <p>Descripción:</p> <p>Descripción del Caso de Prueba</p> <p>Pasos a seguir:</p> <ol style="list-style-type: none"><li>1. Paso 1</li><li>2. Paso 2</li><li>3. Paso 3</li><li>4. Paso 4</li><li>5. .....</li></ol> <p>Resultado Esperado:</p> <ol style="list-style-type: none"><li>1. Resultado 1</li><li>2. Resultado 2</li><li>3. .....</li></ol> <p>Precondiciones</p> <ol style="list-style-type: none"><li>1. Precondición 1</li><li>2. Precondición 2</li></ol>
--

Figura 8. Plantilla actual de caso de prueba de la empresa GBSYS.SA

Fuente: Activo de GBSYS.SA

#### **4.1.2.5 Herramienta de gestión de casos de prueba e incidentes**

La empresa GBSYS.SA dispone de dos herramientas de software libre para la gestión de casos de pruebas e incidentes. El objetivo de estas herramientas es facilitar el control y seguimiento de los planes de prueba, automatizar las tareas de los probadores relacionadas a registro de incidencias y ejecución de casos de prueba, informar adecuadamente al equipo del proyecto sobre las pruebas (defectos encontrados, resueltos y pendientes) y llevar un seguimiento del ciclo de vida de un defecto desde que inicia hasta que es cerrado.

Para la gestión de casos de prueba la empresa utiliza el software Testlink el cual tiene funcionalidades para agregar requerimientos, planes de prueba, casos de prueba y visualizar el seguimiento de la ejecución de los planes de prueba. El Testlink permite generar informes sobre el avance, de este modo el líder de calidad puede visualizar el porcentaje de casos de prueba ejecutados, fallados y en estado de pendiente. El Testlink es únicamente utilizado por los integrantes del equipo de QA.

Navegador - Especificación de Pruebas	Especificación de Pruebas
<p><b>Opciones</b> <span>▲ ?</span></p> <p>Actualizar el Árbol de Navegación después de cada operación <input checked="" type="checkbox"/></p> <p><b>Filtros</b> <span>▲</span></p> <p>ID del Caso de Prueba: <input type="text" value="PRU-"/></p> <p>Título del Caso de Prueba: <input type="text"/></p> <p>Suite de Pruebas: <input type="text"/></p> <p>Tipo de ejecución: <input type="text" value="[cualquiera]"/> ▼</p> <p><input type="button" value="Aplicar Filtro"/> <input type="button" value="Resetear Filtros"/></p> <p><input type="button" value="Expandir árbol"/> <input type="button" value="Recoger árbol"/></p> <p> <ul style="list-style-type: none"> <li>▲ PROYECTO PRUEBA (2)               <ul style="list-style-type: none"> <li>▲ CU01- Nombre de Suite de Pruebas (2)                   <ul style="list-style-type: none"> <li>PRU-1:CU001-CP001 Nombre del Caso c</li> <li>PRU-2:CU001-CP002 Nombre del Caso c</li> </ul> </li> </ul> </li> </ul> </p>	<p><b>Propósito:</b></p> <p>La <i>Especificación de Pruebas</i> permite a los usuarios ver y editar todo el contenido existente para <i>Suites de Pruebas</i> y <i>Casos de Prueba</i>. Los Casos de Prueba son versionados y todas las versiones anteriores están disponibles y pueden ser vistas y gestionadas desde aquí.</p> <p><b>Primeros pasos:</b></p> <ol style="list-style-type: none"> <li>1. Selecciona tu Proyecto de Pruebas en el árbol de navegación (el nodo raíz). <i>Por favor, ten en cuenta que: Siempre puedes cambiar el Proyecto de Pruebas activo seleccionando uno diferente en la lista desplegable de la esquina superior derecha.</i></li> <li>2. Crea una nueva Suite de Pruebas pulsando en <b>Nueva Suite de Pruebas</b>. Las Suites de Pruebas pueden ser utilizadas para estructurar tus documentos de prueba de acuerdo a tus necesidades (tests funcionales/no funcionales, componentes del producto o características, peticiones de cambio, etc.). La descripción de una Suite de Pruebas puede contener el alcance de los casos de prueba incluidos, la configuración por defecto, enlaces a documentos importantes, limitaciones y otra información de utilidad. En general, todas las anotaciones que son comunes a los Casos de Prueba incluidos. Las Suites de Pruebas se comportan como un directorio escalable, por lo que los usuarios pueden mover y copiar las Suites de Pruebas dentro del Proyecto de Pruebas. Además, las Suites de Pruebas pueden ser importadas o exportadas (incluyendo los casos de prueba que contienen).</li> <li>3. Las Suites de Pruebas son directorios escalables. Los usuarios pueden mover y copiar las Suites de Pruebas dentro del Proyecto de Pruebas. Las Suites de Pruebas podrían ser importadas o exportadas (incluidos los Casos de Prueba).</li> <li>4. Seleccionar tu recién creada Suite de Pruebas en el árbol de navegación y crea un nuevo Caso de Prueba pulsando en <b>Crear Caso de Prueba</b>. Un Caso de Prueba especifica un escenario de pruebas en particular, resultados esperados y campos personalizados definidos en el Proyecto de Pruebas (consulta el manual de usuario para más información). Además es posible asignar <b>keywords</b> para mejorar la trazabilidad.</li> <li>5. Navega por la vista en árbol del lado izquierdo y edita la información. Los Casos de Prueba almacenan su propio historial.</li> <li>6. Asigna tu Especificación de Pruebas al <a href="#">Plan de Pruebas</a> cuando tus Casos de Prueba estén preparados.</li> </ol> <p>Con TestLink organizas los casos de prueba en suites de pruebas. Las Suites de Pruebas pueden ser anidadas dentro de otras suites de pruebas, permitiéndote crear jerarquías de suites de pruebas. Entonces puedes imprimir esta información junto con los casos de prueba.</p>

Figura 9. Lista de casos de prueba en la herramienta Testlink

Fuente: Herramienta Testlink de la empresa GBSYS.SA

**Navegador - Especificación de Pruebas**

**Opciones**

Actualizar el Árbol de Navegación después de cada operación

**Filtros**

ID del Caso de Prueba: PRU-

Título del Caso de Prueba:

Suite de Pruebas:

Tipo de ejecución: [cualquiera]

Aplicar Filtro | Resetear Filtros

Expandir árbol | Recoger árbol

PROYECTO PRUEBA (0)

CU01- Nombre de Suite de Pruebas (0)

---

**Suite de Pruebas : CU01- Nombre de Suite de Pruebas - Crear Caso de Prueba**

Crear | Cancelar |  marca para crear otro caso de prueba después de guardar

**Título del Caso de Prueba**

CU001-CP001 Nombre del Caso de prueba

**Resumen**

Fuente HTML

Formato: Normal

**Descripción del caso de prueba**

**Pasos a Seguir**

1. Paso 1
2. Paso 2
3. Paso 3
4. ....

**Resultados Esperados**

1. ....
2. ....
- 3

body ol li b

**Precondiciones**

Fuente HTML

Formato: Normal

**Descripción de precondiciones**

body p

Estado: Draft | Estimated exec. duration (min):

Keywords

Figura 10. Registro de casos de prueba en la herramienta Testlink

Fuente: Herramienta Testlink de la empresa GBSYS.SA

Última Ejecución (cualquier build) - Build : Version del Producto

Fecha : 05/07/2018 16:02:22 - Testeado por : - Build : Version del Producto - Estado : Pasado

Última Ejecución (build actual) - Build : Version del Producto

Fecha	Build	Testeado por	Estado	Exec (min)	Versión	Modo de ejecución
05/07/2018 16:02:22	Version del Producto		Pasado		1	

**Resumen**

Descripción del caso de prueba

**Pasos a Seguir**

- Paso 1
- Paso 2
- Paso 3
- .....

**Resultados Esperados**

- ....
- ....
- ....

**Precondiciones**

Descripción de precondiciones

Tipo de ejecución : Manual  
Estimated exec. duration (min) :

**Resultado**

Notas/Descripción

No Ejecutado  
 Pasado  
 Fallado  
 Bloqueado

Execution duration (min)

Figura 11. Registro de ejecución de planes de prueba de en la herramienta Testlink

Fuente: Herramienta Testlink de la empresa GBSYS.SA

TestLink 1.9.9(Lone Ranger) [Mi Configuración | Cerrar la sesión]

Inicio | Especificación | Ejecutar | Resultados | Gestión de Usuarios | Eventos

Formato del informe (HTML) | Imprimir

Plan de Pruebas | Mostrar planes de pruebas inactivos

**Métricas Generales del Plan de Pruebas**

Estado Global de los Builds

Build	Asignado	No Ejecutado	(%)	Pasado	(%)	Fallado	(%)	Bloqueado	(%)	Completado	(%)
Estado Global de los Builds los porcentajes son calculados usando sólo los casos de prueba que tienen tester asignado en la build											

**Resultados por Suites de Pruebas de Nivel Superior**

Suite de Pruebas	Total	No Ejecutado	(%)	Pasado	(%)	Fallado	(%)	Bloqueado	(%)	Completado	(%)
CU001 NOMBRE DE CASO DE USO	3	0	0.0	0	0.0	3	100.0	0	0.0	100.0	
CU002 NOMBRE DE CASO DE USO	5	0	0.0	1	20.0	4	80.0	0	0.0	100.0	
CU003 NOMBRE DE CASO DE USO	5	0	0.0	1	20.0	4	80.0	0	0.0	100.0	

Este informe muestra los resultados por cada suite de pruebas de nivel superior. Los resultados de las suites de pruebas de nivel inferior son contabilizados en su correspondiente suite de pruebas de nivel superior.

**Información General**  
Todos los informes consideran solo la última ejecución de cada caso de prueba. Pueden ordenar las tablas haciendo clic en la cabecera de la columna.  
Generado por TestLink el 02/06/2018 09:27:33  
Tiempo de procesado (segundos) 0.14

Figura 12. Informes y métricas generados por la herramienta Testlink

Fuente: Herramienta Testlink de la empresa GBSYS.SA

Para la gestión de incidencias, se utiliza la herramienta Mantis Bug Tracker, el cual permite realizar las siguientes funciones:

1. Crear proyectos y registrar los incidentes asociados.
2. Asignación de defectos a los desarrolladores.
3. Adjuntar evidencia (imágenes) a las incidencias.
4. Seguimiento del ciclo de vida de una incidencia permitiendo cambiar estados y agregar notas.
5. Obtener métricas de número de incidentes registrados clasificándolos en pendientes, resueltos y cerrados.

Esta herramienta es utilizada por todo el equipo de proyecto y en ocasiones puede ser utilizada por los usuarios para registrar incidentes en las pruebas de aceptación.

Conectado como: [Nombre de usuario] Proyecto: PROYECTO X

Principal | Mi Vista | Ver Incidencias | Reportar Incidencia | Registro de cambios | Roadmap | Resumen | Scrum Board | FAQ | Administración | Editar Noticias | Mi Cuenta | Cerrar Sesión

Vistas recientemente: 0014432, 0014431, 0014429, 0014408, 0014427

Informador:	Reportado por:	Asignado a:	Categoría:	Severidad:	Resolución:
Cualquiera	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Cualquiera
Estado:	Cualquier con Estado:				Prioridad:
Cualquiera	Ninguno				Cualquiera
Ver:	Visibilidad:	Ver Incidencias Filtradas:	Modificadas (Pa.):	Usar Filtros de fecha:	Relaciones:
SE	Cualquiera	SI	6	No	Cualquiera
			<a href="#">Etiquetas:</a>		
Nota De:	Cualquiera	Ordenadas por:	Actualizado Descendente		
<a href="#">Ver los comentarios</a>	Todas las condiciones				

B Buscar  [ Filtros avanzados ] [ Crear Enlace Permanente ] [ Reinicializar filtro ] [ Usar filtro ] [ Administrar filtros ] [ Guardar filtro actual ]

Mostrando Incidencias (1 - 2 / 2) [ Imprimir informes ] [ Exportar a CSV ] [ Exportar a Excel ] [ Exportar XML ]

P	ID	#	Categoría	Severidad	Estado	Actualizado	Resumen
<input type="checkbox"/>	0014432		Bug / Pulga	menor	nueva	02-06-2018	Error en pantalla - Descripción Breve
<input type="checkbox"/>	0014431		Bug / Pulga	funcionalidad	asignada (Número del tester)	02-06-2018	Nombre de la Incidencia - Descripción Breve de la Incidencia

Seleccionar todos | Mover  | Aceptar

nueva	se necesitan más datos	aceptada	confirmada	asignada	resuelta	cerrada
-------	------------------------	----------	------------	----------	----------	---------

Figura 13. Lista de incidencias en la herramienta Mantis Bug Tracker

Fuente: Herramienta Mantis Bug Tracker de la empresa GBSYS.SA

Introduzca los detalles de la incidencia.

<b>*Categoría</b>	<input type="text" value="(seleccionar)"/>
<b>Reproducibilidad</b>	<input type="text" value="no se ha intentado"/>
<b>Severidad</b>	<input type="text" value="menor"/>
<b>Prioridad</b>	<input type="text" value="normal"/>
<b>Asignar a</b>	<input type="text"/>
<b>*Resumen</b>	<input type="text"/>
<b>*Descripción</b>	<div style="border: 1px solid gray; height: 80px;"></div>
<b>Pasos para reproducir</b>	<div style="border: 1px solid gray; height: 80px;"></div>
<b>Información Adicional</b>	<div style="border: 1px solid gray; height: 80px;"></div>
<b>Subir archivos</b> (Tamaño máximo: 2,097k)	<input type="button" value="Choose File"/> No file chosen <input type="button" value="Choose File"/> No file chosen <input type="button" value="Choose File"/> No file chosen
<b>Visibilidad</b>	<input type="radio"/> Público <input checked="" type="radio"/> Privado
<b>Continuar reportando</b>	<input type="checkbox"/> Marque para reportar más incidencias

\* Requerido

Figura 14. Registro de incidencias en la herramienta Mantis Bug Tracker

Fuente: Herramienta Mantis Bug Tracker de la empresa GBSYS.SA



Resumen										
<b>por Estado</b>										
	Abiertas	Resueltas	Cerradas	Total						
nueva	1	-	-	1						
asignada	1	-	-	1						
<b>por Severidad</b>										
	Abiertas	Resueltas	Cerradas	Total						
funcionalidad	1	0	0	1						
menor	1	0	0	1						
<b>por Categoría</b>										
	Abiertas	Resueltas	Cerradas	Total						
Bug / Pulga	2	0	0	2						
<b>Estadísticas de Tiempo para Incidencias Resueltas (días)</b>										
Incidencia abierta durante más tiempo										
Mayor tiempo abierta	0.00									
Tiempo Promedio	0.00									
Tiempo Total	0.00									
<b>Estadísticas por Desarrollador</b>										
	Abiertas	Resueltas	Cerradas	Total						
testlink	1	0	0	1						
<b>por Fecha (días)</b>										
	Abierto	Resuelto	Balance							
1	0	0	0							
2	0	0	0							
3	0	0	0							
7	0	0	0							
30	0	0	0							
60	2	0	+2							
90	2	0	+2							
180	2	0	+2							
365	2	0	+2							
<b>Más Activos</b>										
				Puntaje						
0014431 - Nombre de la Incidencia - Descripción Breve de la Incidencia				4						
0014432 - Error en pantalla - Descripción Breve				1						
<b>Mayor tiempo abierta</b>										
				Días						
0014431 - Nombre de la Incidencia - Descripción Breve de la Incidencia				33						
0014432 - Error en pantalla - Descripción Breve				33						
<b>por Resolución</b>										
	Abiertas	Resueltas	Cerradas	Total						
abierta	2	0	0	2						
<b>por Prioridad</b>										
	Abiertas	Resueltas	Cerradas	Total						
normal	2	0	0	2						
<b>Estadísticas por Informador</b>										
	Abiertas	Resueltas	Cerradas	Total						
	2	0	0	2						
<b>Efectividad del Informador</b>										
	Severidad	Incorrecto	Total							
	6	0	6							
<b>Informador por Resolución</b>										
	abierta	corregida	reabierta	no reproducible	no es corregible	duplicada	no es una incidencia	suspendida	no se arreglará	% Incorrecto
i	2	0	0	0	0	0	0	0	0	0%
<b>Desarrollador por Resolución</b>										
	abierta	corregida	reabierta	no reproducible	no es corregible	duplicada	no es una incidencia	suspendida	no se arreglará	% Corregido
testlink	1	0	0	0	0	0	0	0	0	0%

Figura 15. Resumen y métricas de las incidencias en la herramienta Mantis Bug Tracker

Fuente: Herramienta Mantis Bug Tracker de la empresa GBSYS.SA

#### 4.1.3 Análisis de la gestión de calidad actual

A continuación, se resume y analiza la situación actual de los procesos que realiza la empresa en el área de gestión de calidad. La siguiente tabla es una recolección de información obtenidos de entrevistas a los integrantes de QA y gerente de proyectos de la empresa.

Cuadro 14. Datos recolectados de entrevistas al equipo de trabajo de GBSYS.SA

Pregunta	Datos recolectados
<b>Sobre el proceso de gestión de calidad</b>	
¿Para cada proyecto, se crea un plan de pruebas de calidad donde se defina	No se define un plan de pruebas al inicio del proyecto. En los proyectos

Pregunta	Datos recolectados
los objetivos de las pruebas, enfoque, criterios de salida y herramientas a utilizar?	grandes se incluye en el plan de proyectos un apartado de calidad sobre los tipos de prueba a ejecutar.
¿Existe un documento en el proyecto que determine las actividades de aseguramiento y control de calidad?	No existe un documento donde se especifiquen las tareas de aseguramiento y control. Las mismas son asignadas por el líder de prueba en el transcurso de proyecto de acuerdo a las solicitudes de administrador de proyectos
¿Se define un documento de estándares visuales y de programación antes de iniciar los procesos de gestión de calidad?	En algunos proyectos se define estándares visuales y de programación. Pero no es obligatorio crearlos por ende existen proyectos donde no se crean.
¿Existe un cronograma único para las tareas del equipo de QA?	No existe un cronograma individual para las tareas de calidad. Existe un cronograma del proyecto gestionado por el administrador de proyecto, que indica únicamente la duración y fechas de las pruebas.
¿Utiliza los equipos de QA las prácticas que recomienda Scrum?	El equipo realiza reuniones diarias por medio de conversaciones virtuales donde detalla las tareas del día anterior, las tareas actuales e impedimentos. No realizan otras prácticas sugeridas por el marco de trabajo. Las reuniones de equipo se realizan a necesidad del líder de calidad.
¿Se establece fechas para la entrega	No existe fechas de entrega de

Pregunta	Datos recolectados
de informes de avance de prueba y métricas al administrador de proyectos ?	informes, generalmente se entregan cuando el administrador de proyectos lo solicita.
¿Se realizan actividades de cierre tales como cerrar los reportes de incidencias, almacenar información histórica de las pruebas y documentar las acciones aprendidas?	La única actividad de cierre es asignar el estado de cerrado a los reportes de incidencias que se encuentren en algún estado diferente.
¿Se realizan inspecciones en los proyectos para verificar que se cumple con los procesos de calidad estipulados?	No se realizan inspecciones a los procesos de calidad. La calidad de los procesos y productos recae en el administrador de proyectos, quien decide cuales son las actividades de calidad que deben ejecutarse.
<b>Sobre el proceso de pruebas</b>	
¿Conoce las tareas que realizan el rol de probador?	La mayoría de probadores concuerdan que sus tareas son el análisis de documentación, creación de casos de pruebas, ejecución de casos de pruebas y reporte de incidencias
¿Realizan una reunión previa para seleccionar las condiciones de prueba?	No se realizan reuniones de diseño de pruebas. Se asigna a cada probador un caso de uso para ser analizado, el resultado de dicho análisis son los casos de prueba que revisa la funcionalidades descrita en el caso de uso.
¿Qué tipos de pruebas y técnicas son utilizadas para la confección de los	Las técnicas utilizadas son pruebas basadas en casos de uso, pruebas

Pregunta	Datos recolectados
casos de prueba?	exploratorias, pruebas de estrés y rendimiento.
¿Realizan pruebas automatizadas?	Los probadores no realizan ningún tipo de automatización de pruebas, su perfil va orientado a analistas de calidad y no poseen conocimientos técnicos.
¿Utilizan alguna herramienta o técnica que ayude a realizar las revisiones?	Se utilizan herramientas como los diagramas de flujo para las pruebas integrales y lista de comprobación.
¿Qué tipo de pruebas no funcionales son ejecutadas en el proceso de pruebas?	Las pruebas no funcionales ejecutadas son pruebas de carga y estrés.

Fuente: Elaboración propia

**4.2 Objetivo 2. Realizar la una comparación de la situación actual de empresa en gestión de calidad de software respecto a los procesos y técnicas de calidad descritas en la norma para software IEEE 829, el estándar ISO/IEC 9126 y en el marco de conocimiento Foundation Level Syllabus de ISTQB.**

En este apartado se presenta una comparación entre procesos, herramientas de gestión de calidad, documentación y técnicas de pruebas utilizadas actualmente por GBSYS.SA y lo que indica la norma IEEE 829, estándar ISO y el marco de conocimiento internacional para calidad de software.

**4.2.1 Comparación de procesos**

A continuación, se realiza una comparación entre los procesos sugeridos por el marco del conocimiento que propone ISTQB y los procesos seguidos por la empresa.

*Cuadro 15. Comparación de los procesos y actividades de calidad expuesta en el marco del conocimiento ISTQB y los procesos de calidad realizados por GBSYS.SA*

Proceso	Actividades según el marco de conocimiento ISQTB	Según empresa GBSYS
Planeación	Crear el plan de pruebas ( determinar el alcance, riesgos, objetivos, criterios de salida, calendario y enfoque.	No se crea ningún plan de prueba antes de empezar las actividades de calidad.

Proceso	Actividades según el marco de conocimiento ISQTB	Según empresa GBSYS
	Implementar una estrategia para las pruebas ( descripción a alto nivel de los niveles de pruebas a llevar a cabo)	El nivel de pruebas se indica dentro de plan de gestión del proyecto, sin embargo existe una confusión entre los conceptos de tipo de prueba y nivel de prueba.
	Adquirir/ obtener los recursos para pruebas ( personal, entorno)	Esta actividad se realiza en la etapa de diseño o implementación de las pruebas.
Diseño	Identificar las condiciones de prueba y datos de prueba.	Se identifica condiciones de prueba relacionados a los casos de uso. Los datos de prueba se identifican en la implementación.
	Crear y priorizar los casos de prueba	Se crean los casos de pruebas y se especifican. No se priorizan los casos de prueba, se ejecutan dependiendo de la fecha de entrega por parte de equipo de desarrollo.
Implementación y Ejecución	Finalizar, implementar los casos de pruebas	Se finaliza y se revisan los casos de prueba

Proceso	Actividades según el marco de conocimiento ISQTB	Según empresa GBSYS
	Crear los juegos de prueba de los procedimientos.	Los casos de prueba se agrupan por casos de uso conformando un juego de pruebas
	Identificar y crear los datos de prueba	Se identifican y crean los datos de prueba.
	Redactar guiones de pruebas automatizadas	No se crean guiones para pruebas automatizadas
	Verificar y actualizar la trazabilidad ( bases de prueba casos de prueba)	La trazabilidad se define en el diseño de las pruebas indicando el código del caso de uso asociado al caso de prueba
	Ejecutar las pruebas ( manuales y automatizadas)	Se ejecutan las pruebas manuales. Se ejecutan los objetos de pruebas que estén desarrollados.
	Registrar los resultados de prueba y análisis	Cada vez que se ejecuta un caso de prueba se registra el resultado de las prueba (fallado, pasado, bloqueada) en la herramienta de gestión de pruebas.

Proceso	Actividades según el marco de conocimiento ISQTB	Según empresa GBSYS
	Informar y analizar incidencias con el objeto de establecer sus causas	Se registran los incidentes en la herramienta de gestión de incidentes para informar a los desarrolladores.
	Repetir actividades de prueba para confirmar una corrección.	Se revisan los incidentes para confirmar que han sido corregidos.
	Ejecutar pruebas de regresión.	Se vuelve a ejecutar los casos de pruebas relacionados a los defectos corregidos
Evaluación de criterios de salida	Evaluar la ejecución de pruebas con respecto a los objetivos definidos	No se realiza evaluaciones de los criterios de salida.
	Proporcionar información con el objeto de dar lugar a la decisión de llevar a cabo pruebas adicionales	Se realizan reuniones no formales entre el líder de prueba y el Project manager para determinar cuándo acabar las pruebas.
Cierre	Recopilar datos de las actividades del proceso de pruebas finalizadas con el objeto de consolidar la experiencia.	Se revisa el estado de los casos de prueba en la herramienta de gestión de pruebas.



Proceso	Actividades según el marco de conocimiento ISQTB	Según empresa GBSYS
	Cerrar informes de incidencias.	En algunas ocasiones se revisan los incidentes para cerrarlos.
	Comprobar qué entregables planificados han sido entregados y probados.	Por lo general si existe algún documento de las pruebas solicitado por el administrador de proyectos, se efectúa al terminar la ejecución.
	Finalizar y archivar los productos de soporte de pruebas.	No se archivan los productos de soporte de pruebas.
	Analizar las acciones aprendidas y utilizar la información recopilada para mejorar la madurez de las pruebas.	No se analiza las acciones aprendidas para mejorar el proceso.

Fuente: Elaboración propia

#### 4.2.2 Comparación entre tipos y técnicas de prueba

A continuación, se realiza una comparación entre los tipos y técnicas de pruebas sugeridos por el marco metodológico que propone ISTQB y los utilizados por la empresa actualmente.

*Cuadro 16. Comparación de las técnicas de diseño de pruebas expuesta en el marco del conocimiento ISTQB y las técnicas de diseño realizadas por GBSYS.SA*

<b>Tipos de Técnicas</b>	<b>Tipos de pruebas / diseño de pruebas según ISTQB y estándar ISO 9156</b>	<b>Según empresa GBSYS</b>
Pruebas funcionales	Técnicas de diseño de pruebas de caja negra: <ul style="list-style-type: none"> <li>• Partición de equivalencia</li> <li>• Análisis de valores límite</li> <li>• Pruebas de transición de estado</li> <li>• Basadas en casos de uso</li> </ul>	Se utiliza la técnica de diseño de pruebas basado en caso de uso
Pruebas no funcionales	Tipos de pruebas: <ul style="list-style-type: none"> <li>• Usabilidad</li> <li>• Carga</li> <li>• Rendimiento</li> <li>• Volumen</li> <li>• Estrés</li> <li>• Estabilidad</li> <li>• Robustez</li> </ul>	Se ejecutan pruebas de carga y estrés en algunos proyectos
Pruebas estructurales	Técnicas de diseño de caja blanca: <ul style="list-style-type: none"> <li>• Cobertura de</li> </ul>	No se aplica ninguna técnica de diseño de caja blanca

	<ul style="list-style-type: none"> <li>• Cobertura de rama</li> <li>• Cobertura de condición</li> <li>• Cobertura de caminos</li> </ul>	
Pruebas asociadas al cambio	<ul style="list-style-type: none"> <li>• Pruebas de confirmación</li> <li>• Pruebas de regresión</li> </ul>	Se realizan pruebas de confirmación y regresión
Pruebas estáticas	<ul style="list-style-type: none"> <li>• Análisis de flujo de control</li> <li>• Revisiones</li> <li>• Métricas de compilador / analizador</li> </ul>	No se realizan pruebas estáticas.
Pruebas basadas en la experiencia	<ul style="list-style-type: none"> <li>• Predicción de errores</li> <li>• Pruebas exploratorias</li> </ul>	No se realizan pruebas exploratorias.

Fuente: Elaboración propia

#### 4.2.3 Comparación entre documentación de prueba

La norma 829 especifica la documentación para las pruebas de software y sistemas. A continuación, se realiza una comparación de la documentación elaborada por la empresa GBSYS y lo que indica la norma.

*Cuadro 17. Comparación de documentación de pruebas de software expuesta en la norma IEEE 829 y los documentos realizados por GBSYS.SA*

<b>Documentos</b>	<b>Secciones según IEEE 829</b>	<b>Según la empresa GBSYS</b>
Plan maestro de pruebas	Secciones del documento: <ul style="list-style-type: none"> <li>• Introducción</li> <li>• Suposiciones</li> <li>• Elementos de prueba</li> <li>• Características sujetas a las pruebas</li> <li>• Características no sujetas a las pruebas</li> <li>• Enfoque</li> <li>• Criterios de paso/Fallo</li> <li>• Criterios de suspensión / reanudación</li> <li>• Entregables de pruebas</li> <li>• Necesidades relativas al entorno</li> <li>• Responsabilidades</li> <li>• Dotación de personal y formación</li> <li>• Calendario</li> <li>• Riesgos y contingencias</li> <li>• Aprobación</li> </ul>	No se realiza un plan de pruebas formal
Caso de Prueba	Secciones de documento: <ul style="list-style-type: none"> <li>• Precondiciones</li> </ul>	El documento de caso de uso de GBSYS está dividido

Documentos	Secciones según IEEE 829	Según la empresa GBSYS
	<ul style="list-style-type: none"> <li>• Valores de entrada</li> <li>• Resultados esperados</li> <li>• Post condiciones</li> <li>• Dependencias</li> <li>• Identificador distinguible</li> <li>• Requisitos</li> </ul>	<p>por las siguientes secciones</p> <ul style="list-style-type: none"> <li>• Nombre</li> <li>• Descripción</li> <li>• Pasos a seguir</li> <li>• Resultados esperados</li> <li>• Precondiciones</li> </ul>
Reporte de incidencias	<p>Secciones del reporte</p> <ul style="list-style-type: none"> <li>• Número único del defecto</li> <li>• Objeto de prueba (denominación, versión)</li> <li>• Entorno de prueba</li> <li>• Nombre de autor del informe de incidencia</li> <li>• Fecha de la primera ocurrencia.</li> <li>• Clase de defecto</li> <li>• Estado de defecto</li> <li>• Prioridad</li> <li>• Opcionales <ul style="list-style-type: none"> <li>○ Casos de prueba</li> <li>○ Descripción de la desviación</li> <li>○ Comentarios</li> </ul> </li> </ul>	<p>El reporte de incidencias se registra por medio de la herramienta Mantis, las secciones son las siguientes:</p> <ul style="list-style-type: none"> <li>• Número de defecto</li> <li>• Clase de defecto</li> <li>• Reproducibilidad</li> <li>• Severidad</li> <li>• Prioridad</li> <li>• Nombre autor a quien se le asigna la incidencia</li> <li>• Nombre del autor quien registra la incidencia</li> <li>• Resumen</li> <li>• Descripción</li> <li>• Imagen de la incidencia</li> </ul>

Documentos	Secciones según IEEE 829	Según la empresa GBSYS
	<ul style="list-style-type: none"> <li>○ Acciones correctivas</li> </ul>	<ul style="list-style-type: none"> <li>• Registro de notas</li> <li>• Estado</li> </ul>
<p>Informes de estado de pruebas</p>	<p>Secciones de reporte</p> <ul style="list-style-type: none"> <li>• Objeto u objetos de prueba</li> <li>• Nivel de prueba, ciclo de prueba, periodo del informe</li> <li>• Avance de las pruebas (utilizando métricas)</li> <li>• Recursos utilizados/presupuestos consumidos</li> <li>• Hitos alcanzados</li> <li>• Informes de defectos (número de defectos descubiertos, y corregidos)</li> <li>• Evaluación de riesgos (nuevos riesgos/ riesgos modificados)</li> <li>• Pronóstico (actividades planificadas para el próximo período de informe)</li> </ul>	<p>No se realiza informes de estado de pruebas</p>

Documentos	Secciones según IEEE 829	Según la empresa GBSYS
	<ul style="list-style-type: none"> <li>Evaluación general/estado</li> </ul>	

Fuente: Elaboración propia

#### 4.2.4 Comparación de herramienta de gestión.

A continuación, se muestra una comparación entre los tipos de herramientas sugeridos por el marco metodológico ISTQB para pruebas de software y los utilizados por la empresa GBSYS.SA.

*Cuadro 18. Comparación entre las herramientas para pruebas expuesta en el marco del conocimiento ISTQB y las técnicas de diseños realizadas por GBSYS.SA*

Tipo de herramientas para pruebas	Herramientas utilizadas en la empresa GBSYS.
Herramientas de gestión de pruebas	La herramienta utilizada para la gestión de pruebas es Testlink
Herramientas de gestión de requerimientos	Los requerimientos son realizados en Microsoft Word y son gestionados por medio de una herramienta de gestión de versiones.
Herramientas de gestión de incidentes	La herramienta para la gestión de incidentes es Mantis Bug Tracker
Herramientas para automatización de pruebas	No se utilizan herramientas para automatización de pruebas.
Herramientas de análisis estático	No se utilizan herramientas de análisis estático.

Fuente: Elaboración propia

### **4.3 Objetivo 3. Determinar las oportunidades de mejora en la gestión de la calidad de software tomando como fuente el análisis realizado.**

Para proponer los cambios que se deben de aplicar en el proceso de gestión de calidad, además de realizar una investigación de la situación actual, es necesario indagar sobre las causas que ocasionan el problema de calidad en el proyecto y su producto. Para cumplir este objetivo se realiza un análisis de las causas que originan el problema y el efecto que produce en el área de calidad. Como resultado del análisis se identifica las oportunidades de mejora tomando como referencia la norma IEEE 829 para documentación de pruebas, el marco de conocimiento ISTQB y el estándar ISO/IEC 9126 para la evaluación de calidad de software. La figura 16 muestra un diagrama de causa y efecto el cual divide las causas en las siguientes categorías:

- Personal: Causas relacionados con el recurso humano de la empresa.
- Proceso: Causas relacionadas con el proceso de gestión de calidad.
- Insumos: Causas relacionadas con los insumos o entradas para iniciar con las actividades de calidad.
- Herramientas: Causas relacionadas con el uso o faltante de herramientas para las actividades de calidad



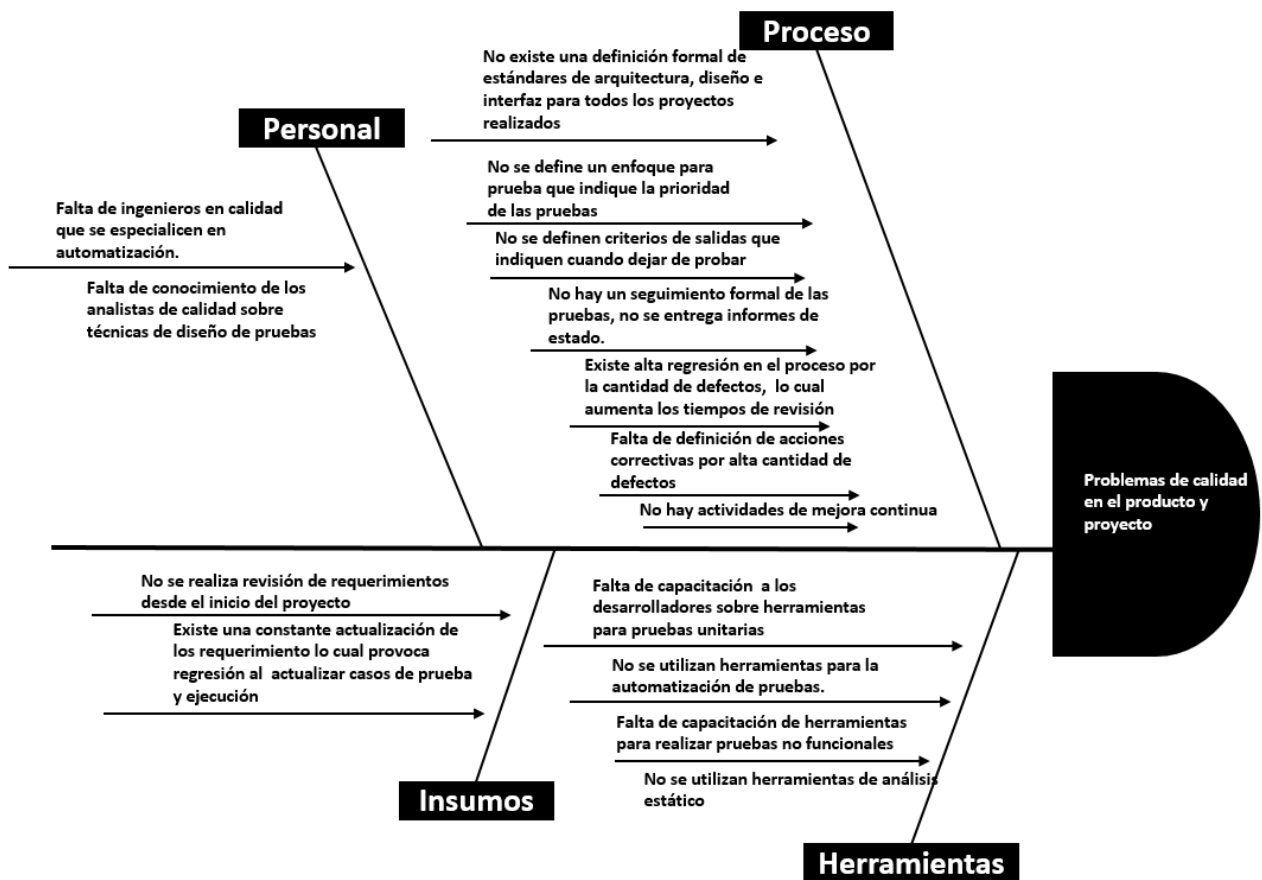


Figura 16. Diagrama de causa y efecto del proceso de gestión de calidad de GBSYS.SA

Fuente. Elaboración propia

El cuadro 19 detalla las oportunidades de mejora obtenidas del análisis del diagrama causa y efecto.

Cuadro 19. Causas del problema en la calidad de software y oportunidades de mejora en los procesos de gestión de calidad de software en la empresa GBSYS.SA

Categoría	Causas	Oportunidades de mejora
Proceso	<ul style="list-style-type: none"> <li>No existe una definición formal de estándares de arquitectura, diseño</li> </ul>	<ul style="list-style-type: none"> <li>Definir una plantilla para la definición del estándar de arquitectura para el</li> </ul>

Categoría	Causas	Oportunidades de mejora
	<p>e interfaz para todos los proyectos realizados</p>	<p>sistema, diseño de base de datos e interfaz.</p> <ul style="list-style-type: none"> <li>• Definir un plan de pruebas donde se defina el enfoque del proceso de pruebas y los criterios de salida.</li> <li>• Definir la plantilla para informes de estado.</li> <li>• Definición de estándares para evitar incidencias visuales.</li> <li>• Definición en el plan de pruebas las acciones preventivas y correctivas por alta cantidad de defectos por los desarrolladores</li> <li>• Dentro del proceso de calidad, se deben establecer tareas de cierre</li> </ul>
	<ul style="list-style-type: none"> <li>• No se define un enfoque para prueba que indique la prioridad de las pruebas.</li> </ul>	
	<ul style="list-style-type: none"> <li>• No se definen criterios de salidas que indiquen cuando dejar de probar</li> </ul>	
	<ul style="list-style-type: none"> <li>• No hay un seguimiento formal de las pruebas, no se entrega informes de seguimiento.</li> </ul>	
	<ul style="list-style-type: none"> <li>• Existe alta regresión en el proceso por la cantidad de defectos, lo cual aumenta los tiempos de revisión</li> </ul>	
	<ul style="list-style-type: none"> <li>• No hay actividades de mejora continua</li> </ul>	

Categoría	Causas	Oportunidades de mejora
Personal	<ul style="list-style-type: none"> <li>• Falta de ingenieros en calidad que se especialicen en automatización.</li> <li>• Falta de conocimiento de los analistas de calidad sobre técnicas de diseño de pruebas</li> </ul>	<ul style="list-style-type: none"> <li>• para obtener información para la actividad de mejora continua</li> <li>• Definir las herramientas de automatización a utilizar y capacitar al analista de calidad en la utilización de las mismas</li> </ul>
Insumos	<ul style="list-style-type: none"> <li>• No se realiza revisión de los requerimientos desde el inicio del proyecto</li> <li>• Existe una constante actualización de los requerimientos lo cual provoca regresión en actualizar casos de prueba y ejecución</li> </ul>	<ul style="list-style-type: none"> <li>• Capacitar a los analistas de calidad en técnicas de diseño de pruebas</li> <li>• Plantear actividades de revisiones estáticas desde el inicio de proyecto.</li> <li>• Definir herramientas para la automatización y análisis estático de pruebas y</li> </ul>
Herramientas	<ul style="list-style-type: none"> <li>• Falta de capacitación a los</li> </ul>	<ul style="list-style-type: none"> <li>capacitar a</li> </ul>

Categoría	Causas	Oportunidades de mejora
	desarrolladores sobre herramientas para pruebas unitarias	desarrolladores y analistas de calidad. <ul style="list-style-type: none"> <li>• Definir herramientas para ejecutar pruebas no funcionales y capacitar a ingenieros de calidad sobre su utilización.</li> <li>• Definir actividades de aseguramiento de calidad que verifique el seguimiento de la metodología.</li> </ul>
<ul style="list-style-type: none"> <li>• No se utilizan herramientas para la automatización de pruebas.</li> </ul>		
<ul style="list-style-type: none"> <li>• Falta de capacitación en la utilización de herramientas para pruebas no funcionales</li> </ul>		
<ul style="list-style-type: none"> <li>• No se utilizan herramientas de análisis estático</li> </ul>		

Fuente: Elaboración propia

#### **4.4 Objetivo 4. Diseñar una guía de los procesos y técnicas para la gestión de la calidad de software que se ajusten a las necesidades de la empresa considerando el análisis elaborado**

En esta sección se define una propuesta de metodología para la gestión de calidad de GBSYS.SA. Cabe aclarar que la siguiente propuesta es aplicable para

proyectos de desarrollo web a la medida, los cuales constituyen la mayor parte de la cartera de proyectos de la empresa. Según el análisis realizado en las secciones previas en este documento, se lograron identificar las necesidades y mejoras al proceso de gestión de calidad de la empresa GBSYS.SA, dichas mejoras se incluyen en la siguiente propuesta y se implementan algunas técnicas y herramientas que no son utilizadas actualmente por los equipos de calidad de software.

#### **4.4.1 Gestión de calidad**

La gestión de calidad establece un proceso y actividades para garantizar que las necesidades del proyecto y producto sean obtenidas. Este proceso contiene los siguientes sub procesos, entradas y salidas:

##### Procesos

1. Planificar la Gestión de la Calidad
2. Realizar el aseguramiento de la gestión de la calidad
3. Controlar la gestión de la calidad:

##### Entradas

1. Requerimientos funcionales y no funcionales entregados por el cliente.
2. EDT del proyecto
3. Casos de Uso
4. Documentación del contrato o condiciones de cliente

##### Salidas

1. Plan de calidad con los objetivos de calidad del proyecto y producto
2. Solicitudes de cambio
3. Entregables Verificados
4. Actualizaciones de los activos de la empresa

*Cuadro 20. Grupos de procesos de la gestión de proyecto y actividades del área de conocimiento de gestión de calidad*

GRUPO DE PROCESOS	PROCESOS DE GESTIÓN DE CALIDAD
Grupo de Procesos Iniciación	
Grupo de Procesos Planificación	<ul style="list-style-type: none"> <li>• Planificar la calidad de Software               <ul style="list-style-type: none"> <li>○ Análisis de suposiciones</li> <li>○ Análisis de alcance de las pruebas</li> <li>○ Análisis de la estrategia</li> <li>○ Definición de los niveles de prueba</li> <li>○ Definición de las necesidades del entorno</li> <li>○ Definición de los tipos de prueba</li> <li>○ Definición de los recursos para las pruebas</li> <li>○ Análisis de riesgos asociados a las pruebas</li> <li>○ Definición de los criterios de salida</li> <li>○ Definición de los criterios de suspensión / reanudación</li> <li>○ Definición acciones preventiva y correctivas</li> <li>○ Definición de entregables asociados a las pruebas</li> <li>○ Selección de herramientas y técnicas</li> <li>○ Diseño de plan de pruebas</li> </ul> </li> </ul>
Grupo de Procesos Ejecución	<ul style="list-style-type: none"> <li>• Realizar el aseguramiento de la calidad               <ul style="list-style-type: none"> <li>○ Definición de estándares</li> <li>○ Definición de reglas para herramientas automatizadas y análisis estático</li> </ul> </li> </ul>
Grupo de Procesos Seguimiento y control	<ul style="list-style-type: none"> <li>• Realizar el control de la calidad               <ul style="list-style-type: none"> <li>○ Seleccionar las técnicas de diseño de casos de prueba</li> <li>○ Análisis y diseño de los casos de prueba</li> <li>○ Implementación y ejecución de las pruebas</li> <li>○ Evaluación de criterios de salida y elaboración de reportes</li> <li>○ Cierre de pruebas</li> </ul> </li> </ul>

GRUPO DE PROCESOS	PROCESOS DE GESTIÓN DE CALIDAD
Grupo de Procesos Cierre	

Fuente: Elaboración propia.

#### 4.4.1.1 Planificar la gestión de la calidad

El proceso de planificar es el encargado de definir las actividades de aseguramiento y control de la calidad necesarias para cumplir los objetivos de calidad. Una de las tareas valiosas que se realizan en el proceso de planificación es definir las actividades de pruebas para el proceso de control de calidad de software. Para este proceso es necesario los siguientes insumos:

- **Requerimientos funcionales del sistema:** Documento de requerimientos que especifica las funcionalidades del sistema. La lista de requerimiento es otorgada por el cliente y analizada posteriormente por los analistas de negocio, en la mayoría de los casos se crean casos de uso para mejor entendimiento del equipo de proyecto.
- **Requerimientos no funcionales del sistema:** Lista de requisitos que especifican características de la operación o funcionamiento del sistema los cuales son otorgados por el cliente o indicados por el arquitecto del sistema.
- **Estructura de descomposición del trabajo:** La EDT es la representación gráfica de los elementos de trabajo y entregables. La misma debe estar incluida dentro del plan de proyectos del sistema.
- **Información sobre contratos:** Lista de restricciones o datos relevantes incluidos en el contrato con cliente que afecta directa o indirectamente el desarrollo del proyecto.

##### 4.4.1.1.1 Niveles de pruebas

Para iniciar la planeación de las pruebas es necesario distinguir el alcance de cada nivel de pruebas, los mismos deben estar indicados dentro del plan. A continuación, se detalla los niveles de pruebas de software:

- Pruebas Unitarias: El objetivo de las pruebas unitarias es probar una unidad o componente funcional. Se recomienda que este tipo de pruebas sean programadas por los desarrolladores y ejecutadas automáticamente por medio de algún framework de pruebas para validar unidades de código.
- Pruebas de Integración: El alcance de este nivel de prueba es probar que los elementos unitarios interaccionen correctamente a lo especificado en los requerimientos. Las pruebas de integración pueden automatizarse, sin embargo, hay que tomar en cuenta que la automatización eleva la dificultad en la elaboración del caso de prueba, por ello hay que analizar si los casos de prueba son de alta de regresión y dan valor agregado a las revisiones.
- Pruebas de Sistema: El alcance es validar la funcionalidad completa e integrada del sistema. Son diseñadas y ejecutadas por el equipo de aseguramiento de calidad.
- Pruebas de Usuario: Su objetivo es confirmar el cumplimiento de una serie de criterios de aceptación que confirman la satisfacción del usuario. En la mayoría de los casos son diseñadas por el usuario final.

#### **4.4.1.1.2 Tipos de pruebas**

- Pruebas funcionales: Este tipo de prueba válida que el sistema cumple con los requerimientos solicitados por el usuario. Las pruebas funcionales utilizan mayormente técnicas de caja negra y no validan la construcción interna de la aplicación (código fuente). Pueden ser manuales o automatizadas, en el proceso de análisis y diseños de pruebas se debe analizar cuales casos de prueba son candidatos a la automatización.
- Prueba de humo: Las pruebas de humo o Smoke test pueden realizarse previo a las pruebas funcionales con la finalidad de probar superficialmente toda la funcionalidad de los objetos de prueba con alto riesgo, para asegurar que estos están listos para pruebas de mayor profundidad y detectar errores de forma temprana. Las pruebas de humo también pueden ser una alternativa para cuando el tiempo de prueba es limitado, en este caso se prueba las funciones más importantes de los objetos de prueba.



- Pruebas estructurales: En este tipo de prueba el probador conoce la estructura interna del programa (código fuente). El porcentaje de cobertura es medido y utilizado como fuente para la creación de casos de pruebas adicionales. Las técnicas de caja blanca pueden ser efectivas en las pruebas unitarias, pero puede fallar a encontrar errores en etapas de integración.
- Pruebas no funcionales: Las pruebas no funcionales verifican el rendimiento, carga, estrés, usabilidad, fiabilidad y otros aspectos no funcionales del sistema de software. Para las pruebas rendimiento, carga, estrés, volumen es necesario de uso de herramientas para la efectividad de la prueba.
- Pruebas de confirmación y regresión: El tipo de pruebas de confirmación y regresión se realiza después de que la aplicación ha sido modificada. La diferencia entre la confirmación y regresión es el alcance de la prueba, las pruebas de confirmación validan la corrección del error o cambio, por otro lado, la prueba de regresión es volver a realizar la prueba de los objetos del programa que fueron impactados por un cambio o corrección de un error.
- Pruebas basadas en la experiencia: En esta clasificación se encuentran las pruebas exploratorias y predicción de errores. Los casos de prueba son realizados por el tester basada en su experiencia, habilidad e intuición. Son útiles cuando no hay requisitos o casos de prueba formales.

#### **4.4.1.1.3 Estrategia de las pruebas**

Al seleccionar y diseñar una estrategia o enfoque para las pruebas de un software, hay que conocer la estrategia general del producto, negocio y usuario. Es importante tomar en cuenta cual es la cobertura de prueba que brinda el mayor valor de acuerdo al cronograma, riesgos asociados al proyecto y restricciones de presupuesto. Es necesario una estrategia de prueba dado que no es viable probar un sistema de forma completa, probar con todas las combinaciones de datos de prueba, estados internos y restricciones temporales es casi imposible. A continuación, se indica algunos enfoques/estrategias para pruebas:

- Enfoque preventivo: Las pruebas son diseñadas tan pronto sea posible.
- Enfoque reactivo: Las pruebas se diseñan después de desarrollo. Esta estrategia funciona cuando no está disponible el equipo de calidad al inicio del proyecto o existen cuellos de botella en torno a la arquitectura del sistema en donde no se aprovecharía el rendimiento del equipo de QA.
- Enfoque analítico: Se realiza un análisis de los factores de afectan fuertemente el entorno del proyecto, por ejemplo, están las pruebas basadas en los riesgos en las cuales se diseñan las pruebas de modo que los objetos de prueba asociados a riesgos críticos son los primeros en probarse para la detección temprana de errores. De igual manera suceden con un diseño de prueba funcional, donde se analiza los requerimientos para importantes los cuales deben probarse de primero.
- Enfoque heurístico: Como las pruebas exploratorias o de humo donde las pruebas son más reactivas a las que son planificadas previamente, y donde la ejecución y la evaluación son tareas paralelas.
- Enfoque de reutilización: Se reutiliza los juegos de prueba de proyectos anteriores con el objetivo de lograr un mayor avance.
- Enfoque centrado en fallo: Se utiliza tipo de pruebas como predicción de errores o ataque de faltas donde se usa la experiencia del probador para adivinar los objetos problemáticos de la aplicación.
- Enfoque basado en lista de comprobación: Uso de listas de comprobación de proyectos previos o de la planificación de pruebas.
- Enfoque basado en consultoría: Consultores expertos guían el proceso de pruebas.

#### 4.4.1.1.4 Plan de pruebas

Para cada proyecto debe definirse un plan maestro de pruebas para software para la validación de los entregables. Dentro del plan maestro debe definirse las actividades para garantizar la calidad del producto.

*Cuadro 21. Propuesta de plan maestro de pruebas para la empresa GBSYS.SA*

PLAN DE PRUEBAS	
<b>IDENTIFICADOR</b>	<Identificador único>
<b>NOMBRE DEL PROYECTO</b>	<Nombre del proyecto>
<b>SUPOSICIONES</b>	<Suposiciones del proyecto relacionadas a calidad >
<b>ELEMENTOS DE PRUEBA ALCANCE</b>	<Alcance de las pruebas, se deben definir los elementos de prueba>
<b>ENFOQUE</b>	<Implementación de la estrategia de pruebas para el cumplimiento de los objetivos del proyecto>
<b>NIVELES DE PRUEBA</b>	<Niveles de prueba que deben ejecutarse>
<b>NECESIDADES RELATIVAS AL ENTORNO DE PRUEBAS</b>	<Insumos necesarios para comenzar el proceso de pruebas >
<b>TIPOS DE PRUEBAS</b>	<Tipos de pruebas que deben ejecutarse>
RECURSOS	
NOMBRE	ROL
<Nombre del recurso de calidad >	<Nombre del rol que desempeña>

ACTIVIDADES DE CALIDAD				
RESPONSABLE	FECHA DE INICIO	DURACIÓN	FECHA FINALIZACIÓN	DESCRIPCIÓN
<i>&lt;Responsable de la actividad&gt;</i>	<i>&lt;Fecha de inicio de la actividad&gt;</i>	<i>&lt;Duración de la actividad&gt;</i>	<i>&lt;Fecha que finaliza la actividad&gt;</i>	<i>&lt;Descripción de la actividad&gt;</i>
				2
				3
				4
				5
				6
RIESGOS				
<i>&lt;Riesgos del proyecto asociados a la calidad&gt;</i>				
CRITERIOS DE SALIDAS				
<i>&lt;Indicadores que asegura que los requisitos del proyecto ha sido cumplidos y determinan el fin de las pruebas&gt;</i>				
CRITERIOS DE SUSPENSIÓN / REANUDACIÓN				
<i>&lt;Indicadores que determinan cuando suspender el proceso de pruebas&gt;</i>				
ACCIONES PREVENTIVAS Y CORRECTIVA				
ACCIÓN	TIPO		CRITERIO DE ACTIVACIÓN	
<i>&lt;Descripción de la acción&gt;</i>	<i>&lt;Preventiva o correctiva&gt;</i>		<i>&lt;Indicador que determina si la acción debe ejecutarse&gt;</i>	

ENTREGABLES DE PRUEBA
<i>&lt;Entregables del proyecto relacionados a la calidad&gt;</i>
HERRAMIENTAS Y TÉCNICAS
<i>&lt;Herramientas y técnicas de calidad a implementarse en el proceso de pruebas&gt;</i>

Fuente: Elaboración propia

#### **4.4.1.1.5 Administración de la configuración**

Después de realizar el análisis y diseño de plan de pruebas, se identifica la configuración para controlar los cambios de los insumos de pruebas (planes de pruebas, casos de prueba, gestor de incidentes, requerimientos).

- Gestor de planes de prueba: La herramienta Testlink permite gestionar planes de prueba y asociarlos a un proyecto.

Gestión de Planes de Pruebas - Proyecto de Pruebas PROYECTO PRUEBA

Editar el Plan de Pruebas Plan de pruebas v1

**Nombre** Plan de pruebas v1

**Descripción**

Formato Normal B I U abc X<sub>2</sub> X<sup>2</sup> A A

**PLAN DE PRUEBAS**

IDENTIFICADOR	<Identificador único>
NOMBRE DEL PROYECTO	<Nombre del proyecto>
SUPOSICIONES	<Suposiciones del proyecto relacionadas a calidad >
ELEMENTOS DE PRUEBA ALCANCE	<Alcance de las pruebas, se deben definir los elementos de prueba>

body p

**Activo**

**Público**

**API Key** 5a15d329b5bda9ceeee1c3a62ad10ecff26dc0e7db5cc998356d61f2c76d939b

Actualizar Cancelar

**Archivos Adjuntos :**

Adjuntar nuevo archivo

Figura 17. Plan de pruebas en la herramienta Testlink

Fuente: Herramienta Testlink de la empresa GBSYS.SA

Un elemento importante que se agrega a los planes de prueba son los Build. Los Build en el contexto de software es la versión de un programa, en el proceso de Testing se utiliza para indicar que versión del sistema está siendo probada. Los requisitos del sistema también se pueden gestionar en el Testlink, sin embargo, la herramienta no permite el control de cambios en los documentos, por lo tanto, se debe solicitar un repositorio el cual sea gestionado por una herramienta de control de versiones. La empresa actualmente utiliza la herramienta TortoiseSVN de Apache.

Gestión de Builds - Plan de Pruebas : Plan de pruebas v1

Editar Build - Version del Producto

Título: Version del Producto

Descripción:

Activo:

Abierta:

Fecha de Lanzamiento: 05/07/2018

Cada Build se identifica a través de un Título y tiene dos atributos:

- Activo/Inactivo -> define si la Build está disponible para la funcionalidad de TestLink. Los Builds Inactivos no se muestran ni en la página de Ejecución de Pruebas.
- Abierto/Cerrado -> define si los Resultados de Pruebas pueden ser modificados para la Build.

Guardar Cancelar

Figura 18. Ingreso del Build asociado al plan de pruebas en el Testlink

Fuente: Herramienta Testlink de la empresa GBSYS.SA

#### 4.4.1.2 Realizar el aseguramiento de la calidad

El proceso de aseguramiento de la calidad de software son las actividades para verificar que los procesos y actividades realizadas para crear el producto cumplen con los objetivos de calidad de proyecto. En los proyectos de desarrollo de software es común asociar las actividades de aseguramiento como preventivas para evitar la alta ocurrencia de defectos o problemas en el proceso.

##### 4.4.1.2.1 Estándares Técnicos

Los estándares son documentos que contienen especificaciones técnicas, por lo general se crean para el equipo de trabajo y contienen observaciones brindadas por el cliente, también definen las características de los elementos de la interfaz gráfica y los objetos de la arquitectura del sistema.

Las siguientes plantillas es un ejemplo de especificación de estándares de interfaz y base de datos. Los mismos pueden variar por proyecto, sin embargo, cada

proyecto debe incluir dentro de sus actividades de aseguramiento la creación y revisión de estándares tomando como base las siguientes plantillas:

Cuadro 22. Plantilla base para estándar de interfaz gráfica

<b>ESTÁDARES DE DISEÑO PARA INTERFAZ</b>	
<b>CRITERIO</b>	<b>PARAMETROS</b>
<b>TABLAS Y ASPECTOS DE LA PÁGINA</b>	
Tipo de letra	<Formato de letra, default Arial>
Tamaño de letra de las etiquetas	<Tamaño de letra de la etiqueta de los campos>
Tamaño de letra de los Fieldsets	<Tamaño de letra en los Fieldsets, Canvas o componente similar>
Tamaño de letra en títulos de página	<Tamaño de letra de los títulos>
Formatos de fechas y números	<El formato de las fechas (dd/MM/yyyy), monto decimales(formato con 2 decimales y con separador de miles), hora (hh:mm tt) , campo fecha y hora (dd/mm/yyyy hh:mm tt) , si el campo es una cédula debe aplicar formato de cédula> <Excepciones en filtros>
Ubicación de los campos de búsqueda	<Ubicación de campos de búsqueda en la pantalla>
Formato de campos deshabilitados	<Tipo de letra, color, característica del campo>
<b>FORMULARIOS</b>	
Ubicación de botones	<Ubicación de los botones de la pantalla, alineación, separación entre botones>
Ubicación de campos y componentes	<Agrupación de componentes, columnas en la pantalla>
<b>BOTONES</b>	
Formato de iconos	<Nombre de los botones, alineamiento y nombre del icono a utilizar>
<b>TABLAS</b>	
Cantidad de registros mostrados en las tablas (paginación)	<Cantidad de registros mostrados en las tablas de la pantalla>
Ubicación de los filtros de las columnas	<Ubicación de filtros de las columnas, si aplica >
Ubicación de los botones de acciones de la tabla	<Ubicación de botones de la tablas>
Formato de título de la tabla	<Ubicación, tipo de letra del título de las tablas>
<b>VENTANAS EMERGENTES</b>	
Ubicación de botones	<Ubicación de los botones, botón por default>



Formato del título	<Tamaño de letra, alineamiento, ubicación>
Formato de la ventana	<Tamaño máximo, ubicación, características de la ventana>
<b>MENSAJES</b>	
Formatos de los mensajes de éxito	<Color del mensaje, ubicación, mensaje default>
Formatos de los mensajes de error	<Color del mensaje, ubicación, mensaje default>
Formatos de los mensajes de Advertencia	<Color del mensaje, ubicación, mensaje default>
Campos requeridos	<Color del mensaje, ubicación, mensaje default>
<b>REPORTES</b>	
<b>ENCABEZADO</b>	
Ubicación de logo	<Ubicación de logo>
Ubicación y formato de la fecha	<Ubicación y formato de la fecha>
Formato de letra	<Tipo de letra y otras características>
Ubicación y nombre del título	<Ubicación y nombre del título>
<b>PARÁMETROS DE CONSULTA</b>	
Formato título de la sección de parámetros	<Tamaño de letra, ubicación>
Formato y tipo de Letra	<Tipo de letra, características de la letra de la sección de parámetros>
Formato de los filtros	<Ubicación, otras características de los campos para filtros>
<b>CONTENIDO</b>	
Formato del encabezado	<Características de la letra, alineamientos, tamaño del encabezado de las tablas>
Formato de fechas	<Formato de las fechas>
Formatos de números, montos	<Alineamiento, máscara de números y montos>
Formatos especiales	<Aplica para otros datos con formato especial>
Formato de un reporte vacío	<Mensaje del reporte vacío, si aplica>
<b>PIÉ DE PÁGINA</b>	
Formato del pie de página	<Campos y alineamientos del pie de página>

Fuente: Elaboración propia

La siguiente tabla es la plantilla base para definir el estándar para los nombres de los objetos en la base de datos.

Cuadro 23. Plantilla base para estándar de base de datos

<b>ESTÁNDARES DE DISEÑO PARA BASE DE DATOS</b>	
<b>CRITERIO</b>	<b>ESTÁNDAR</b>
Nombres de tablas	<seudónimo>.<nombre de tabla>

Nombre de campos	<Estándar para los nombre de campos en las tablas>
Nombre de relaciones	<seudónimo>.<nombre de relación>
Nombre de procedimientos	<seudónimo>.<nombre de procedimiento>
Nombres de vistas	<seudónimo>.<nombre de vista>
Nombre de secuencia ( si aplica)	<seudónimo>.<nombre de secuencia>
Nombre de indexes	<seudónimo>.<nombre de indexes>
Nombre de triggers	<seudónimo>.<nombre de triggers>
Nombre de esquema ( si aplica)	<seudónimo>.<nombre de esquema>
Nombre de parámetros	<seudónimo>_<nombre de parámetro>
Formato para nombre de procedure	<Indicar como deben nombrar los procedimientos>
Máximo de línea de un procedimiento de BD	<Máximo de líneas para un procedimiento almacenado>

Fuente: Elaboración propia

#### 4.4.1.2.2 Análisis estático

Es análisis estático son revisiones que se realizan sin ejecutar la aplicación, por lo general, se realiza al código fuente o en la revisión de documentos en fases tempranas. Para el código fuente, existen herramientas muy útiles que se instalan en el IDE que automatizan las revisiones. A continuación, se detalla dos herramientas que pueden utilizarse para el análisis estático:

- Listas de chequeo: Las listas de verificación pueden convertirse en una herramienta veraz para la revisión estática de casos de uso y documentos de requerimiento, se puede verificar el estándar del documento y aspectos de ambigüedad.
- PMD analizador de código: El PMD es una herramienta de revisión estática automatizada y consiste en una extensión que se instala en un IDE para los sistemas programados en JAVA. Permite definir una serie de reglas de programación que el desarrollador puede ejecutar para verificación del código.
- Reshaper Visual Studio Extension: El Reshaper es un analizador de código para Visual Studio y permite verificar código en sistemas programados en .NET.

#### 4.4.1.3 Controlar la calidad

El proceso de control de calidad comprende todas aquellas actividades para validar los entregables de acuerdo a los requerimientos brindados por el cliente. En siguiente imagen muestra las actividades para cada proceso de gestión de calidad, la primera actividad del sub proceso de prueba es la planificación de las pruebas, la cual se realiza al inicio del proyecto. Las demás actividades, diseño, ejecución, elaboración de reportes y cierre, se realizan en el proceso de control de calidad.

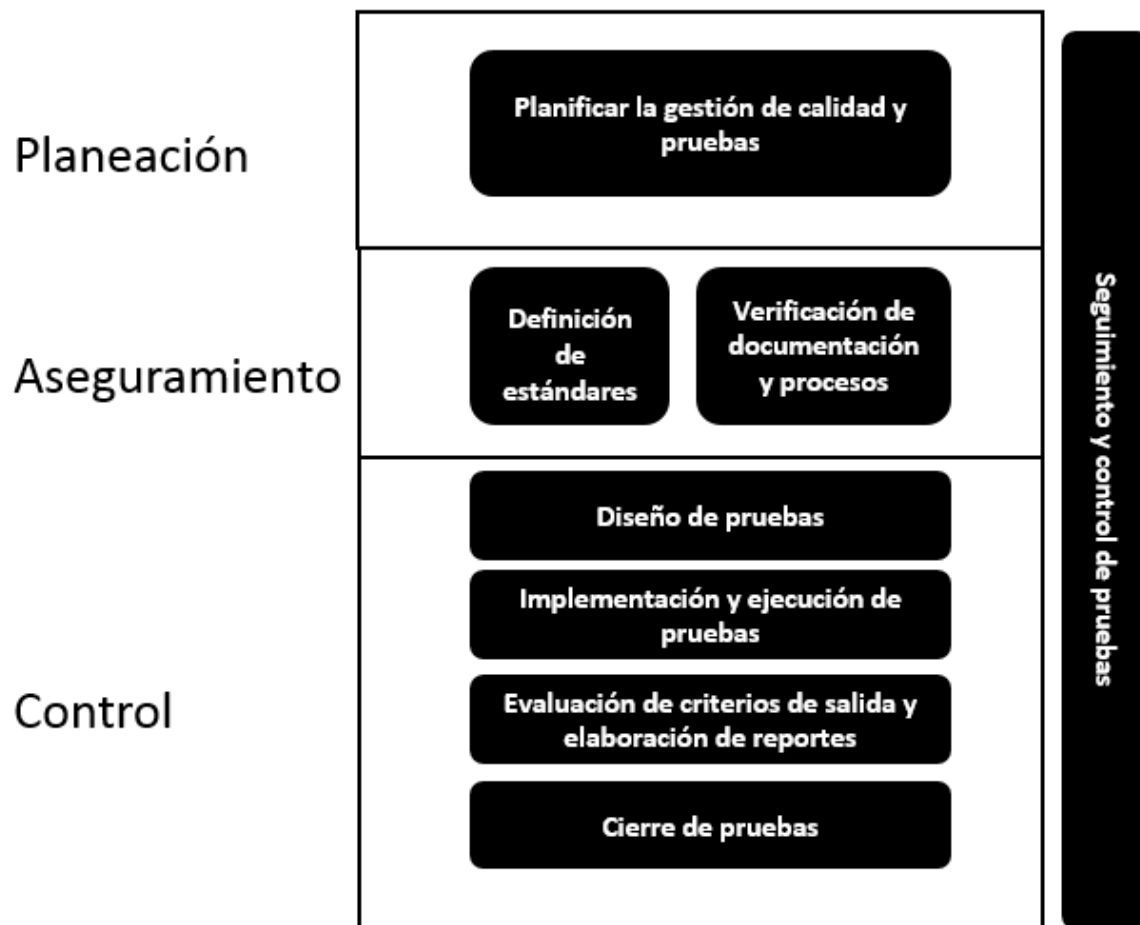


Figura 19. Actividades de los procesos de la propuesta de metodología de gestión de calidad para proyectos de desarrollo software para GBSYS.SA

Fuente: Elaboración propia

A continuación, se detallan las actividades propuestas para el proceso de control de calidad de software.

#### 4.4.1.3.1 Análisis y diseño de pruebas

En la actividad de diseño de pruebas se analizan los requerimientos, casos de uso y diseño del sistema para identificar las condiciones de pruebas. De acuerdo a las condiciones identificadas, se diseñan y priorizan los casos de prueba, posteriormente se identifican los primeros datos de prueba.

En este proceso también se debe crear el entorno de pruebas de acuerdo a las necesidades identificadas en la planificación de las pruebas. La siguiente plantilla es la propuesta para realizar el diseño del caso de prueba.

Cuadro 24. Plantilla de caso de prueba para GBSYS.SA

<b>CASO DE PRUEBA</b>				
<b>Nombre</b>	<Nombre del caso de prueba ID Nombre del Caso>			
<b>ID del caso de uso o requerimiento</b>	<Identificador del caso de uso o requerimiento origen>			
<b>Descripción</b>	<Descripción de caso de prueba>			
<b>Tipo de Prueba / Nivel de Prueba</b>	<Tipo de prueba (Funcional, No Funcional) / Nivel de Prueba ( Unitaria, Integral, Sistema)>			
<b>Dependencias</b>	<Dependencia con otro casos de prueba>			
<b>Prioridad</b>	<Prioridad del caso de prueba (ALTA, MEDIA, BAJA, MUY BAJA)>			<b>QA ejecutador:</b> <Nombre de QA>
<b>PROCEDIMIENTO</b>				
<b>Precondiciones</b>	<b>#</b>	<b>Paso a seguir</b>	<b>Resultado Esperado</b>	<b>Postcondiciones</b>
<Condición o estado del sistema antes de ejecutar el paso, solo si aplica>	1	<Descripción del paso a seguir>	<Resultado esperado al ejecutar el paso>	<Estado o condición del sistema después de ejecutar el paso, solo si aplica>
	2			
	...			

	3		
	4		
	5		
	6		
DATOS DE PRUEBA			
La siguiente tabla de datos indica los datos de prueba utilizados en la ejecución de esta prueba. Los parámetros utilizados están señalados en el casos de prueba de la siguiente forma<PnombreParametro>			
Parámetros	( Versión 1 )	(Versión 2)	
<Nombre de parámetro>	<Valor de parámetro en la versión que se prueba>	<Valor de parámetro en la versión que se prueba>	

Fuente: Elaboración propia.

La plantilla para casos de prueba se debe agregar en la herramienta Testlink para el seguimiento del caso de prueba como se observa en la figura 20.

The screenshot shows the Testlink web interface for a test case. On the left, there are filter options for 'ID del Caso de Prueba' (set to PRU-), 'Título del Caso de Prueba', 'Suite de Pruebas', and 'Tipo de ejecución' (set to 'cualquiera'). The main content area displays the test case details for 'PRU-1:CU001-CP001 Nombre del Caso de prueba'. A blue banner at the top of the main area states '¡Cuidado!: Esta versión del Caso de Prueba ha sido ejecutada'. Below this, the 'Resumen' section contains a table with the following data:

CASO DE PRUEBA			
Nombre	<Nombre del caso de prueba ID -Nombre del Caso>		
ID del caso de uso o requerimiento	<Identificador del caso de uso o requerimiento origen>		
Descripción	<Descripción de caso de prueba>		
Tipo de Prueba / Nivel de Prueba	<Tipo de prueba (Funcional, No Funcional) / Nivel de Prueba ( Unitaria, Integral, Sistema)>		
Dependencias	<Dependencia con otro casos de prueba>		
Prioridad	<Prioridad del caso de prueba (ALTA, MEDIA, BAJA, MUY BAJA)>		QA ejecutador: <Nombre de QA>

Below the summary is the 'PROCEDIMIENTO' section, which includes a table for preconditions and a table for the test steps:

Precondiciones	#	Paso a seguir	Resultado Esperado	Postcondiciones
<Condición o estado del sistema antes de ejecutar el paso, solo si aplica>	1	<Descripción del paso a seguir>	<Resultado esperado al ejecutar el paso>	<Estado o condición del sistema después de ejecutar el paso, solo si aplica>
	2...			
	3			
	4			
	5			
	6			

At the bottom, there is a 'DATOS DE PRUEBA' section with a table:

Parámetros	( Versión 1 )	(Versión 2)	

Figura 20. Caso de Prueba en la herramienta Testlink

Fuente: Herramienta Testlink de la empresa GBSYS.SA

#### 4.4.1.3.1.1 Técnicas de diseño de pruebas

Uno de los principios del proceso de pruebas es que no existen las pruebas exhaustivas, esto hace alusión que no se puede probar todas las combinaciones, por ellos se debe implementar técnicas para la identificar los casos de prueba. A continuación, se detalla las técnicas de caja negra para diseñar casos de prueba.

- **Particiones de equivalencia:** Esta técnica implica dividir por grupos los datos de entrada donde el objeto de prueba se comporta de la misma manera. El siguiente cuadro es una herramienta para identificar los datos de prueba que se pueden utilizar para construir el caso o los casos de prueba.

*Cuadro 25. Selección de particiones de equivalencia*

Condición de entrada	Tipo	Clase Equivalencia Válida	Clase de Equivalencia No Válida
<Condición específica que produce un resultado>	<Valor, Rango, Conjunto, Comportamiento>	<Grupos de Valores o valor donde hay un comportamiento igual>	<Grupo de Valores o Valor no válidos o no aceptados>

Fuente: Elaboración propia

- **Valores límites:** Los valores límites forma parte de las técnicas de particiones de equivalencia, pero en dado casos es conveniente seleccionar los valores límites de la clase de equivalencia, ya que por lo general existe más probabilidad que el sistema tenga un resultado no esperado al utilizarlos como datos de prueba, además evita que se construya casos de prueba innecesarios para probar varias combinaciones. Para utilizar los valores límites de una clase se selecciona el valor menor al límite inferior y el valor mayor a límite superior.

- **Tabla de decisiones:** La tabla de decisión es una técnica de análisis en la cual se debe de identificar las condiciones que debe de cumplir y las acciones que ejecuta el objeto de la prueba. La información se incluye en una tabla y se crean las reglas de decisión a partir de combinaciones entre las condiciones y las acciones.

*Cuadro 26. Propuesta para tabla de decisiones*

	Regla 1	Regla 2	Regla N...
<b>Condiciones</b>			
<Descripción de la condición>	<Indicador si aplica o no la condición>		
Condición 2			
Condición N....			
<b>Acciones</b>			
<Descripción de la acción>	<Indicador si aplica o no la acción>		
Acción 2			
Acción N...			

Fuente: Elaboración propia

- **Transición de estados:** Es una técnica para identificar las transiciones de estados combinando todos los eventos. Por lo general, se realiza un diagrama de transición, posteriormente se incluyen las transiciones de estado válidas y no válidas en una tabla para identificar los casos de prueba.
- **Basado en casos de uso:** En esta técnica los casos de prueba son diseñado dividiendo las funcionalidades descritas en el caso de uso.

#### 4.4.1.3.1.2 Diseño de pruebas no funcionales

Las pruebas no funcionales se realizan para validar requerimientos no funcionales, los cuales son criterios que se usan para examinar el funcionamiento del sistema y no su comportamiento. A nivel de esta propuesta se detalla herramientas para diseñar pruebas no funcionales:

- **Lista de Verificación**

Las listas de chequeos o verificación es una herramienta para controlar el cumplimiento de actividades, criterios o requisitos. Para el diseño de pruebas no funcionales se puede definir los criterios que definen cada atributo de calidad (usabilidad, mantenibilidad, portabilidad, escabilidad, seguridad) para posteriormente validar si el objeto de prueba cumple con el mismo. El siguiente cuadro es una lista de verificación para validar los criterios de usabilidad en un sistema, dicha plantilla puede utilizarse para definir otros atributos de calidad.

*Cuadro 27. Lista de verificación para validar la usabilidad de una aplicación de software*

CRITERIO	DESCRIPCIÓN DEL CRITERIO	SI	NO	N/A	OBSERVACIONES
Visibilidad del estado del sistema	Al realizar acciones o procesar información, el sistema muestra el estado en que se encuentra.				<i>&lt;Detalle de cómo el sistema cumple el criterio&gt;</i>
Eficiencia de uso y flexibilidad	Se puede realizar las acciones que invocan a funcionalidades de manera sencilla y eficiente				
Control y libertad del Usuario	El usuario tiene la alternativa de guardar cambios de un formulario o salir sin actualizar.				
Consistencia y estándares	El sistema muestra un estándar consistente en todas sus pantallas y reportes				
Prevención de errores	Incluye información preventiva en el punto problemático				
Utiliza el lenguaje del usuario					
Minimizar la carga de memoria del usuario (fácil de aprender)	Al entrar a una pantalla o funcionalidad tiene la información clara como interactuar o digitar los				



	campos mostrados.				
Ayuda y documentación	Existe una manual de usuario en línea donde se puede consultar de las funcionalidades de cada módulo.				

Fuente: Elaboración propia.

- **Apache JMeter**

Es un software de código libre que se utiliza para ejecutar pruebas de rendimiento. El Jmeter permite realizar guiones automatizados que simulan uno o varios escenarios para posteriormente ejecutarlos en paralelo. Para el diseño de los casos de prueba, los mismo debe especificarse en la plantilla de casos de prueba asignándole al tipo de prueba como no funcional de rendimiento (carga, estrés, desempeño) y especificar el procedimiento con los resultados esperados tomando en cuenta las métricas que se detallan a continuación:

*Cuadro 28. Métricas para pruebas de rendimiento*

Métrica	Descripción
Desviación Estándar	Indica la distancia promedio que hay entre los diferentes tiempos de respuesta de todos los hilos ejecutados respecto al tiempo promedio. Una menor distancia indica que los hilos se respondieron relativamente en el mismo tiempo promedio, lo cual indica que los servicios se ofrecen en las mismas condiciones para todos. Un consejo es que la desviación estándar debe ser menor o igual a la mitad del tiempo promedio para una etiqueta.
Periodo de subida	Indica el lapso de tiempo que la herramienta tiene para crear todos los hilos. El valor por defecto es 0, por lo que todos los hilos del grupo serían creados a la vez. Si el período de subida es T segundos y el número total de hilos es N, JMeter creará un hilo cada T/N

Métrica	Descripción
	segundos.
Rendimiento (Throughput):	Número de peticiones procesadas en una unidad de tiempo, que puede ser segundos, minutos y horas
Tiempo medio de respuesta (Media)	Es el valor medio del juego de datos cuando estos son ordenados de menor a mayor. En los casos en que la cantidad de datos es par, se toman los dos valores centrales y se saca el promedio de estos. Indica el tiempo promedio de respuesta en Ms (milisegundos) de todos los hilos ejecutados en el intervalo de tiempo
Tiempo Mínimo de respuesta (Min)	El menor tiempo transcurrido de una petición con la misma etiqueta o paso.
Tiempo Máximo de respuesta (Min)	El tiempo transcurrido más alto de una petición con la misma etiqueta o paso

Fuente: Elaboración propia.

Posteriormente se crean los guiones automatizados con el Jmeter y se especifica en el caso de prueba cuál guion debe ejecutarse en el proceso de implementación.

#### 4.4.1.3.2 Implementación y ejecución de pruebas

En este proceso se ejecuta los juegos de pruebas diseñados, se crean los datos de prueba, se finaliza la preparación del entorno de pruebas y se reportan los incidentes encontrados. El caso de prueba se puede ejecutar de forma manual u optar por la automatización cuando los casos son de alta probabilidades de regresión. A continuación, se detalla las herramientas a utilizar en el proceso de ejecución de pruebas:

- **Testlink:** Esta herramienta permite dar seguimiento a los juegos de prueba. El Testlink permite al analista de calidad indicar cuales casos de prueba no ha sido probados, pasados o fallidos. Cuando un caso de prueba registra

un incidente, la herramienta permite asociar el caso de prueba con el registro de la incidencia en el Mantis Bug Tracker.

- **Mantis Bug Tracker:** El Mantis BT permite el registro y seguimiento de incidentes reportados. El desarrollador puede visualizar los incidentes asignados para posteriormente indicar al analista de calidad que el defecto reportado ha sido corregido. La siguiente imagen muestra el flujo de estados de un incidente en le Mantis BT.



Figura 21. Ciclo de vida de una incidencia en el Mantis BT.

Fuente: Elaboración propia

#### 4.4.1.3.3 Evaluación de criterios de salida y elaboración de reportes

Los criterios de salida son métricas que determinan cuando terminar las pruebas, los mismos se indican en el plan maestro de prueba. Lo ideal es realizar una evaluación de los criterios cada vez que se termina la ejecución de un juego de pruebas para validar el cumplimiento de los casos de prueba asociados, dicha tarea es responsabilidad de líder de calidad de pruebas. Además de la validación, se debe informar al administrador del proyecto sobre estado del proceso de pruebas mediante un informe de seguimiento. En la cuadro 29 se muestra la plantilla para el informe de estado.

Cuadro 29. Informe de seguimiento de pruebas para GBSYS.SA

<b>Reporte de Seguimiento</b>	
Fecha	<Fecha de actual>
Nivel de prueba	<Nivel de prueba actual>
Avance de la pruebas	<Casos de pruebas ejecutados, fallidos, no ejecutados>
Recursos utilizados	<Recursos involucrados en el proceso de prueba>
Informe de defectos	<Número de defectos descubiertos, corregidos y cerrados>
Evaluación de riesgos	<Nuevos riegos o riesgos modificados>
Pronóstico	<Actividades planificadas para el próximo período de informe>
Estado general	<Resumen del estado general del proceso de pruebas>
Autor del Informe	<Nombre del autor del informe>

Fuente: Elaboración propia

#### **4.4.1.3.4 Cierre de pruebas**

En el proceso de cierre los probadores revisan las listas de incidencias en el Mantis BT para asignar el estado de cerrado a los incidentes pendientes. En este proceso también el líder de pruebas debe indicar que el entorno de pruebas debe ser liberado y guardar los documentos para el historial de las pruebas. Por último, debe recolectar y analizarse las lecciones aprendidas del proyecto para mejorar el

proceso. A continuación, se presenta la plantilla para el registro de las lecciones aprendidas.

*Cuadro 30. Registro de lecciones aprendidas para el proceso de pruebas*

<b>REGISTRO DE LECCIONES APRENDIDAS</b>
Versión 1.0
<b>Información general</b>
Nombre del Proyecto: <Nombre del proyecto>
<b>Número de Referencia:</b>
< Consecutivo único>
<b>Registrado por:</b>
<Persona que elabora el documento>
<b>Fecha de Registro:</b>
<Fecha de elaboración del documento>
<b>Descripción de la Situación:</b>
<Describe los situación que ocasionaron la lección aprendida >
<b>Impacto en el proceso de prueba</b>
<Efecto positivo o negativo de la situación descrita en el proyecto >
<b>Acciones Correctivas / Preventivas Implementadas:</b>
<Lista de acciones correctivas implementadas para disminuir o acrecentar los efectos de la situación>
<Acciones preventivas para reducir o incrementar la posibilidad de que se presenten a futuro>
<b>Lección Aprendida / Recomendaciones:</b>
<Acciones y estrategias a considerar a futuro para aprovechar>

Fuente: Activos de GBSYS.SA

#### **4.5 Objetivo 5. Crear un plan de implementación para los involucrados para transmitir el conocimiento sobre la propuesta de metodología en gestión de calidad de software y la apliquen en sus proyectos.**

En la siguiente sección se presenta un plan de implementación para poner en marcha la propuesta de metodología de gestión de calidad para la empresa GBSYS.SA. Se aclara que es vital las capacitaciones para los interesados y el

equipo para el éxito en la implementación, debido a que la utilización de la técnica planteada en la propuesta exige equipos de calidad *knowhow* (saber cómo hacer algo pronto y bien hecho).

Cabe aclarar que los procesos de planificación, seguimiento, ejecución y cierre del este plan de implementación no son parte del alcance de este documento. El plan de implementación va actuar como una guía para la empresa cuando priorice dentro de su estrategia empresarial la implementación de esta propuesta de metodología para gestión de calidad.

#### **4.5.1 Objetivos del plan de implementación**

Se identificaron los siguientes objetivos para alcanzar de manera progresiva la implementación de la metodología:

1. Capacitar al equipo de probadores y desarrolladores sobre herramientas de automatización de pruebas.
2. Capacitar al equipo de probadores sobre técnicas de diseño de casos de prueba
3. Involucrar y capacitar al personal de la empresa sobre los procesos de gestión de calidad incluidos en la propuesta de este documento.
4. Implementar la propuesta metodológica para gestión de calidad.
5. Recolectar y analizar los resultados
6. Mejorar los procesos en base a los resultados obtenidos.

#### **4.5.2 Plan de Implementación**

En el cuadro 31 incluye los objetivos, actividades, involucrados, responsable y tiempo estimado de las actividades que constituyen el plan de implementación para la metodología propuesta en este documento.

Cuadro 31. Plan de Implementación de la propuesta de metodología gestión de calidad para GBSYS.SA.

Objetivo	Actividades	Involucrados	Responsables	Tiempo
1. Capacitar al equipo de desarrolladores y probadores sobre herramientas de automatización de pruebas.	<ol style="list-style-type: none"> <li>1. Buscar consultor experto que desarrolle contenido del curso</li> <li>2. Preparar material para las sesiones de capacitación</li> <li>3. Capacitar por medio de sesiones prácticas a desarrolladores y probadores sobre herramientas para automatización de pruebas</li> </ol>	<ul style="list-style-type: none"> <li>• Desarrolladores</li> <li>• Probadores</li> </ul>	<ul style="list-style-type: none"> <li>• Gerente de proyectos</li> <li>• Líder de pruebas</li> </ul>	2 Meses
2. Capacitar al equipo de probadores sobre técnicas de diseño de casos de prueba.	<ol style="list-style-type: none"> <li>1. Buscar un consultor experto que desarrolle el contenido y realice la capacitación</li> <li>2. Preparar material para el contenido del curso</li> <li>3. Realizar sesiones teóricas y prácticas con los probadores sobre técnicas de diseño de pruebas</li> </ol>	<ul style="list-style-type: none"> <li>• Probadores</li> </ul>	<ul style="list-style-type: none"> <li>• Gerente de proyectos</li> <li>• Líder de pruebas</li> </ul>	2 Meses

Objetivo	Actividades	Involucrados	Responsables	Tiempo
3. Involucrar y preparar al personal de la empresa sobre los procesos de gestión de calidad incluidos en la propuesta de este documento.	<ol style="list-style-type: none"> <li>1. Preparar material para la capacitación</li> <li>2. Buscar instrumentos para la evaluación del curso</li> <li>3. Realizar la sesiones teóricas de capacitación sobre la metodología propuesta en este documento</li> </ol>	<ul style="list-style-type: none"> <li>• Director de proyectos</li> <li>• Gerentes de proyectos</li> <li>• Equipo de proyecto</li> </ul>	<ul style="list-style-type: none"> <li>• Autor de la guía</li> </ul>	1 Mes
4. Implementar la propuesta metodológica para gestión de calidad.	<ol style="list-style-type: none"> <li>1. Preparar plan piloto e identificar el proyecto adecuado para implementar la metodología</li> <li>2. Implementar la metodología</li> <li>3. Definir sesiones de seguimiento dentro de plan piloto</li> </ol>	<ul style="list-style-type: none"> <li>• Equipo de proyecto</li> <li>• Gerentes de Proyecto</li> <li>• Directores de proyecto</li> </ul>	<ul style="list-style-type: none"> <li>• Autor de la guía</li> </ul>	6 Meses
5. Recolectar y analizar los resultados	<ol style="list-style-type: none"> <li>1. Definir las métricas para evaluar los resultados de la implementación</li> <li>2. Analizar y documentar los resultados</li> <li>3. Exponer a los gerentes de proyecto los resultados de la propuesta metodológica</li> </ol>	<ul style="list-style-type: none"> <li>• Autor de la guía</li> <li>• Gerentes de Proyecto</li> </ul>	<ul style="list-style-type: none"> <li>• Autor de la guía</li> </ul>	1 Mes



Objetivo	Actividades	Involucrados	Responsables	Tiempo
6. Mejorar los procesos con base en los resultados obtenidos.	1. Actualizar los procesos y herramientas de acuerdo al análisis obtenido de los resultados 2. Informar a los involucrados sobre los cambios	<ul style="list-style-type: none"> <li>Equipo de proyecto</li> <li>Gerentes de Proyecto</li> <li>Directores del proyecto</li> </ul>	<ul style="list-style-type: none"> <li>Autor de la guía</li> </ul>	1 Mes

Fuente: Elaboración propia

### 4.5.3 Capacitaciones

Para la implementación de la propuesta metodológica se deben planificar varias capacitaciones para los integrantes del equipo de proyecto e interesados con el fin que adquieran el conocimiento necesario para aplicar las técnicas y herramientas sugeridas, cabe aclarar que las capacitaciones enfocan su contenido dependiendo tipo de perfil del interesado, por ende, cada integrante recibirá la capacitación de acuerdo a su rol dentro del equipo del proyecto. A continuación, se muestra un plan con contenidos a abarcar para los objetivos que involucren la planificación de sesiones de capacitación.

- **Objetivo 1 Preparar al equipo de desarrolladores y probadores acerca de herramientas de automatización de pruebas.**

Se detalla en la tabla 32 el plan de capacitación con los temas sugeridos para las sesiones de herramientas de automatización.

*Cuadro 32. Plan de contenidos para capacitación de herramientas de automatización para GBSYS.SA*

Temas	Sesiones	Horas consumidas
Introducción a la automatización	8 sesiones de 8 horas	Desarrolladores = 64 horas
Junit		Probadores = 64 horas
Unit Test en Visual Studio		
Testlink		

Temas	Sesiones	Horas consumidas
Cumcuber	1 sesión de semana	
Jmeter		
Herramientas de pruebas estáticas		

Fuente: Elaboración propia.

- **Objetivo 2 Preparar al equipo de probadores sobre técnicas de diseño de casos de prueba.**

Se detalla en el cuadro 33, el plan de capacitación con los temas sugeridos para las sesiones de capacitación sobre el proceso de pruebas y técnicas de diseño de casos de prueba.

*Cuadro 33. Plan de contenidos para capacitación del proceso de calidad y técnicas de diseño de casos de prueba para GBSYS.SA*

Temas	Sesiones	Horas consumidas
Introducción a la calidad	8 sesiones de 8 horas 1 sesión de semana	Probadores = 64 horas
Principios de las pruebas		
Proceso de pruebas		
Niveles y tipo de pruebas		
Técnicas de diseño de caja negra		
Técnicas de diseño de caja blanca		
Herramientas de gestión de casos de prueba y gestión de incidencias		
Práctica para para la certificación de ISTQB		

Fuente: Elaboración propia.

- **Objetivo 3 Involucrar y preparar al personal de la empresa sobre los procesos de gestión de calidad incluidos en la propuesta de este documento.**

Se detalla en la tabla 34, el plan de capacitación con los temas sugeridos para las sesiones de capacitación sobre la propuesta de metodología de gestión de calidad de software.

*Cuadro 34. Plan de contenidos para capacitación de la metodología propuesta de gestión de calidad de software para GBSYS.SA.*

<b>Temas</b>	<b>Sesiones</b>	<b>Horas consumidas</b>
Proceso de pruebas propuesto en la metodología	4 sesiones de 4 horas 1 sesión de semana	Probadores = 24 horas Desarrolladores = 24 horas
Herramientas y plantillas propuestas		Gerente General = 24 horas
Responsabilidades de los roles		Gerente de proyectos = 24 horas
Evaluación de conocimientos		Desarrollador de la guía = 24 horas

Fuente: Elaboración propia.

## 1 CONCLUSIONES

- Desde la creación del departamento de calidad de software, no se ha implementado una metodología para la gestión de la calidad que estandarice los procesos, actividades y herramientas relacionados a esta área. Lo anterior origina atrasos en el proceso de calidad, aumento en los costos de los proyectos y daños a la reputación.
- Se logró identificar a través de datos históricos de proyecto y entrevista al equipo de calidad, que actualmente se realizan actividades y se utilizan herramientas definidas para el proceso de calidad, sin embargo, existe una carencia de otras alternativas que contribuyan a reducir los tiempos del proceso de revisión. Además, el proceso no incluye algunos tipos de prueba que son necesarias para una validación confiable del producto.
- Con el objetivo de identificar las causas del problema, se realizó un análisis causaefecto para posteriormente plantear las oportunidades y mejoras al proceso de gestión de la calidad de software.
- Se propone una metodología de gestión de calidad de software que involucra los procesos de planificación, aseguramiento y control de la calidad con el proceso de pruebas y sus actividades. Además, brinda herramientas y documentación necesarias para la obtención de los objetivos de calidad.
- El plan de implementación plantea cinco objetivos para la puesta en marcha de la metodología de gestión de calidad, es vital las capacitaciones a los interesados para el éxito de la implementación.

## 2 RECOMENDACIONES

- La gerencia debe implementar una estrategia de calidad para motivar a los involucrados no solo a la adopción de la metodología, sino también a la disposición de adquirir el conocimiento y aplicar las técnicas y herramientas en sus tareas sin delegar las responsabilidades únicamente al equipo de control y aseguramiento de la calidad.
- En el análisis del proceso de gestión de calidad se pudo constatar que actualmente se consume tiempo considerable en la regresión de pruebas originado por un constante cambio en el alcance. Se recomienda aplicar medidas para la gestión de cambios y revisiones de los requerimientos en etapas tempranas del proyecto para evitar consumir tiempos en actualizar los casos de prueba y pruebas de regresión.
- Para las capacitaciones de los involucrados se define una lista de temas, los cuales requiere la selección de un asesor experto que transfiera conocimiento a los involucrados.
- La gerencia podría considerar el aprovechamiento del plan de implementación y capacitaciones para certificar a todos los integrantes del equipo de aseguramiento y calidad en las certificaciones de Internacional Software Testing Qualifications, las cuales pueden aportar valor al negocio.

### 3 BIBLIOGRAFIA

- CCM. (2017, noviembre). Ciclo de vida del 'software' [web log post]. Recuperado de <http://es.ccm.net/contents/223ciclodevidadelsoftware>.
- Chappell, D. (2011). THE THREE ASPECTS OF SOFTWARE QUALITY. Recuperado de [http://www.davidchappell.com/writing/white\\_papers/The\\_Three\\_Aspects\\_of\\_Software\\_Quality\\_v1.0Chappell.pdf](http://www.davidchappell.com/writing/white_papers/The_Three_Aspects_of_Software_Quality_v1.0Chappell.pdf)
- Chappell, D. (2011). THE BUSINESS VALUE OF SOFTWARE QUALITY. Recuperado de [http://www.davidchappell.com/writing/white\\_papers/The\\_Business\\_Value\\_of\\_Software\\_Qualityv1.0Chappell.pdf](http://www.davidchappell.com/writing/white_papers/The_Business_Value_of_Software_Qualityv1.0Chappell.pdf)
- Black, R & Mitchell, L (2011). Advanced Software Testing Vol. 3: Guide to the ISTQB Advanced Certification as an Advanced Technical Test Analyst. USA. Rocky Nook
- GBSYS. (2015). Global Business System – GBSYS. Recuperado de <https://gbsys.com/nosotros/>
- Gray, C.F & Larson, E.W. (2009). Administración de Proyecto. México. McGrawHill.
- Eyssautier, M. (2006). Metodología de la Investigación. Desarrollo de la inteligencia. (5ªed). México, Thomson Editores.
- Espejo, C.A., Bayona, O.S. & Pastor, C. (2016). Aseguramiento de la Calidad en el Proceso de Desarrollo de Software utilizando CMMI, TSP y PSP. doi: 10.17013/risti.20.6277
- IEEE (2008). IEEE Standard for Software and System Test Documentation. IEEE Std 8292008.
- Institute Software Engineering. (2010). CMMI® para Desarrollo, Versión 1.3. Recuperado de <https://www.sei.cmu.edu/library/assets/whitepapers/Spanish%20Technical%20Report%20CMMI%20V%201%203.pdf>
- Internacional Software Testing Qualifications Board (2011). Foundation Level Syllabus. Recuperado de <https://www.istqb.org/downloads/send/2foundationleveldocuments/3foundationlevellsyllabus2011.html>

- Project Management Institute Inc. (2013). Guía de los fundamentos para la dirección de proyectos (Guía PMBOK). Pennsylvania. Project Management Institute.
- ISO/IEC (2001). ISO/IEC 9126. Software engineering Product quality. ISO/IEC.
- LUNGU, I. (2017, 01 de Agosto). Integrating Quality Management System into Software Development Processes. Assistsoftware.net. Recuperado de <https://assistsoftware.net/blog/integratingqualitymanagementsystemssoftwaredevelopmentprocesses>
- Mari, J.A. (2002). Manual de redacción científica. Quinta edición. Publicación especial N° 3, Caribbean Journal of Science.
- Martín, A. V. (1995). Fuentes de información general. Gijón: Ediciones Trea.
- Osorio, B.N & Castro, L.G. (2011) Management of quality in software development. Revista de Investigación de Sistemas e Informática 8(1),6569 Recuperado de <http://revistasinvestigacion.unmsm.edu.pe/index.php/sistem/article/viewFile/5642/4889>
- Schwaber, K.&Sutherland. (2017). La Guía de Scrum. Recuperado de <http://www.scrumguides.org/docs/scrumguide/v2017/2017ScrumGuideSpanishSouthAmerican.pdf#zoom=100>
- Torres, Z y Torres, H. (2014). Administración de proyectos. México: Patria.

## 4 ANEXOS

### Anexo 1: ACTA DEL PFG

<b>ACTA DEL PROYECTO</b>	
<b>Fecha</b>	<b>Nombre del proyecto</b>
06/11/2017	Propuesta de metodología de la gestión de calidad para los proyectos de desarrollo de software en la empresa GBSYS.SA
<b>Áreas de conocimiento / procesos:</b>	<b>Área de aplicación (Sector / Actividad):</b>
Grupos de Procesos: <ul style="list-style-type: none"> <li>• Planeación</li> <li>• Ejecución</li> <li>• Seguimiento y control</li> </ul> Áreas de Conocimiento: <ul style="list-style-type: none"> <li>• Calidad</li> </ul>	Área de tecnologías de información y desarrollo de software.
<b>Fecha de inicio del proyecto</b>	<b>Fecha tentativa de finalización del proyecto</b>
06/11/2017	15/07/2018
<b>Objetivos del proyecto (general y específicos)</b>	
<p><b>Objetivo general</b></p> <p>Desarrollar una propuesta de una metodología para la gestión de la calidad, de tal manera que se estandaricen las prácticas y procesos asociados al control de la calidad en los productos de software desarrollados por la empresa.</p> <p><b>Objetivos específicos</b></p> <ol style="list-style-type: none"> <li>1. Determinar la situación actual y deficiencias de la empresa GBSYS.SA en el área de gestión de la calidad.</li> <li>2. Realizar una comparación de la situación actual de empresa en gestión de calidad</li> </ol>	



## ACTA DEL PROYECTO

de software respecto a los procesos y técnicas de calidad descritas en la norma para software IEEE 829, el estándar ISO/IEC 9126 y en el Foundation Level Syllabus de ISTQB.

3. Determinar las oportunidades de mejora en la gestión de la calidad de software tomando como fuente el análisis realizado.
4. Diseñar una guía de los procesos y técnicas para la gestión de la calidad de software que se ajusten a las necesidades de la empresa considerando el análisis elaborado
5. Crear un plan de capacitación para los involucrados para transmitir en conocimiento sobre la propuesta de metodología en gestión de calidad de software y puedan aplicarla en sus proyectos.

### **Justificación o propósito del proyecto (Aporte y resultados esperados)**

A principios del 2017, la empresa GBSYS.SA inauguró un departamento de aseguramiento de la calidad (QA) el cual está compuesto de 8 profesionales de calidad, a pesar de los esfuerzo de implementar conceptos de calidad, todavía no se ha realizado una metodología que alinea los procesos realizados en cada proyecto , uno de los objetivos de este proyecto es realizar un investigación sobre las mejores práctica en el área de desarrollo de software para crear una metodología que sea la base de los procesos y procedimientos relacionados a la gestión de la calidad.

Beneficios esperados

- Mejorar los procesos de planeación, diseño y ejecución de calidad de software.
- Reducir los tiempos de revisiones de sistemas por parte de los equipos de QA.
- Disminuir el número de incidencias reportadas por el usuario final.
- Aumentar la confiabilidad de los productos entregados al usuario final.

### **Descripción del producto o servicio que generará el proyecto – Entregables finales del proyecto**

El documento final de este proyecto es la propuesta de una metodología que detalla los procesos y procedimientos para realizar la gestión de la calidad de los proyectos de

## ACTA DEL PROYECTO

desarrollo de software en los que el equipo de aseguramiento de la calidad (equipo de QA) se integre. El documento final contiene las plantillas en cada proceso y las herramientas que se deben utilizar para realizar una gestión de la calidad que se adecue a las mejores prácticas.

- Documento con la recopilación de información sobre mejores prácticas en temas de calidad para los procesos de software
- Diseño de procedimientos y procesos propuestos para la metodología de gestión de calidad.
- Plantillas para las técnicas propuestas para los procedimientos.
- Plan de capacitación para los interesados.

### Supuestos

- El tiempo de desarrollo del proyecto final es de 4 meses.
- Existe un gran interés de la gerencia de la empresa en este proyecto.
- Los costos de proyectos son asumidos por la empresa patrocinadora.

### Restricciones

1. El objetivo de documento es definir un conjunto de prácticas, técnicas, procedimientos y normas usadas para quienes trabajan en del departamento de aseguramiento y control de calidad. Por ello, este documento es limitándose en la gestión de la calidad con el fin de profundizar en el área debido a las características que posee proyecto de desarrollo de software.
2. La metodología propuesta para calidad debe adaptarse a los procesos de desarrollo de software que posee la empresa.
3. Terminar el desarrollo del documento final del proyecto al inicio de julio del 2018.

### Identificación riesgos

1. Si existe una dedicación parcial del estudiante al desarrollo del proyecto por motivos laborales y personales puede producir atrasos en el cronograma
2. Si existe un faltante de información o confusión debido al intentar implementar varios estándares para la metodología propuesta puede afectar el alcance del

## ACTA DEL PROYECTO

proyecto.

3. Si existe un plazo limitado para realizar las correcciones en el documento debido a que el tutor no envía las correcciones en los plazos indicados puede producir atrasos en el cronograma.

### Presupuesto

- \$330 en refrigerios de capacitación para 25 personas.
- \$2050 honorarios asesor experta primera capacitación
- \$2050 honorarios asesor experta segunda capacitación
- \$760 honorarios asesor experta tercera capacitación

### Información histórica relevante

Global Business System (GBSYS S.A.) inició operaciones el 16 de julio de 1987, agrupando las actividades particulares de consultoría de un grupo de profesionales en computación, para dedicarse al desarrollo de sistemas de información en ambientes de bases de datos, contando para tal efecto con la colaboración de un selecto grupo de profesionales en diferentes especialidades de la informática. Hoy en día es una empresa especializada en Administración de Bases de Datos ORACLE y SQL Server además del desarrollo, mantenimiento y migración de aplicaciones en plataformas Developer de Oracle, Java y .Net. Adicionalmente cuenta con una línea de distribución de productos de software especializados incluyendo productos propios y otros ofrecidos por casas extranjeras reconocidas mundialmente. También se ofrecen servicios de tercerización (outsourcing) ya sea en soporte técnico para administración de sistemas y bases de datos, así como para desarrollo de aplicaciones. Documentación de esfuerzos similares anteriores

### Principales Hitos y fecha

Nombre hito	Fecha Inicio	Fecha Fin
Descripción de roles, procesos, herramientas utilizados por la empresa GBSYS.SA que colaboran a la gestión de calidad.	13/04/2018	30/04/2018

<b>ACTA DEL PROYECTO</b>		
Análisis de las deficiencias actuales en el área de gestión de calidad.		
Cuadros comparativos de los procesos, técnicas de diseño, pruebas, documentación y herramientas utilizadas por la empresa GBSYS y técnicas de calidad descritas en las normas, estándares y marcos metodológicos conocidos para software	31/04/2018	08/05/2018
Lista de necesidades y oportunidades de mejora en los procesos de gestión de calidad utilizados por la empresa	08/05/2018	15/05/2018
Propuesta de metodología para la gestión de calidad software para la empresa GBSYS.SA	15/05/2018	25/05/2018
Plan de capacitación para los interesados para retroalimentar sobre la propuesta de la metodología de gestión de calidad.	25/05/2018	06/05/2018
<b>Identificación de grupos de interés (involucrados)</b>		

## ACTA DEL PROYECTO

Involucrados Directos:

- Director de proyecto
- Gerente de proyectos de GBSYS.SA
- Gerente General de GBSYS.SA
- Profesionales de calidad y aseguramiento

Involucrados Indirectos:

- Directores de proyectos de la empresa.
- Integrantes del equipo de desarrollo de sistemas

**Director de proyecto:**

**María Angélica Villalobos Acuña**

**Firma:**

**Autorización:**

**Firma:**

## Anexo 2: EDT

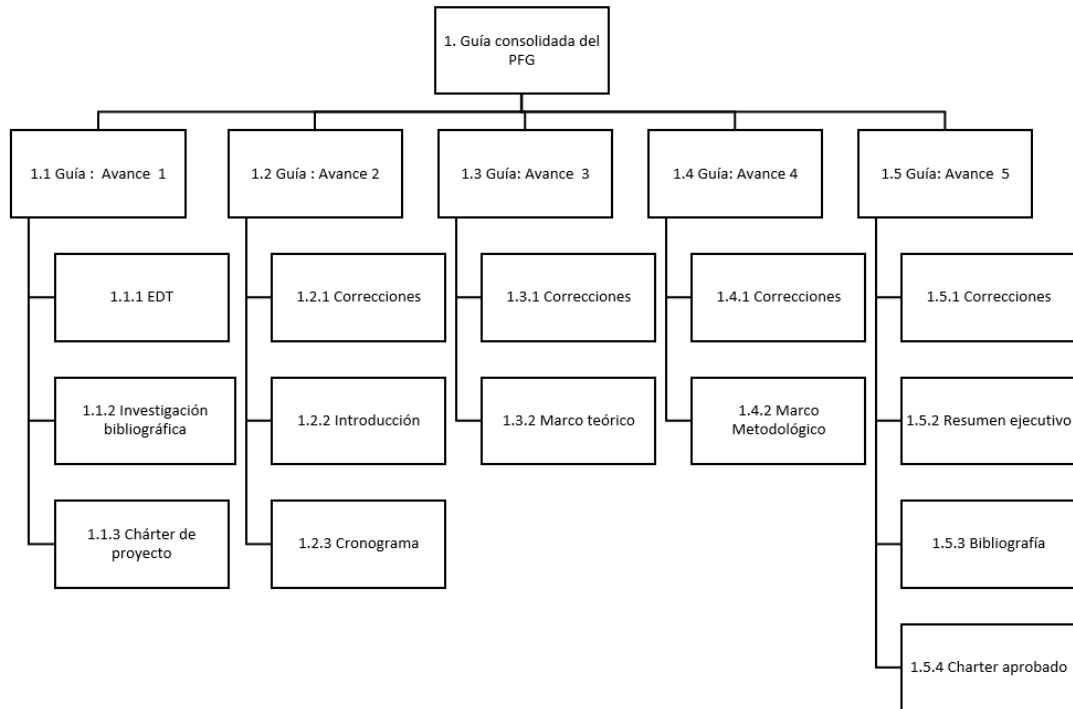


Figura 22. EDT del seminario del proyecto de graduación.

Fuente: Elaboración propia.

### Anexo 3: CRONOGRAMA

	WBS	Task Mode	Task Name	Dur:	Start	Finish	Predecessors
1	1		▲ Guía consolidada del PFG	65 hrs	Mon 06/11/17	Sun 10/12/17	
2	1.1		▲ Guía: Avance 1	13 hrs	Mon 06/11/17	Sun 12/11/17	
3	1.1.1		Elaboración Del EDT	4 hrs	Mon 06/11/17	Sat 11/11/17	
4	1.1.2		Elaboración de la Investigación bibliográfica	4 hrs	Sat 11/11/17	Sun 12/11/17	3
5	1.1.3		Elaboración del Charter del proyecto	5 hrs	Sun 12/11/17	Sun 12/11/17	4
6	1.1.4		Entrega del documento Actualizado	0 days	Sun 12/11/17	Sun 12/11/17	5
7	1.2		▲ Guía: Avance 2	13 hrs	Mon 13/11/17	Sun 19/11/17	
8	1.2.1		Correcciones	4 hrs	Mon 13/11/17	Sat 18/11/17	6
9	1.2.2		Realizar la Introducción	2 hrs	Sat 18/11/17	Sun 19/11/17	8
10	1.2.3		Realizar el cronograma	7 hrs	Sun 19/11/17	Sun 19/11/17	9
11	1.2.4		Entrega del documento Actualizado Avance 2	0 hrs	Sun 19/11/17	Sun 19/11/17	10
12	1.3		▲ Guía: Avance 3	13 hrs	Mon 20/11/17	Sun 26/11/17	
13	1.3.1		Correcciones	3 hrs	Mon 20/11/17	Thu 23/11/17	11
14	1.3.2		Elaboración del Marco teórico	10 hrs	Sat 25/11/17	Sun 26/11/17	13
15	1.3.3		Entrega del documento Actualizado Avance 3	0 hrs	Sun 26/11/17	Sun 26/11/17	14
16	1.4		▲ Guía: Avance 4	13 hrs	Mon 27/11/17	Sun 03/12/17	
17	1.4.1		Correcciones	3 hrs	Mon 27/11/17	Thu 30/11/17	14
18	1.4.2		Elaboración del Marco Metodológico	10 hrs	Sat 02/12/17	Sun 03/12/17	17
19	1.4.3		Entrega del documento Actualizado Avance 4	0 hrs	Sun 03/12/17	Sun 03/12/17	18
20	1.5		▲ Guía: Avance 5	13 hrs	Mon 04/12/17	Sun 10/12/17	
21	1.5.1		Correcciones	3 hrs	Mon 04/12/17	Thu 07/12/17	19
22	1.5.2		Resumen Ejecutivo	4 hrs	Sat 09/12/17	Sun 10/12/17	21
23	1.5.3		Bibliografía	3 hrs	Sun 10/12/17	Sun 10/12/17	22
24	1.5.4		Charter aprobado	3 hrs	Sun 10/12/17	Sun 10/12/17	23
25	1.5.5		Entrega del documento Actualizado Avance 5	0 hrs	Sun 10/12/17	Sun 10/12/17	24

Figura 23. Cronograma del seminario del proyecto de graduación.

Fuente: Elaboración propia.

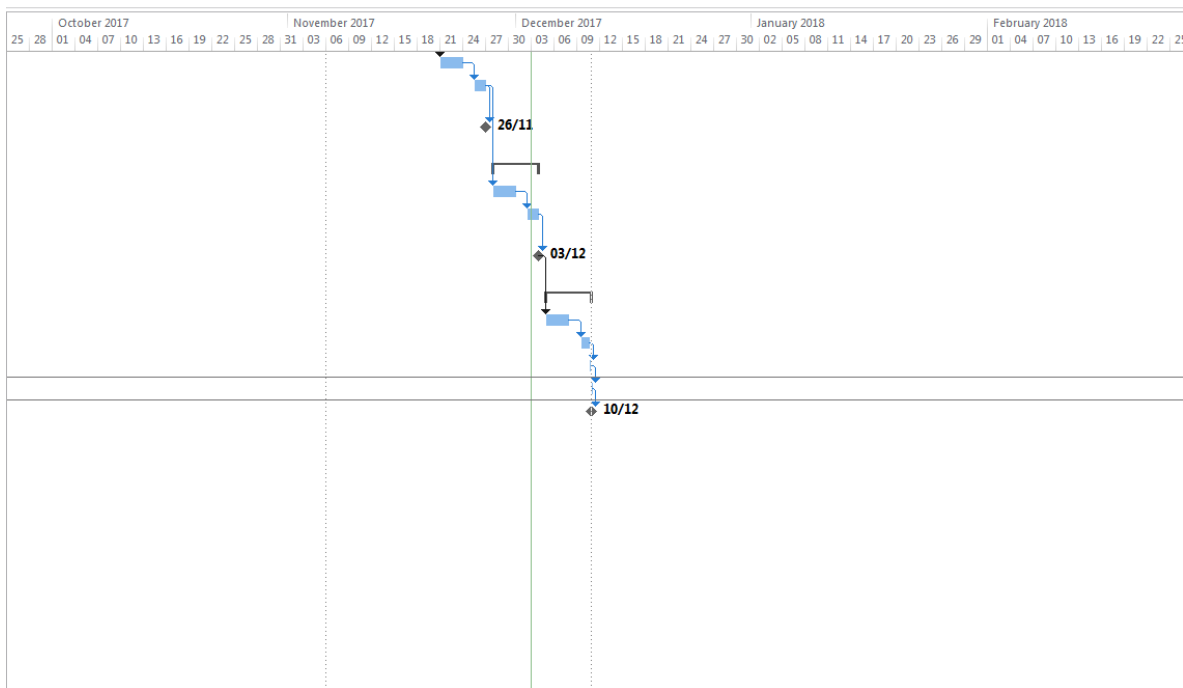


Figura 24. Diagrama de Gantt del Seminario de graduación.

Fuente: Elaboración propia.



## **Anexo 4: Entrevista sobre el proceso de gestión de calidad de la empresa GBSYS S.A.**

### **I. Introducción**

Las siguientes preguntas son parte de la entrevista que fue realizada al Ing. Luis Emilio Ramírez, el cual es gerente del departamento de proyectos de GBSYS.SA para conocer la situación actual del proceso de gestión de calidad en los proyectos de desarrollo de software en la empresa.

1. ¿Cuántos empleados conforman el departamento de calidad de software?
2. ¿Cuál es la razón por la cual la empresa decidió incluir un equipo de calidad en los proyectos?
3. ¿Cuáles han sido los resultados con la experiencia que ha tenido en el transcurso de estos años desde que incluyeron la gestión de calidad en los proyectos?

### **Sobre el proceso de gestión de calidad de Software**

4. ¿Actualmente, como se planifica la gestión de calidad en los proyectos?
5. ¿Se realiza un plan de pruebas de calidad del proyecto que especifique los objetivos de las pruebas, enfoque, criterios de salida y herramientas a utilizar?
6. ¿Existe un documento que determine cuáles las actividades de aseguramiento y control de calidad?
7. ¿Se define un documento de estándares visuales y de programación antes de iniciar los procesos de gestión de calidad?
8. ¿Existe un cronograma único para las tareas del equipo de QA?

9. ¿Se establece fechas para la entrega de informes de avance de prueba y métricas al administrador de proyectos?
10. ¿Se realizan actividades de cierre tales como cerrar los reportes de incidencias, almacenar información histórica de las pruebas y documentar las acciones aprendidas?
11. ¿Se realizan inspecciones en los proyectos para verificar que se cumple con los procesos de calidad estipulados?

## **Anexo 5: Encuestas de sobre el proceso de pruebas, técnicas y herramientas utilizadas por el equipo de calidad de software**

### **Encuestas de sobre el proceso de pruebas, técnicas y herramientas utilizadas por el equipo de calidad de software GBSYS S.A.**

Este instrumento es parte del proyecto final de graduación “Propuesta de una metodología para la gestión de la calidad de software para los proyectos de desarrollo de software en la empresa GBSYS S.A”.

El objetivo de esta entrevista es conocer aspectos sobre el proceso de pruebas, técnicas y herramientas utilizadas.

Agradezco de antemano su valiosa colaboración.

#### **Instrucciones**

De acuerdo a su experiencia, responda las siguientes preguntas

¿Conoce las tareas que realiza el rol de Tester?

---

---

¿Realizan una reunión previa para seleccionar las condiciones de prueba?

---

---

¿Indique los tipos de pruebas y técnicas son utilizadas para la confección de los casos de prueba?

---

---

¿Realiza el diseño de pruebas automatizadas para la revisión de los objetos de prueba?

---

---

¿Utilizan alguna herramienta o técnica que ayude a realizar las revisiones?

---

---

---

¿Diseñan y ejecutan casos de prueba no funcionales? ¿Indique cuales tipo de prueba no funcionales son realizados?

---

---

¿Utiliza los Tester técnicas para la estimación de sus tareas?

---

---