



UNIVERSIDAD  
**SAN IGNACIO  
DE LOYOLA**

**FACULTAD DE INGENIERÍA**

**Carrera Ingeniería Empresarial y de Sistemas**

**IMPLEMENTACIÓN DE ISO/IEC 12207:2008 PARA LA  
MEJORA DEL PROCESO DE DESARROLLO DE  
SOFTWARE: CASO TCI**

**Tesis para optar el Título Profesional de Ingeniería Empresarial y  
de Sistemas**

**BERNAL PERALTA, MARICELA**

**HERMOSILLA JARA, JENNY**

**Asesor:**

**Salcedo Huarcaya, Marco Antonio**

**Lima – Perú**

**2018**

## JURADO DE LA SUSTENTACION ORAL

.....  
Zalatiel, Carranza Avalos  
Presidente

.....  
Angela Teresa, Barreda Ramírez  
Jurado 1

.....  
Jose Carlos, Benitez Palacios  
Jurado2

---

**Entregado el:** \_\_\_\_\_

**Aprobado por:**

.....  
Maricela Bernal Peralta  
DNI N° 10153382

.....  
Ing. Marco Antonio Salcedo Huarcaya  
DNI N° 08155491

.....  
Jenny Hermosilla Jara  
DNI N° 40594015

**UNIVERSIDAD SAN IGNACIO DE LOYOLA**  
**FACULTAD DE INGENIERIA**

**DECLARACIÓN DE AUTENTICIDAD**

Nosotras; Maricela Bernal Peralta, identificada con DNI N° 10153382 y Jenny Hermosilla Jara, identificada con DNI N° 40594015 ambas Bachilleres del Programa Académico de la Carrera de Ingeniería Empresarial y de Sistemas de la Facultad de Ingeniería de la Universidad San Ignacio de Loyola, presentamos nuestra tesis titulada: "Implementación De ISO/IEC 12207:2008 Para La Mejora Del Proceso De Desarrollo De Software: Caso TCI".

Declaramos en honor a la verdad, que el trabajo de tesis es de nuestra autoría; los datos, resultados, su análisis e interpretación, constituyen nuestro aporte. Todas las referencias han sido debidamente consultadas y reconocidas en la investigación.

En tal sentido, asumimos la responsabilidad que corresponda ante cualquier falsedad u ocultamiento de la información aportada. Por las afirmaciones manifestadas, ratificamos lo expresado, a través de nuestras firmas correspondientes.

Lima, Diciembre del 2017

.....

Maricela Bernal Peralta  
DNI N° 10153382

.....

Jenny Hermosilla Jara  
DNI N° 40594015

## ÍNDICE

ÍNDICE DE TABLAS.....	1
ÍNDICE DE FIGURAS.....	2
INDICE DE ANEXOS.....	3
DEDICATORIA.....	4
AGRADECIMIENTOS.....	5
RESUMEN.....	6
ABSTRACT.....	7
INTRODUCCIÓN.....	8
<b>PROBLEMA DE INVESTIGACIÓN.....</b>	<b>9</b>
<b>Identificación del Problema.....</b>	<b>9</b>
<b>Formulación del Problema.....</b>	<b>9</b>
<b>Justificación de la Investigación.....</b>	<b>23</b>
<b>MARCO REFERENCIAL.....</b>	<b>23</b>
<b>Antecedentes.....</b>	<b>23</b>
<b>Antecedentes internacionales.....</b>	<b>23</b>
<b>Antecedentes nacionales.....</b>	<b>25</b>
<b>Estado del arte.....</b>	<b>26</b>
<b>Marco Teórico.....</b>	<b>26</b>
<b>La industria del software.....</b>	<b>26</b>
<b>ISO/IEC 12207:2008.....</b>	<b>27</b>
<b><i>Propósito.....</i></b>	<b><i>27</i></b>
<b><i>Limitaciones.....</i></b>	<b><i>28</i></b>
<b><i>Conformidad.....</i></b>	<b><i>28</i></b>
<b><i>Conformidad Completa.....</i></b>	<b><i>28</i></b>
<b><i>Conformidad a la Medida.....</i></b>	<b><i>28</i></b>
<b><i>Recomendación de la “Norma ISO/IEC 12207:2008 de acuerdo a los procesos del ciclo de vida del software”.....</i></b>	<b><i>29</i></b>
<b>ISO 15504.....</b>	<b>32</b>
<b>CMMI.....</b>	<b>35</b>

Metodologías de Desarrollo.....	38
Metodologías de Desarrollo Tradicionales.....	39
<i>Rup.</i> .....	40
<i>Msf.</i> .....	45
Metodologías de desarrollo ágil.....	51
<i>Scrum.</i> .....	52
<i>Kanban.</i> .....	54
<i>Scrumban.</i> .....	56
<b>OBJETIVOS E HIPÓTESIS</b> .....	60
Objetivos.....	60
Justificación de la investigación.....	60
Matriz de Consistencia.....	63
<b>METODO Y MARCO METODOLOGICO</b> .....	64
Tipo de Investigación.....	64
Diseño de la Investigación.....	64
Variables.....	64
Variable Independiente.....	64
Variables Dependientes.....	64
Población y Muestra.....	65
Técnicas.....	65
Instrumentos.....	65
Procedimientos de recolección de datos.....	65
Método de análisis.....	65
<b>RESULTADOS</b> .....	66
<b>DISCUSIONES</b> .....	68
<b>CONCLUSIONES</b> .....	69
<b>RECOMENDACIONES</b> .....	70
REFERENCIAS.....	71
ANEXOS.....	75

**ÍNDICE DE TABLAS**

Tabla 1. <i>Portafolio de soluciones y cambios realizados.</i>	12
Tabla 2. <i>Cumplimiento de procesos ISO 12207 proceso 6.1.</i>	16
Tabla 3. <i>Cumplimiento de procesos ISO 12207 proceso 6.2.</i>	16
Tabla 4. <i>Cumplimiento de procesos ISO 12207 proceso 6.3.</i>	17
Tabla 5. <i>Cumplimiento de procesos ISO 12207 proceso 6.4.</i>	18
Tabla 6. <i>Cumplimiento de procesos ISO 12207 proceso 7.1.</i>	18
Tabla 7. <i>Cumplimiento de procesos ISO 12207 proceso 7.2.</i>	19
Tabla 8. <i>Cumplimiento de procesos ISO 12207 proceso 7.3.</i>	19
Tabla 9. <i>Niveles de madurez de CMMI.</i>	36
Tabla 10. <i>Niveles de Capacidad de CMMI.</i>	37
Tabla 11. <i>Comparativa entre los modelos CMMI vs. ISO 15504.</i>	37
Tabla 12. <i>Comparación de enfoque ágil vs el tradicional.</i>	39
Tabla 13. <i>Roles de RUP.</i>	45
Tabla 14. <i>Análisis comparativo RUP y MSF.</i>	50
Tabla 15. <i>Comparativa entre scrum, kanban, scrumban.</i>	59
Tabla 16. <i>Distribución de muestra.</i>	65

**ÍNDICE DE FIGURAS**

<i>Figura 1.</i> Incidencias en las aplicaciones de TCI.	12
<i>Figura 2.</i> Diagrama de solución factura electrónica.	14
<i>Figura 3.</i> Organigrama de TCI.	14
<i>Figura 4.</i> Matriz FODA.	21
<i>Figura 5.</i> Diagrama de Ichikawa.	22
<i>Figura 6.</i> Ciclo de vida de los procesos de software de ISO/IEC 12207:2008.	30
<i>Figura 7.</i> Estructura de los procesos de ISO/IEC 12207:2008.	31
<i>Figura 8.</i> Mejora de la calidad de los procesos software.	33
<i>Figura 9.</i> Niveles de madurez.	33
<i>Figura 10.</i> Mejora de procesos de software iso/iec 15504 – iso/iec 12207:2008.	34
<i>Figura 11.</i> RUP-Esfuerzo en actividades según fase del proyecto.	40
<i>Figura 12.</i> Actividades del despliegue	43
<i>Figura 13.</i> Iterativo e incremental (Rup).	44
<i>Figura 14.</i> Modelo de equipo de trabajo en msf.	47
<i>Figura 15.</i> Etapas o fases del modelo de proceso de MSF.	47
<i>Figura 16.</i> Ventajas de RUP.	50
<i>Figura 17.</i> Marco de trabajo SCRUM.	52
<i>Figura 18.</i> Tablero kanban.	55
<i>Figura 19.</i> Matriz de consistencia.	63
<i>Figura 20.</i> Dispersión de quejas e incidencias.	66

**INDICE DE ANEXOS**

<b>ANEXO 1.</b> Cronograma y recursos de la implementación de los procesos.	75
<b>ANEXO 2.</b> Actividades y plan de acción para la implementación del Proceso de pruebas de calificación de software.	77
<b>ANEXO 3.</b> Procedimiento establecido para el cumplimiento del Proceso de Pruebas de Calificación del Software.	79
<b>ANEXO 4.</b> Plantilla de enunciado de alcance del proyecto.	85
<b>ANEXO 5.</b> Plantilla de estimación de pruebas.	87
<b>ANEXO 6.</b> Plantilla del Cronograma.	88
<b>ANEXO 7.</b> Plantilla del plan de pruebas.	89
<b>ANEXO 8.</b> Plantilla del informe de pruebas.	95
<b>ANEXO 9.</b> Resultado de la auditoria.	96
<b>ANEXO 10.</b> Diagrama AS-IS del Proceso de Pruebas de Calificación del Software.	98
<b>ANEXO 11.</b> Diagrama TO-BE del Proceso de Pruebas de Calificación del Software.	99
<b>ANEXO 12.</b> SubProceso del Proceso Calificación de Pruebas de Software.	100
<b>ANEXO 13.</b> SubProceso Gestionar Incidencias.	101
<b>ANEXO 14.</b> Proceso de Resolución de Problemas de Software.	102
<b>ANEXO 15.</b> Procedimiento del Proceso de Resolución de Problemas de Software.	103
<b>ANEXO 16.</b> Registro de Problemas categorizados por aplicación, prioridad, fecha y estado del Proceso de Resolución de Problemas de Software.	107
<b>ANEXO 17.</b> Informe del problema del Proceso de Resolución de Problemas de Software	108
<b>ANEXO 18.</b> Diagrama AS-IS del Proceso de Resolución de Problemas de Software.	109
<b>ANEXO 19.</b> Diagrama TO-BE del Proceso de Resolución de Problemas de Software.	110
<b>ANEXO 20.</b> Diagrama SubProceso del Proceso de Resolución de Problemas de Software.	111
<b>ANEXO 21.</b> Valorización del proyecto.	112
<b>ANEXO 22.</b> Estimación del ahorro utilizando el método de Fermi.	113

## DEDICATORIA

A mi esposo, quien me alentó y brindo su apoyo durante el desarrollo de esta Tesis. A mis padres por demostrarme siempre su cariño y apoyo. A Rodrigo, mi hijo, quien es mi gran motivador para seguir adelante con mis metas.

Jenny Hermosilla Jara

A mis padres, quienes han sido mis grandes pilares, por el gran apoyo que me brindaron. A mi esposo, que me aconseja a seguir desarrollándome profesionalmente.

Maricela Bernal Peralta

## **AGRADECIMIENTOS**

Estamos agradecidas a Dios y a nuestros padres por habernos guiado y acompañado en todo momento y darnos fortaleza en los momentos de debilidad y brindarnos el aprendizaje en base a su experiencia y sobre todo mucha felicidad. Agradecemos también a nuestra familia, ya que siempre encontramos un apoyo y consejos para nuestro bien.

## RESUMEN

### “IMPLEMENTACIÓN DE ISO/IEC 12207:2008 PARA LA MEJORA DEL PROCESO DE DESARROLLO DE SOFTWARE: CASO TCI”

#### **MARICELA BERNAL PERALTA**

Maricela.bernalp@gmail.com

#### **JENNY HERMOSILLA JARA**

jennyhermosillajara@hotmail.com

Esta tesis pretende implementar mejoras en el proceso de desarrollo de software de la empresa TCI, con el fin de acrecentar la calidad de las aplicaciones que forman parte del servicio Factura Electrónica, puesto que la baja calidad de las aplicaciones ha generado que la satisfacción de los clientes decaiga en el último año.

Las áreas de Desarrollo/Mantenimiento y Soporte de FE, presentan problemas debido a la falta de organización, mala definición de procesos, poca comunicación entre las áreas y por no contar con un área que se encargue de realizar pruebas de las aplicaciones.

El objetivo de la presente tesis es evaluar el impacto de la implementación de la norma “ISO/IEC 12207:2008” en el proceso de desarrollo de software con respecto a las incidencias, quejas y rotación de personal. Se encontró que la implementación disminuyó la media y dispersión de las tres variables.

**Palabras Clave:** Ciclo de vida de desarrollo de software, ISO/IEC 12207:2008, “Mejora del Proceso de Desarrollo de Software”, “Calidad de Software, Servicios de TI”.

## ABSTRACT

### **“IMPLEMENTATION OF ISO / IEC 12207: 2008 FOR IMPROVEMENT OF THE SOFTWARE DEVELOPMENT PROCESS: TCI CASE”**

This thesis aims to implement improvements in the software development process of the company TCI SA, in order to increase the quality of the applications that are part of the Electronic Invoice service, since the low quality of the applications has generated that the satisfaction of The customers declined in the last year.

The areas of Development / Maintenance and FE Support, present problems due to the lack of organization, poor definition of processes, little communication between the areas and not having an area that is responsible for testing applications.

The objective of this thesis is to asses the impact of the implementation of the standard "ISO / IEC 12207: 2008" on errors, incidents and staff turnover of the software development process. It was found that the implementation decreased the average of the three variables, also decreasing their dispersion.

**Keywords:** Software Development Lifecycle, “Software Development Process”, “ISO / IEC 12207”, “Agile Methodology”.

## INTRODUCCIÓN

En la actualidad existe un gran porcentaje de empresas pequeñas que fracasan en su intento de crecimiento, entre los factores que influyen se encuentran: La falta de planificación, no contar con objetivos a largo plazo; falta de análisis de riesgos potenciales; prevención de los mismos, no prever que no solo es malo no vender, sino también vender mucho.

Con respecto a la operación, los procesos productivos y de servicios no se encuentran definidos ni organizados, tienen altos niveles de desperdicios, altos costos por procedimientos repetidos, errores y falta de controles de calidad.

Por estos motivos se encuentra un creciente interés por la implementación de estándares que permitan mejorar su “proceso de desarrollo de software”. Obteniéndose resultados positivos y evidenciando una notable “Mejora en los procesos de ciclo de vida de software” que fue lo que encontramos como resultado de nuestra investigación.

En el presente proyecto, nosotras hemos optado por implementar una “mejora de proceso de desarrollo de software para la empresa TCI”, para ordenar, controlar, mejorar sus procesos de calidad con el fin de reducir falta de satisfacción de sus cliente por errores en el software.

En la primera parte de la tesis, mencionamos el problema de investigación donde contiene la identificación del problema, formulación del problema y justificación de la investigación.

En la segunda parte de la tesis, mencionamos el marco referencial, se detallan los antecedentes basados en investigaciones anteriores, tesis y demás información bibliográfica, marco teórico que contiene todos los conceptos a tener en cuenta para la comprensión de la presente tesis.

Finalmente, en la última parte se encuentran los resultados.

## **PROBLEMA DE INVESTIGACIÓN**

### **Identificación del Problema**

La baja satisfacción de un cliente para una empresa que brinda servicios es un gran problema teniendo en cuenta que, así como las recomendaciones llegan rápidamente también ocurre lo mismo con las quejas por el mal servicio recibido, lo cual es una desventaja para la empresa ya que abre el camino a los competidores. Por ello es importante identificar las razones que generan la baja satisfacción de los clientes y se pueda encontrar soluciones adecuadas, lo cual conllevará al crecimiento de la empresa.

Una de las razones por las cuales los clientes presentan baja satisfacción es porque no se sienten bien atendidos, lo cual muchas veces es reflejo de falta de definición de procesos y estándares de la empresa que le brinda servicios. Por ese motivo las empresas sobre todo en crecimiento tienden a organizarse, definir procesos, trabajar con estándares para mejorar el servicio al cliente.

En ese sentido, la mejora del proceso de desarrollo de software de la empresa TCI es crucial para lograr el incremento de los indicadores pertinentes para la organización. Luego de un análisis de la situación actual, se llegó a la conclusión que el rápido crecimiento organizacional y comercial de la empresa ha superado los procesos de supervisión y control actuales, requiriendo la implementación de guías estandarizadas para las operaciones de la empresa. Al haber implementado el ISO/IEC 12207:2008, la empresa desea verificar si la implementación tuvo el impacto deseado en los indicadores relevantes para la misma, siendo estos el número de incidencias, quejas y rotación de personal. Esto nos lleva a formular nuestra pregunta de investigación: ¿Cuál es el impacto de la implementación del ISO/IEC 12207:2008 en los resultados del proceso de desarrollo de software de TCI?.

### **Formulación del Problema**

#### **Problema General**

¿Cuál es el impacto de la implementación del ISO/IEC 12207:2008 en los resultados del proceso de desarrollo de software de TCI?

**Problemas específicos**

1. ¿Cuál es el impacto de la implementación de ISO/IEC 12207:2008 en el número de incidencias de la empresa TCI?
2. ¿Cuál es el impacto de la implementación de ISO/IEC 12207:2008 en el número de quejas de la empresa TCI?
3. ¿Cuál es el impacto de la implementación de ISO/IEC 12207:2008 en la rotación de personal de la empresa TCI?

**Descripción de la Empresa.**

TCI S.A. es una empresa peruana que brinda servicios de tecnologías de información, fue creada en el año 1993 con 10 trabajadores. Inicialmente brindaron servicios de correo electrónico inter-empresarial, formando así su propia red privada de comunicaciones.

En la actualidad se brindan los de servicios de facturación Electrónica y Soluciones B2B aduanero.

La empresa ofrece el servicio Factura Electrónica desde el año 2013, siendo actualmente el servicio estrella de la empresa.

A pesar de tener más de treinta competidores entre ellos seis empresas extranjeras, ha logrado captar hasta el momento 350 clientes y este número sigue en aumento. Cabe resaltar que este crecimiento hizo que se incremente su número de trabajadores de 10 a 90.

TCI, ha participado activamente en el proyecto Factura Electrónica de SUNAT desde que entro a vigencia la Ley de Factura Electrónica en el Perú, que fue en Julio del 2010. En noviembre del 2012 TCI se convierte en la Primera empresa autorizada como emisor electrónico en Perú según Resolución de Intendencia N° 0320050000001/SUNAT.

En Octubre del 2015 TCI obtiene, mediante Resolución de Intendencia N° 032-005-0000154/SUNAT – publicada el 22 de Octubre de 2015-la autorización para pertenecer al Registro de Proveedores de Servicios Electrónicos, por cumplir con las condiciones previstas en la normativa vigente (Resolución de

Superintendencia N° 199-2015/SUNAT), en la cual deben inscribirse los sujetos que desean prestar servicios a emisores electrónicos para la realización de todas las actividades inherentes a esta modalidad.

A continuación, en la tabla 1, nos muestra el portafolio de soluciones que ofrece TCI: mediante las aplicaciones y cambios realizados entre el año 2016 y 2017.

Tabla 1. *Portafolio de soluciones y cambios realizados.*

Solución	Aplicación	Puesta en producción	Cantidad clientes	Cambios realizados entre 2016 y 2017	
				Por normativa	Por requerimiento
B2B	Carganet	Mar-10	35	1	2
aduanero	SitransPlus	Jul-08	18	1	8
Factura	eGestor	Nov-13	350	45	36
Electrónica	ePortal	Nov-13	350	10	2

En la figura 1 se puede apreciar las estadísticas para el periodo de Enero 2015 a Diciembre 2017, el cual nos muestra el porcentaje de incidencias ocurridas en este lapso de tiempo, donde se puede observar que el porcentaje mayor está concentrándose en los servicios de Facturación electrónica (eGestor y ePortal).

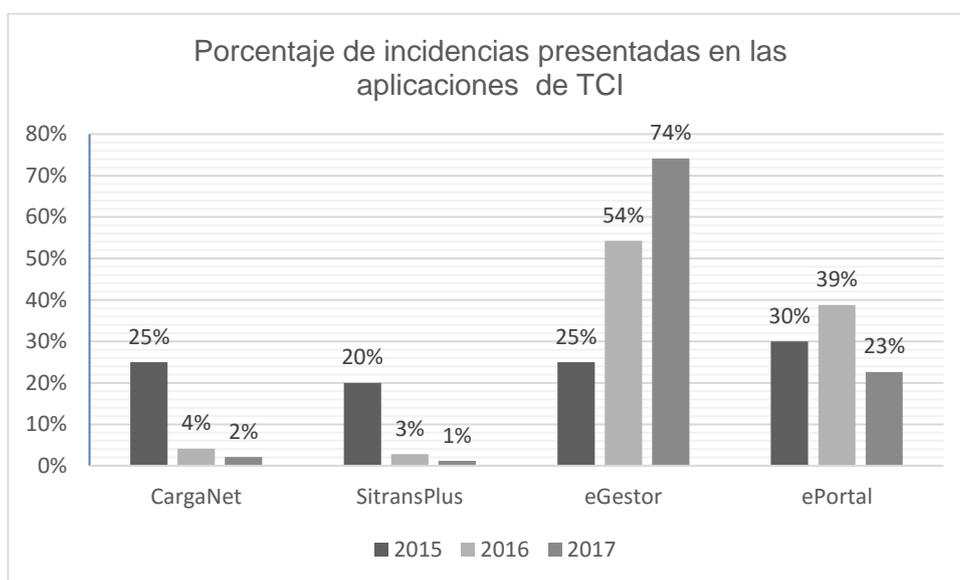


Figura 1. Incidencias en las aplicaciones de TCI.  
Fuente: Elaboración propia.

### **Servicio factura electrónica.**

Este servicio consiste en la emisión y recepción de comprobantes de pago de forma electrónica, siguiendo los estándares regidos por SUNAT.

Cada empresa puede desarrollar su propia solución para la transmisión de comprobantes electrónicos pero también pueden optar por contratar los servicios de un proveedor autorizado por SUNAT.

Cada proveedor ofrece distintos tipos de soluciones. El fin del servicio es brindarle todas las facilidades a las empresas para la integración y transmisión a SUNAT de los comprobantes de pago.

La SUNAT cada año publica un listado de empresas obligadas las cuales tienen un plazo para empezar a emitir comprobantes electrónicos.

La solución que brinda TCI está compuesta por los siguientes componentes:

**EGestor.** Son las aplicaciones que se instalan en el cliente, que son: Servicios Web, Servicios Windows y una aplicación de escritorio, a estas aplicaciones se les brinda constante mantenimiento por mejoras o por cambios en la norma de SUNAT.

El servicio web permite obtener la información de los comprobantes, para ello el cliente deberá consumir un método y enviar la información solicitada, el servicio se encarga de la generación del archivo XML y firma.

Los servicios Windows se encargan de la transmisión a SUNAT y del envío correos electrónicos con las representaciones impresas y archivos XML.

**ePortal.** Está compuesto por un servicio web que se encarga de la recepción de documentos del eGestor, registro en base de datos y transmisión a SUNAT.

Cuenta con un Portal donde los usuarios pueden visualizar sus comprobantes transmitidos.

En la figura 2, se puede apreciar la representación del Diagrama de Solución Factura Electrónica.

# Solución Factura Electrónica

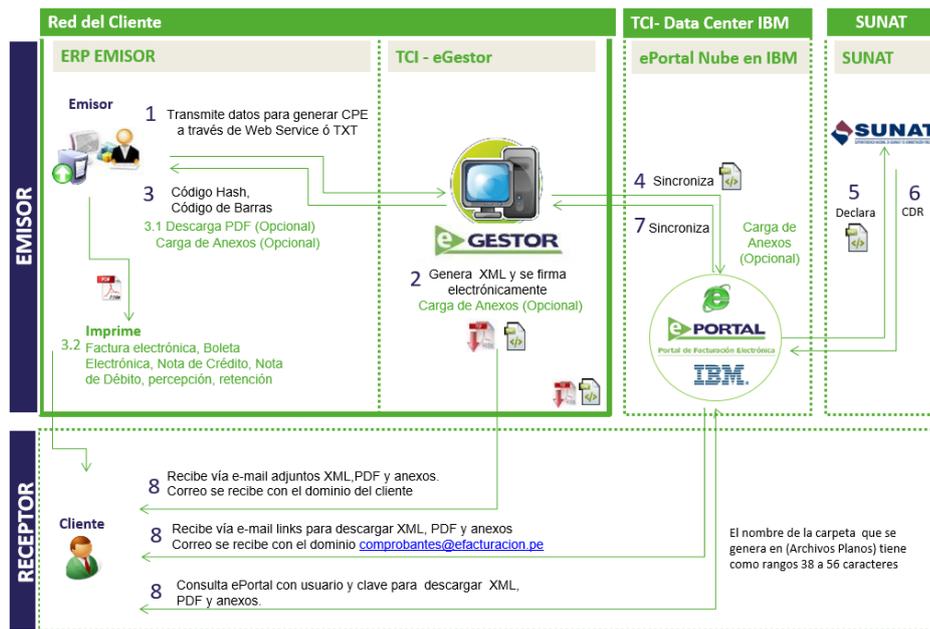


Figura 2. Diagrama de solución factura electrónica. Fuente. Elaboración propia/.

## Estructura Organizacional.

En la figura 3, podemos visualizar la estructura organizacional de TCI, que está compuesta de la siguiente manera:

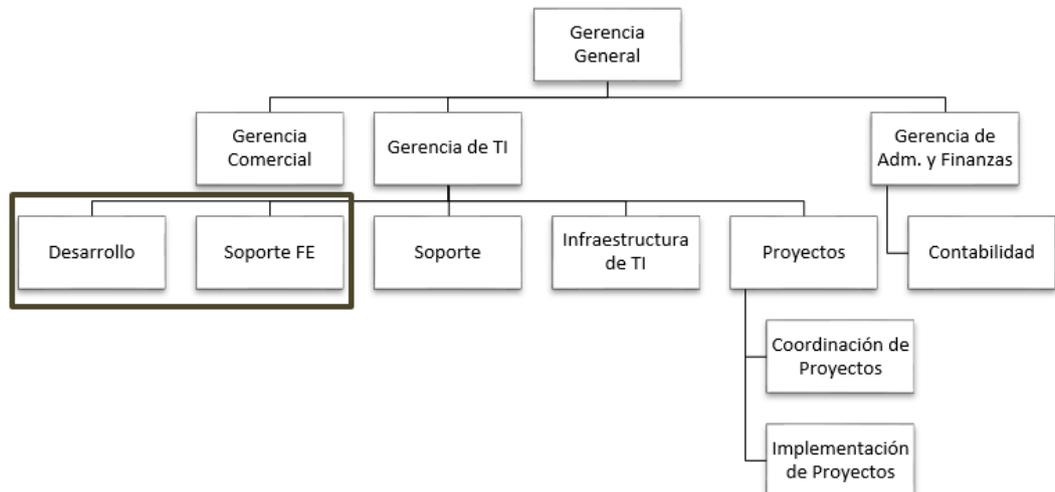


Figura 3. Organigrama de TCI. Fuente: Elaboración propia.

## Área Funcional.

TCI cuenta con dos áreas Desarrollo/Mantenimiento y Soporte FE que es el área que brinda soporte de segunda línea.

El equipo de Desarrollo/Mantenimiento cuenta con cinco analistas programadores y un líder.

El equipo de Soporte FE cuenta con cinco analistas programadores y un líder.

A continuación se describirá cada uno de los procesos de cada área y los problemas que están generando la baja satisfacción a los clientes.

### ***Proceso del área de desarrollo y mantenimiento.***

Este equipo se encarga de desarrollar nuevos módulos en las aplicaciones eGestor y ePortal que corresponden a los nuevos documentos normados por SUNAT, así como de brindar mantenimiento a los módulos existentes.

Actualmente trabajan de la siguiente manera (véase el anexo 10):

Gerencia de TI reciben el requerimiento de la Gerencia General para el análisis de un nuevo tipo de comprobante electrónico normado por SUNAT o por alguna mejora.

El líder del equipo realiza el análisis del nuevo comprobante electrónico. El líder de equipo es la persona que más conoce la solución.

Elabora un documento donde registra los requerimientos y elabora un cronograma, donde indica el tiempo y cantidad de recursos que trabajaron en el proyecto.

Una vez aprobado el cronograma se inicia el desarrollo. Cada desarrollador se encarga de realizar las pruebas de lo que desarrolló.

Finalizados sus cambios los suben al controlador de versiones.

Se observa que trabajan de manera empírica, no realizan las pruebas adecuadas porque los mismos desarrolladores las realizan, no se especifica un estándar de documentación ni repositorio. No existe comunicación con el equipo de Soporte FE. Existe dependencia de personal.

A continuación, se muestra tablas del 2 al 8, en donde indican que se cumplen con “los procesos de la ISO/IEC 12207:2008”; para esto se tomará como referencia uno de los proyectos desarrollados en este año 2017.

### Procesos ISO 12207 en el desarrollo de un proyecto

Proyecto: Desarrollo del módulo Guías de Remisión.

Fecha Inicio: 15/02/2017

Fecha Inicio: 31/05/2017

Tabla 2. *Cumplimiento de procesos ISO 12207 proceso 6.1.*

Organización de Procesos		Cumplimiento	Comentario	
Procesos de Contexto del Sistema	6.1.1 Proceso de Adquisición	Preparación para la adquisición.	Si cumple	
		Divulgación de la adquisición.	Si cumple	
		Selección del proveedor.	Si cumple	Se ha comprado componentes para usarlos en el desarrollo de las aplicaciones
		Acuerdo del contrato.	Si cumple	
		Seguimiento del acuerdo	Si cumple	
		Aceptación por el adquiriente	Si cumple	
	Cierre.	Si cumple		
	6.1.2 Proceso de Suministro	Identificación de la oportunidad.	Si cumple	
		Licitación del proveedor.	Si cumple	
		Acuerdo de contrato.	Si cumple	
Ejecución del contrato.		Si cumple		
	Entrega y soporte del producto/servicio.	Si cumple		
	Cierre.	Si cumple		

Tabla 3. *Cumplimiento de procesos ISO 12207 proceso 6.2.*

Organización de Procesos		Cumplimiento	Comentario	
Procesos de Contexto del Sistema	6.2.1 Proceso de Gestión del Modelo de Ciclo de Vida	Establecimiento del proceso.	No cumple	Cada equipo maneja un gestión distinta
		Evaluación del Proceso.	No cumple	
		Mejora del proceso.	No cumple	
	6.2.2 Proceso de Gestión de la Infraestructura	Implementación del proceso.	Si cumple	Si se realiza pero no existe documentación de ello
		Establecimiento de la infraestructura.	Si cumple	
		Mantenimiento de la infraestructura.	Si cumple	
	6.2.3 Proceso de Gestión del Portafolio del Proyecto	Inicio del proyecto.	No cumple	El presupuesto es manejado por la Gerencia General. Cada proyecto se maneja de manera diferente sin un estándar.
		Evaluación del portafolio.	No cumple	
		Cierre del proyecto.	No cumple	
	6.2.4 Proceso de Gestión de los Recursos Humanos	Identificación de habilidades.	No cumple	No se tiene registros de evaluación de desempeño, entrenamientos realizados
		Desarrollo de habilidades.	No cumple	
		Adquisición y provisión de habilidades.	No cumple	
		Gestión del conocimiento.	No cumple	
	6.2.5 Proceso de Gestión de la Calidad	Gestión de la calidad.	No cumple	No se gestiona la calidad
		Acción correctiva de la gestión de la calidad.	No cumple	

Tabla 4. Cumplimiento de procesos ISO 12207 proceso 6.3.

Organización de Procesos		Cumplimiento	Comentario	
Procesos de Contexto del Sistema	6.3.1 Proceso de Planificación del Proyecto	Iniciación del proyecto.	Si cumple	En la planificación del proyecto no se incluye la gestión de riesgos y medidas de aseguramiento de la calidad
		Planificación del proyecto.	Si cumple	
		Activación del proyecto.	Si cumple	
		Seguimiento del proyecto.	Si cumple	
	6.3.2 Proceso de Evaluación y Control del Proyecto	Control del proyecto.	Si cumple	Se presentan informes verbales internos.
		Evaluación del proyecto.	Si cumple	
		Cierre del proyecto.	Si cumple	
	6.3.3 Proceso de Gestión de Decisiones	Planificación de la decisión.	No cumple	
		Análisis de la decisión.	No cumple	
		Seguimiento de la decisión.	No cumple	
	6.3.4 Proceso de Gestión del Riesgo	Planificación de la gestión del riesgo.	No cumple	
		Gestión del perfil del riesgo.	No cumple	
		Análisis del riesgo.	No cumple	
		Tratamiento del riesgo.	No cumple	
Monitoreo del riesgo.		No cumple		
6.3.5 Proceso de Gestión de la Configuración	Evaluación del proceso de gestión de riesgos.	No cumple	Se realizan cambios pero no se gestionan, ni documentan.	
	Planificación de la gestión de la configuración.	No cumple		
6.3.6 Proceso de Gestión de la Información	Ejecución de la gestión de la configuración.	No cumple		
	Planificación de la gestión de la información.	No cumple		
6.3.7 Proceso de Medición	Ejecución de la gestión de información.	No cumple		
	Planificación de la medición.	No cumple		
	Desempeño de la medición.	No cumple		
	Evaluación de la medición.	No cumple		

Tabla 5. Cumplimiento de procesos ISO 12207 proceso 6.4.

Organización de Procesos		Cumplimiento	Comentario	
Procesos de Contexto del Sistema		Identificación de las partes interesadas.	Si cumple	
	6.4.1 Proceso de Definición de Requisitos de las Partes interesadas	Identificación de los requisitos.	Si cumple	
		Evaluación de los requisitos.	Si cumple	
		Acuerdo sobre los requisitos.	Si cumple	
		Registro de los requisitos	Si cumple	
	6.4.2 Proceso de Análisis de Requisitos del Sistema	Especificación de los requisitos.	Si cumple	Se realiza un evaluación de los requisitos de manera empírica, no se documenta
		Evaluación de los requisitos.	Si cumple	
	6.4.3 Proceso de Diseño Arquitectural del Sistema	Establecimiento de la arquitectura.	No cumple	
		Evaluación de la arquitectura.	No cumple	
	6.4.4 Proceso de Implementación	El propósito del Proceso de Implementación es producir un elemento especificado del sistema.	Si cumple	
	6.4.5 Proceso de Integración del Sistema	Integración.	Si cumple	
		Reparación de las pruebas.	No cumple	
	6.4.6 Proceso de Pruebas de Calificación del Sistema	Prueba de calificación.	No cumple	
	6.4.7 Proceso de Instalación del Software	Instalación del software.	Si cumple	
	6.4.8 Proceso de Soporte de la Aceptación del Software	Soporte de aceptación del software.	Si cumple	
	6.4.9 Proceso de Operación del Software	Preparación para la operación.	No cumple	Se realiza una serie de pasos para la operación del sistema pero no se documenta, no se definen procedimientos
		Activación y verificación de la operación	Si cumple	
Uso operativo.		Si cumple		
Soporte al cliente.		Si cumple		
6.4.10 Proceso de Mantenimiento del Software	Solución de los problemas de la operación.	Si cumple		
	Implementación del proceso.	No cumple	No se cuenta con planes, procedimientos para realizar el progracode mantenimiento de software, se trabaja de forma empírica. Se realiza la implementación no se documenta el resultado de las pruebas y el mismo desarrollador realiza las pruebas	
	Análisis de problemas y modificaciones	Si cumple		
	Implementación de la modificación.	Si cumple		
Revisión/aceptación del mantenimiento.	Si cumple			
6.4.11 Proceso de Retiro del Software	Migración.	Si cumple		
	Planificación del retiro del software.	No cumple		
		Ejecución del retiro del software.	No cumple	

Tabla 6. Cumplimiento de procesos ISO 12207 proceso 7.1.

Organización de Procesos		Cumplimiento	Comentario	
Procesos Específicos del Software	7.1.1 Proceso de Implementación del Software	Estrategia de Implementación del Software.	Si cumple	Pero no cumple con todos los procesos del nivel superior como el diseño arquitectural del software y proceso de pruebas de calificación del software
	7.1.2 Proceso de Análisis de Requisitos del Software	Análisis de requisitos del software.	Si cumple	
	7.1.3 Proceso de Diseño Arquitectural del Software	Diseño arquitectural del software	Si cumple	
	7.1.4 Proceso de Diseño Detallado del Software	Diseño detallado del software.	No cumple	
	7.1.5 Proceso de Construcción del Software	Construcción del software.	Si cumple	No se cuentan con estándares de desarrollo
	7.1.6 Proceso de Integración del Software	Integración del Software	Si cumple	
	7.1.7 Proceso de Pruebas de Calificación del Software	Prueba de calificación del software.	No cumple	Se realizan pruebas, pero cada desarrollador prueba lo que desarrolló, no se realiza pruebas de integración ni regresión

**Tabla 7. Cumplimiento de procesos ISO 12207 proceso 7.2.**

Organización de Procesos		Cumplimiento	Comentario		
Procesos Específicos del Software		Implementación del proceso.	Si cumple		
	7.2.1 Proceso de Gestión de la Documentación del Software	Diseño y desarrollo.	Si cumple	Cada equipo maneja diferentes formatos de documentos, es revisado por cada jefe. Cuando se realizan cambios en el software no se actualizan los documentos.	
		Producción.	Si cumple		
		Mantenimiento.	No cumple		
	7.2.2 Proceso de Gestión de la Configuración del Software		Implementación del proceso.	No cumple	
			Identificación de la configuración.	No cumple	
			Control de la configuración.	No cumple	
			Determinación del estado de la configuración.	No cumple	
			Evaluación de la configuración.	No cumple	
	7.2.3 Proceso de Aseguramiento de la Calidad del Software		Gestión de liberación y entrega.	No cumple	
			Implementación del proceso.	No cumple	
			Aseguramiento del producto.	No cumple	
	7.2.4 Proceso de Verificación del Software		Aseguramiento del sistema de la calidad.	No cumple	
			Implementación del proceso.	No cumple	Verificación de los requisitos. Verificación del diseño. Verificación del código. Verificación de la integración. Verificación de la documentación.
			Verificación.	No cumple	
	7.2.5 Proceso de Validación del Software		Implementación del proceso.	No cumple	No se realizan casos de prueba
			Validación.	No cumple	
	7.2.6 Proceso de Revisión del Software		Implementación del proceso.	Si cumple	
		Revisiones de la gestión del proyecto	Si cumple		
		Revisiones técnicas.	No cumple		
7.2.7 Proceso de Auditoría del Software		Implementación del proceso.	No cumple		
		Auditoría del software.	No cumple		
7.2.8 Proceso de Resolución de Problemas del Software		Implementación del proceso.	Si cumple	Si bien se registran en un archivo Excel no existe una gestión y control adecuado, ha ocurrido que no se le ha dado prioridad a problemas que debían ser resueltos con prontitud.	
		Solución de problema.	Si cumple		

**Tabla 8. Cumplimiento de procesos ISO 12207 proceso 7.3.**

Organización de Procesos		Cumplimiento	Comentario	
Procesos Específicos del Software	7.3.1 Proceso de Ingeniería de Dominio	Implementación del proceso.	No cumple	
		Análisis de dominio.	No cumple	
		Diseño de dominio.	No cumple	
		Suministro de activos.	No cumple	
	7.3.2 Proceso de Gestión de Activos de Reutilización	Mantenimiento de activos.	No cumple	
			Implementación del proceso.	No cumple
			Definición del almacenamiento y la recuperación de activos.	No cumple
	7.3.3 Proceso de Gestión de Programas de Reutilización		Gestión y control de los activos.	No cumple
			Iniciación.	No cumple
			Identificación de dominio.	No cumple
			Evaluación de la reutilización.	No cumple
			Planificación.	No cumple
	Ejecución y control.	No cumple		
	Revisión y evaluación.	No cumple		

### **Proceso del área de soporte de FE.**

Este equipo se encarga de brindar soporte de segunda línea a las aplicaciones eGestor y ePortal que se encuentran en producción.

Actualmente trabajan de la siguiente manera (véase anexo 18):

El líder del equipo recibe las incidencias de los clientes por correo electrónico diariamente, las cuales son delegadas a los analistas programadores que se encuentran disponibles, en el caso que no se encuentren disponibles y de ser urgente se prioriza la revisión de la incidencia.

Se corrige la incidencia y se actualiza la versión en el cliente.

Se puede observar que entre las áreas de Desarrollo/Mantenimiento y Soporte FE no existe comunicación por más que ambos realizan cambios a las mismas aplicaciones, el mismo desarrollador realiza las pruebas y no se coordina el despliegue de versiones con el área de Desarrollo/mantenimiento.

Al iniciar la tesis se realiza un análisis FODA (Figura 4) para analizar a detalle lo que está generando este problema, en donde encontraremos “las fortalezas, debilidades, oportunidades y amenazas” que en la actualidad tiene la empresa por seguir laborando con una baja productividad.

	<b>Fortalezas (F)</b>	<b>Debilidades (D)</b>
<b>MATRIZ FODA</b>	F1 Empresa consolidada en el mercado F2 Capacitación del personal F3 Empresa está en crecimiento	D1 Procesos internos parcialmente no desarrollados, demanda de tiempo en atención de reporte de fallas del software. D2 Falta de comunicación entre el equipo de trabajo. D3 No existe una estandarización de desarrollo de software.
<b>Oportunidades (O)</b>	<b>Estrategias (FO)</b>	<b>Estrategias (DO)</b>
O1 Captación de nuevos clientes.	Incrementar la venta de servicios con los recursos disponibles. (F1, O1)	Completar el desarrollo de los procesos estándares, que contribuyan a la excelencia operativa como una de las ventajas competitivas. (D1, O1)
O2 Certificaciones	Consolidar actividades de capacitación presencial y virtual a los colaboradores, para lograr certificaciones ISO 12207.(F2, O2)	Disponer de capital de trabajo idóneo que permita hacer frente al desarrollo del área de software y entrenamiento. (D2, O2)
O3 Ampliación de nuevas líneas de servicio	Fomentar la creación de nuevas soluciones. (F3, O3)	Incrementar actividad de asesoría y captar nuevos mercados. (D3, O3)
<b>Amenazas (A)</b>	<b>Estrategias (FA)</b>	<b>Estrategias (DA)</b>
A1 Empresas que ofrecen servicios similares a menor costo. (Ex colaboradores que forman su empresa del mismo rubro en base al conocimiento obtenido.)	Publicidad intensiva para dar a conocer a la empresa haciendo énfasis en la diferenciación del servicio, en aras de contrarrestar la presencia de servicios sustitutos. (F1, A1)	Dar agilidad a la consolidación de procedimientos internos y administrativos, para poder ofrecer mayores ventajas ante las empresas sustitutas. (D1, A1)
A2 Bajo conocimiento del cliente para el desarrollo de un requerimiento/aplicativo.	Instruir/Coordinar al cliente para lograr que este perciba las diferencias del servicio. (F2, A2)	Dar énfasis en la comunicación interna y externa, para abordar oportunidad, permanecer en el mercado competitivo y propiciar la expansión
A3 Posicionamiento y curva de aprendizaje de competidores actuales	Asumir estrategia de diferenciación, calidad y precio. (F3, A3)	

Figura 4. Matriz FODA.  
Fuente: Elaboración propia.

En la matriz FODA se identifican las estrategias para implementar, mejoras en los procesos internos que contribuyan a la mejora operativa y de igual manera los problemas que se están presentando en la empresa.

Sobre la importancia de la estrategia D1O1 en el negocio, podemos indicar; que nos permite enriquecer el “proceso de desarrollo de software” mediante el uso de la estandarización que tienen como objetivo la consolidación de la mejora en el desarrollo.

A continuación, se visualizar el Diagrama Ishikawa, en donde muestra las principales causas de la baja satisfacción de los clientes por errores en la solución Factura Electrónica.

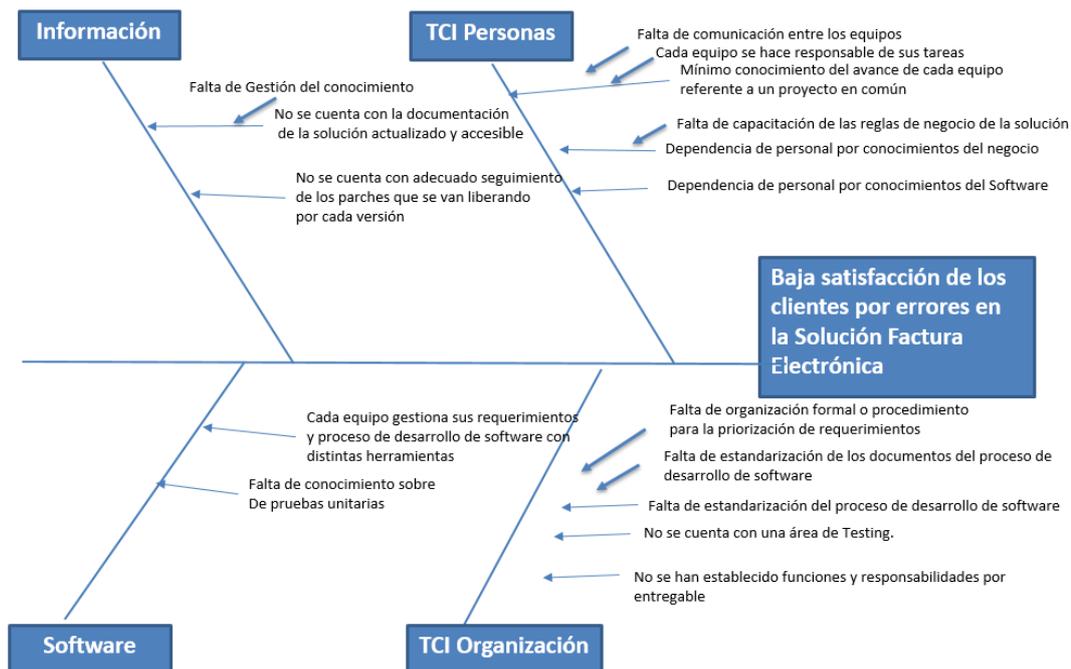


Figura 5. Diagrama de Ichikawa.  
Fuente: Elaboración propia.

Se decidió trabajar en la implementación del ISO/IEC 12207:2008 en fases, lo cual permitirá ver resultados en poco tiempo. En la primera fase que corresponde a la implementación de esta tesis, se priorizó la implementación de dos procesos:

Proceso de Calificación de Pruebas de Software.

Proceso de Resolución de Problemas de Software.

El Proceso de Calificación de Pruebas de Software, que contiene las actividades a realizar para mejorar la calidad de las aplicaciones.

Proceso de Resolución de Problemas de Software, permite asegurar que los problemas sean descubiertos, se identifiquen, analicen, administren y controlen hasta su posterior solución.

Los procedimientos, diagramas de procesos y plantillas de ambos procesos se encuentran en el capítulo de anexos.

La implementación de las siguientes fases se deberá realizar según la prioridad de los procesos a mejorar.

## **Justificación de la Investigación**

La implementación de un eficiente proceso de “Desarrollo de Software” muchas veces es dejado de lado, sobre todo en empresas que pasan de ser microempresas a medianas, porque consideran que pueden seguir trabajando de la misma manera empírica; sin contar con un área de control de calidad; esto ocurre hasta que ya no pueden brindar la misma atención a sus clientes.

El “Proceso de Desarrollo de Software”, cumple un rol muy importante en una organización, la cual se dedica al “desarrollo de software” propiamente dicho o que brinda servicios que incluyen el desarrollo y mantenimiento de aplicaciones como es el caso de la empresa TCI. Por lo tanto es importante aplicar el estándar ISO/IEC 12207:2008 para ordenar, concientizar y culturizar, adoptando mecanismos que les permita estandarizar y mejorar el proceso, dado que actualmente no se aplica ningún estándar ni metodología, y esto conlleva al desorden, la desorganización y hace que no se lleve un adecuado control de los cambios que se realizan en el software, lo cual muchas veces genera que el equipo deba volver a realizar el procedimiento. Esto genera costos e inclusive se desencadena falta de cumplimiento de contrato por las fallas que ocasiona el software y por entregas fuera de plazo afrontando con ello quejas que perjudican a la imagen y reputación de la Empresa.

## **MARCO REFERENCIAL**

### **Antecedentes**

#### **Antecedentes internacionales.**

Sánchez (2016), realiza el “Metamodelo para la definición e implantación de los procesos de planificación y control de proyectos en pequeños equipos de desarrollo, que plantea el desarrollo de un metamodelo orientado a los equipos pequeños que son formados dentro de las pequeñas empresas, de tal forma que les permita establecer de forma correcta los procesos de planificación y control de proyectos con el soporte de una herramienta computarizada que ayudará a implantar correctamente los componentes creados para dicho metamodelo. Se identificaron los problemas actuales en el desarrollo de software, concretamente aquellos relacionados con la gestión de los proyectos”.

Marroquín (2013), en su investigación “Gestión de la Seguridad Informática en Procesos de Desarrollo de Software a Medida.”, establece una guía de implementación de controles de seguridad informática que facilite la aplicación de estándares y buenas prácticas de desarrollo seguro, en proyectos de construcción de aplicaciones web a medida; indica que se debe de definir y aplicar un proceso de validación de la guía, que permita interactuar con referentes de diferentes organizaciones desarrolladoras de software, y cuyos resultados se han registrado y analizado para identificar puntos de mejora.

Mesquida (2012), en su estudio “Un Modelo para Facilitar la Integración de Estándares de Gestión de TI en Entornos Maduros. Elaborar un Modelo Integrado de Estándares de Gestión de TI, compuesto por: un sistema de gestión integrado según las normas ISO 9001, ISO/IEC 20000-1 e ISO/IEC 27001, y un mapa de procesos que toma como base los procesos del ciclo de vida del software de la norma ISO/IEC 15504-5 y los amplía a partir de las relaciones detectadas con los procesos de la norma ISO/IEC 20000-4 y los controles de la norma ISO/IEC 27002”.

Gallegos & Ortiz (2011), elaboran “Elaboración del Estándar de Aplicación de la norma ISO/IEC 12207, al desarrollo de aplicaciones de Software para la UTIC De La ESPE, quienes plantean elaborar un estándar de aplicación basado en la norma al desarrollo de aplicaciones de software; mencionan que no se pueden implementar de una sola vez todos los procesos que contemplan la norma ISO/IEC 12207, se debe de generar un plan de aplicación q contemple procesos de la norma, destinados a una área específica de la unidad, los cuales sirvan de base para posteriormente implementar el resto de procesos”.

Cardona (2009), desarrolla “Propuesta Metodológica para la realización de pruebas de Software en un Ambientes Productivos, quien plantea que la Industria de software en los últimos años, ha tomado la ruta comercial de vender productos con altos estándares de fiabilidad y eficiencia, por lo que las pruebas de carga y rendimiento se han convertido en obligatorias en todos los planes de prueba de los proyectos de software, y ésta necesidad es claramente evidenciada en la cantidad y calidad de productos de éste tipo que actualmente existen en el mercado”.

### **Antecedentes nacionales.**

Pacheco (2016), en su investigación de la “Propuesta Metodológica para la documentación de desarrollos de software basado en el ISO 12207:2008 en instituciones financieras de la ciudad de Arequipa, que propone una metodología para la documentación del desarrollo de software en instituciones financieras, con la propuesta desarrollada se logró la generación de información útil para los usuarios desde las etapas del análisis y definición del requerimiento hasta la documentación final logrando beneficios para la institución financiera; se logró generar información que contribuyó en el desarrollo del software como también durante su ejecución y mantenimiento. Con la metodología propuesta se logró otorgar eficiencia en el proceso de documentación durante el ciclo del desarrollo del software”.

Según Díaz (2015), desarrolla un estudio a una pequeña organización quien evaluó varios modelos de referencia aplicando finalmente ProCal-ProSer e ISO / IEC 29910, determinando la situación inicial de la organización, se ejecutó el ciclo de mejora obteniendo resultados positivos evidenciando una mejora en los procesos involucrados.

Según Alegre (2015), realiza la “Mejora de proceso software en una pequeña organización desarrolladora de software: caso PROCAL-PROSER-LIM. Nu-1er ciclo, sostiene que el proyecto cubre desde el análisis de la situación actual y finaliza con el reporte técnico, esto incluye la evaluación del ciclo de mejora realizado y las directrices para iniciar un nuevo ciclo de mejora. Se logró ejecutar el ciclo de mejora acorde a lo planificado mediante pilotos en proyectos en ejecución permitiendo mediante capacitaciones, manuales e inducciones corregir los problemas de adaptación que tuvieron los operarios de la organización”.

Merino (2014), elabora la “Implementación de la ISO/IEC 12207:2008 para mejorar los procesos asociados al ciclo de vida de software en una micro empresa peruana cuyo objeto social es el desarrollo de sistemas de información, el presente proyecto de tesis, pretende evaluar los procesos asociados al ciclo de vida de desarrollo de software en una mype y plantear propuestas de mejora de procesos tomando como marco de referencia la ISO/IEC 12207:2008 para la evaluación y mejora de procesos en microempresas y pequeñas empresas desarrolladoras de software. La ejecución de las mejoras en el proyecto contó

con el apoyo de la gerencia, que fue llevada de manera gradual con el objetivo de ir creando conciencia en las personas que cambiando de a pocos en lo que se realiza, se llegará a que la empresa sea mejor y logre un bienestar a nivel general”.

Llontop (2011), en su estudio la “Mejora En el Proceso de Desarrollo de Nuevas Línea de Productos basado en la Metodología Qfd Asegurando la Calidad Del Producto a lo largo de su ciclo de vida. El QFD, es un método para desplegar, antes del arranque de producción en masa, los puntos importantes de aseguramiento de calidad necesarios para asegurar la calidad de diseño a través del proceso de producción. Se concluyó que el uso de la herramienta QFD permite integrar las necesidades del cliente como principal inputs para el desarrollo de las nuevas líneas de productos convirtiéndose en el principal componente en la transformación de las necesidades de los clientes hacia nuevos productos”.

### **Estado del arte**

Según la investigación realizada en tesis nacionales e internacionales se ha podido determinar que en la actualidad muchas empresas que se encuentran en crecimiento y que brindan servicios de desarrollo de software muestran su preocupación en la mejora de sus procesos optando por diversas opciones entre ellas ISO/IEC 12207, RUP, Scrum, Kanban.

### **Marco Teórico**

#### **La industria del software.**

Según Merino (2014) en su publicación “Implementación de la ISO/IEC 12207:2008 para mejorar los procesos asociados al ciclo de vida de software en una micro empresa peruana cuyo objeto social es el desarrollo de sistemas de información” afirmó:

La industria del software, podemos decir que representa una actividad económica de suma importancia para toda la urbe; ofreciéndonos múltiples fuentes de negocio y dándonos oportunidades a los países en vía de desarrollo. Por ende, la industria de “software” en los países latinoamericanos todavía es considerada incipiente e inmadura.

Asimismo, consideramos que las empresas de software cuentan con procesos optimizados y ágiles que podrían cubrir con una gran demanda en el rubro empresarial; pero debería de orientarse hacia la obtención de la calidad a niveles de productos, servicios y procesos; esto conllevará en un futuro a generar la confianza del cliente, lo cual se verá reflejado en un futuro y así obtener un producto deseable, eficiente y sobre todo cumplir con el tiempo acordado/estimado dando como resultado final un cliente satisfecho.

El proyecto está en búsqueda de brindar propuestas sobre los procesos que actualmente están generando más problemas, los cuales son los procesos de “Desarrollo y Mantenimiento”; vinculados al “Ciclo de Vida de Desarrollo de Software” de una empresa mediana, elaborando propuestas de mejoras bajo la norma “ISO/IEC 12207:2008”.

#### **ISO/IEC 12207:2008.**

Implementa una referencia en común para los “procesos de ciclo de vida de software”, estableciéndose las terminologías definidas que regula la industria del software.

Estos procesos, están conformados por; actividades y tareas que se deberán de aplicar durante la adquisición, suministros, desarrollo, operación, mantenimiento y eliminación de productos/servicios de software.

La norma define, controla y mejora los procesos del “Ciclo de vida de Software”; los procedimientos adecuados para conseguir el fin se puede realizar sola o en conjunto con otras normas, dependiendo básicamente de las necesidades que existan en la empresa.

#### ***Propósito.***

El objetivo, es identificar cuáles serán los procesos que nos facilitará en la comunicación entre proveedores, compradores y demás dirigido en “el ciclo de vida del software”.

Está propuesta, puede ser aplicada en una situación que participan dos partes, donde ambas partes pertenecen a una misma organización; dicha situación puede variar desde un simple acuerdo informal, hasta un contrato

que este jurídicamente establecido; también puede ser establecido en una única parte a través del auto imposición establecida de los procesos.

La norma, está enfocada a los adquirientes de sistemas/productos de software y servicios, proveedores, desarrolladores, operadores, gerentes, directores de control de calidad y usuarios.

### ***Limitaciones.***

Respecto a la definición métodos o procedimientos que sean necesarios para cumplir con los resultados y requisitos de un proceso, la norma no brinda detalle del “ciclo de vida de los procesos”.

Cada usuario puede decidir que documentación desarrollará ya que no contiene detalle de la documentación en términos de contenido, nombre, formato y medios de grabación.

### ***Conformidad.***

La aplicación de la norma en general, consiste en seleccionar un conjunto de procesos y adecuarlos para determinada organización/proyecto, en vista de que no, será necesaria la inclusión de todos los procesos establecidos en la norma.

Existen dos formas; una implementación se ajusta a esta norma y cualquier declaración de conformidad puede ser citada en una sola de las dos formas que se muestran a continuación.

### ***Conformidad Completa.***

Cuando se demuestra los resultados como evidencia que todos los procesos establecidos por la norma se han logrado satisfactoriamente.

### ***Conformidad a la Medida.***

Cuando se demuestra los resultados como evidencia, que se han utilizado como base un conjunto de procesos específicos y estos se han logrado satisfactoriamente.

***Recomendación de la “Norma ISO/IEC 12207:2008 de acuerdo a los procesos del ciclo de vida del software”.***

La norma recomienda una definición en común, la cual nace de una idea/necesidad, para los “procesos de ciclo de vida del software”; las cuales puede ser satisfactorio, en parte o en su totalidad y que culmina con la terminación de la misma.

La creación de un software, puede estar conformado por diversos modelos de “ciclo de vida” como son: Modelo Lineal, Modelo Cascada, Modelo Evolutivo, Modelo Incremental, Modelo Espiral, Modelo Ágil.

“La norma ISO/IEC 12207:2008”, esta agrupada por las actividades a realizarse durante el proceso de “ciclo de vida de software” en siete grupos: cada uno de estos grupos se describen en función del propósito y resultados que se obtienen y se enumeran las actividades/tareas que se deberán de seguir para cumplir con dichos resultados.

La norma, no requiere la implementación de un “modelo de ciclo de vida de software” específico, pero nos recomienda que para cada proyecto se defina el “modelo de ciclo de vida” adecuado, el cual haya sido establecido por la organización para manejar diversos proyectos.

La figura 6 muestra los procesos agrupados por área de aplicación.

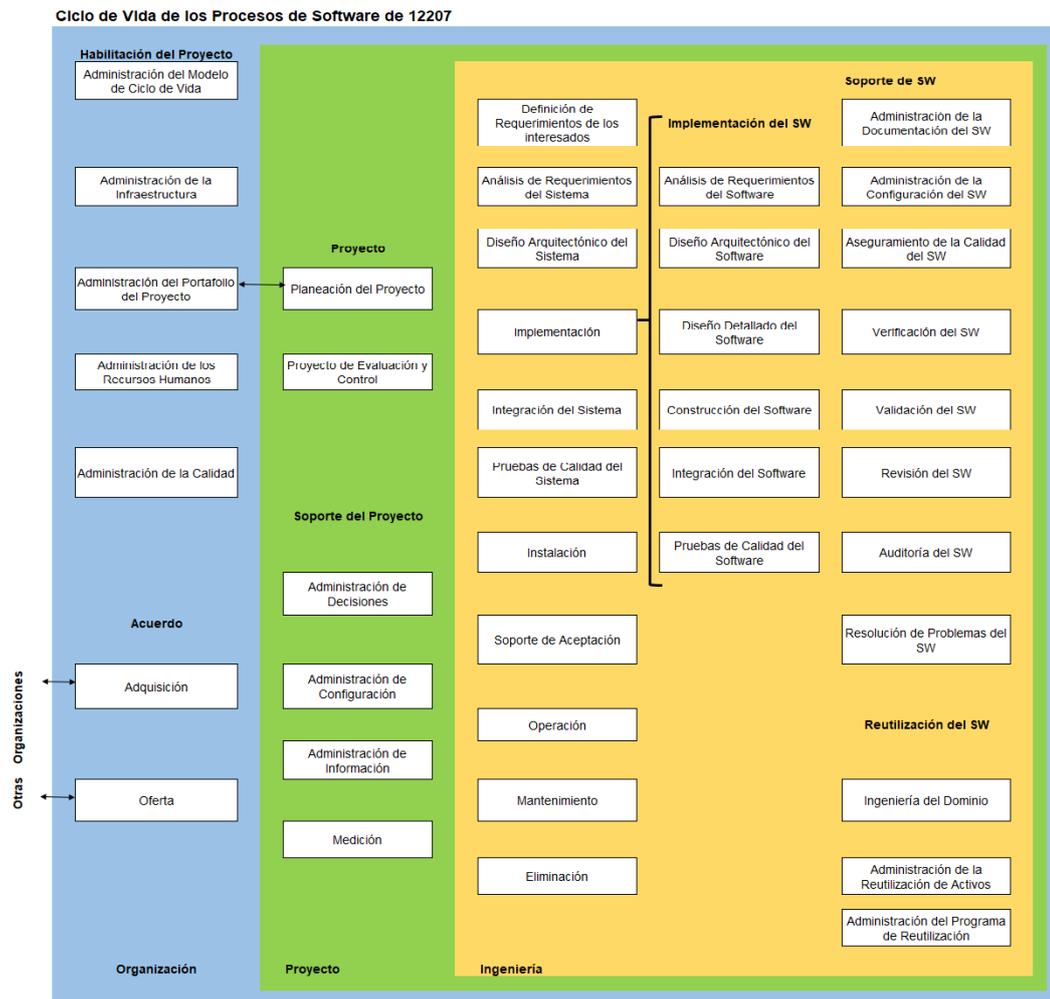


Figura 6. Ciclo de vida de los procesos de software de ISO/IEC 12207:2008.  
 Fuente: Repositorio Institucional de la Universidad de las Fuerzas Armadas ESPE.  
 Recuperado de <https://goo.gl/m5W33N>

Los procesos de la “norma ISO/IEC 12207:2008”, están distribuidos de acuerdo a la figura 7, en donde se detalla la estructura de los procesos.

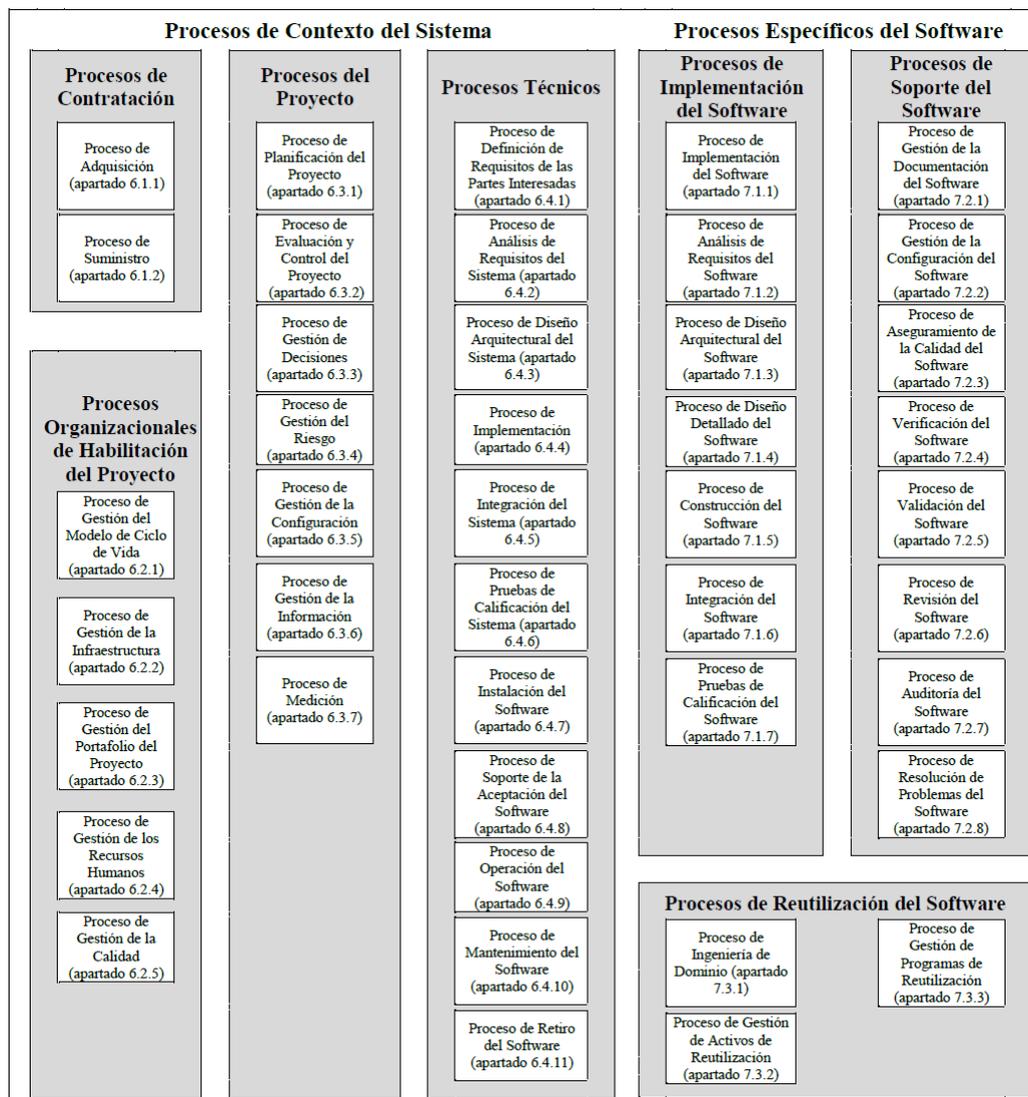


Figura 7. Estructura de los procesos de ISO/IEC 12207:2008.

Fuente: Repositorio Institucional de la Universidad de las Fuerzas Armadas ESPE. Recuperado de <https://goo.gl/m5W33N>

La norma ISO/IEC 12207, se componen de 43 procesos que pueden aplicarse durante la adquisición, suministro, desarrollo, operación, mantenimiento y evolución de productos software.

**ISO 15504.**

La norma ISO 15504 SPICE (“Determinación de la capacidad de mejora del proceso de software”), es una norma que se utiliza para evaluar/mejorar la capacidad y madurez de los procesos; conjuntamente con “la norma ISO 12207” ambas aplican la evaluación y mejora de la calidad del proceso de desarrollo y mantenimiento de software, la cual es una norma abierta e internacional.

***Características.***

Esta comprende: la evaluación de procesos, mejora de procesos y la determinación de capacidad.

¿Que proporciona?, requisitos para los modelos de evaluación de los procesos y modelos de evaluación de organizaciones y sirve como guía para la definición de las competencias de un evaluador de procesos.

***Modelo de “procesos ISO/IEC 15504 – ISO/IEC 12207:2008”***

En los procesos de la mejora de la calidad, participan dos tipos de modelos que son los siguientes:

***Modelo de Procesos***

Se define una colección/catálogo estructurado de aplicar las buenas prácticas que se describe las características de un proceso efectivo y adecuado.

***Modelo de evaluación***

Se requieren los principios que brindan el modelo de procesos en una organización, realizando una evaluación de la calidad e implantación.

Como se observa en la Figura 8, se hace uso del modelo de procesos “ISO/IEC 12207:2008” e “ISO/IEC 15504” como modelo de evaluación.

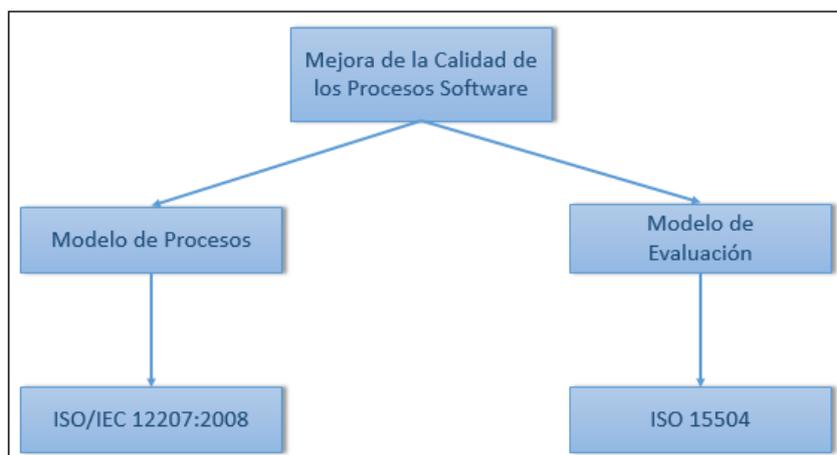


Figura 8. Mejora de la calidad de los procesos software.  
Fuente: Spice/Iso/Iec 15504. Recuperado de <https://goo.gl/426FyU>

“La norma ISO/IEC 15504”, en la parte siete; da a conocer para llevar a cabo evaluaciones por niveles de madurez, que se describe un conjunto de niveles.

La norma establece seis niveles de madurez para mejorar y evaluar a las organizaciones, desde el nivel cero (inferior) hasta el nivel cinco (superior).

A continuación la figura 9 muestra los niveles de madurez.



Figura 9. Niveles de madurez.  
Fuente: Spice/Iso/Iec 15504. Recuperado de <https://goo.gl/426FyU>

Según la norma, la parte siete no describe los procesos a superar en cada uno de ellos pero si los niveles.

La Figura 10, nos muestra un aporte de los procesos definidos para los niveles uno y dos.



*Figura 10.* Mejora de procesos de software iso/iec 15504 – iso/iec 12207:2008.  
Fuente: Spice/Iso/Iec 15504. Recuperado de <https://goo.gl/426FyU>

Los niveles uno y dos los iguales, la condición es que para ir del nivel uno a dos debe mejorar la organización mediante la capacidad o gestión de procesos.

Nos proporciona un modelo de evaluación de procesos, para los procesos del “ciclo de vida del software” definido en “la norma ISO/IEC 12207”, que define los procesos del ciclo de vida del desarrollo, mantenimiento y operación de los sistemas de software.

***Ventajas: norma spice/iso/iec 15504.***

Dimensiones independientes para los procesos y la capacidad, siendo el primer modelo de procesos de dos dimensiones.

Es el modelo que tiene un mayor reconocimiento en el continente Europeo ya que ha sido el más probado y consensuado.

Tiene relación con otros modelos de calidad ya implementados en la organización “ISO9001- ISO 20000- ISO 27000”.

Tiene un menor costo en la certificación. La evaluación es más amplia entre los modelos de mejora, ya que se evalúa por niveles de madurez.

***Desventaja: “Norma SPICE/ISO/IEC 15504”.***

Permite el dominio de procesos sea tan amplio para abarcar para todos los posibles ciclos de vida de la “ISO/IEC TR-15504”, lo cual hace complicado que todos los atributos del proceso sean universales, ocasionando confusión durante la evaluación.

**CMMI.**

Desarrollado por el “SEI” (Software Engineering Institute) para identificar el nivel de madurez, los cuales han alcanzado en función del nivel de capacidad de sus procesos involucrados en el “desarrollo de software” en las organizaciones.

CMMI, se estructura de la siguiente manera: 22 áreas de procesos agrupados en cuatro categorías y tienen dos representaciones:

***Continua***

Cada proceso a mejorar obedece a las necesidades de la organización, los procesos se eligen para ser mejorados y otorga un nivel de capacidad individual por cada área de proceso.

***Por etapas***

Se elige un grupo de áreas de proceso, mediante se concentrará sus esfuerzos y una vez que se cubran todas las áreas de proceso a un nivel “n” se podrá afirmar que la organización ha alcanzado el nivel de madurez “n” en la organización.

El modelo CMMI, define cinco niveles de madurez para las organizaciones; los cuales se alcanzarán por etapas mientras que para los procesos, define seis niveles de capacidad, los cuales se alcanzarán en la representación continua.

Según el cumplimiento definido por CMMI, se lleva a través de prácticas agrupadas en 22 áreas de proceso; éstas a su vez están agrupadas en cuatro categorías: Gestión de Proyectos, Ingeniería, Soporte, Gestión de Procesos.

En la tabla 9 se muestran los niveles de madurez de la organización según CMMI y en la tabla 10 se muestran los niveles de capacidad de procesos.

Tabla 9. *Niveles de madurez de CMMI.*

<b>Niveles de Madurez</b>	
<b>Nivel de madurez de la organización</b>	<b>Centrado en</b>
5. Optimizado	Mejora continua del proceso
4. Gestionando cuantitativamente	Control cuantitativo del proceso
3. Definido	Procedo caracterizado por la organización y proactivo
2. Gestionado	Gestión básica del proyecto
1. Inicial	Proceso impredecible. Control reactivo

Nota: Obtenido de <https://goo.gl/t24uvy>

Según Villa, Ruiz & Ramos (2004), entre las bondades del CMMI podemos encontrar:

Permiten asegurar que los procesos asociados con cada área de proceso serán efectivos, repetibles y duraderos, dándose la inclusión de las prácticas de institucionalización.

La transición del aprendizaje individual al aprendizaje de la organización por mejora continua, nos dará las lecciones aprendidas, al uso de bibliotecas y las bases de datos de los proyectos mejorados.

Para ser completamente implementado al 100%, podría ser difícil de comprenderlo y/o entenderlo. Entre las limitaciones/restricciones del CMMI, se puede mencionar: Para cada tipo de organización puede ser demasiado detallado, la interpretación puede considerarse prescriptiva, asimismo podría requerirse mayor inversión de recursos (tiempo y personas).

Tabla 10. *Niveles de Capacidad de CMMI.*

<b>Capacidad del Proceso</b>	
<b>Nivel de Capacidad</b>	<b>Descripción</b>
Nivel 0 - Incompleto	El proceso está sin implementar, los objetivos no se han alcanzado.
Nivel 1 - Realizado	El proceso logró su propósito a nivel general, existen productos que acreditan el cumplimiento, sin embargo no existe planificación.
Nivel 2 - Gestionado	El proceso es gestionado (planificado y seguido), los productos resultantes se han logrado con requisitos de calidad, tiempo y recursos según lo planteado inicialmente.
Nivel 3 - Establecido	El Proceso es gestionado y realizado usando procesos estándares y basados en buenas prácticas de Ingeniería de Software.
Nivel 4 - Predecible	El proceso se caracteriza por la consistencia de su desempeño, dispone de límites de control para alcanzar las metas ya definidas. Se conoce la capacidad del proceso y habilidad de gestionar y predecir el rendimiento.
Nivel 5 - En Optimización	El proceso es realizado de la forma optimizada, buscando satisfacer las necesidades actuales y futuras del negocio. Los objetivos del proceso se definen en función a los objetivos de la organización. La optimización puede incluir ideas innovadoras y redefinición del proceso.

Nota: Obtenido de <https://goo.gl/t24uvy>

En la tabla 11 muestra la comparativa entre los modelos CMMI vs. ISO 15504.

Tabla 11. *Comparativa entre los modelos CMMI vs. ISO 15504.*

	<b>CMMI</b>	<b>ISO 15504</b>
<b>Ámbito de aplicación</b>	<b>Software y Sistemas</b>	<b>Software y Sistemas</b>
En su favor	El de mayor prestigio	Más consensuado y probado
En su contra	Difícil de entender, mayor inversión, prescriptivo	Difícil en capacidad, complejo para evaluar
Procesos	Estructura propia	Delega en ISO 12207, por mayor aplicabilidad
Validación	Encuestas satisfacción y casos de estudio	"Trials" y esfuerzo empírico
Objetivo	Mejora del proceso, determinación capacidad contratista	Valoración del proceso y guía para la mejora
Representación	Continua y por etapas	Continua (Por etapas a nivel de proceso)
Técnicas análisis	Cuestionario de evaluación	Varios
Método para mejora de procesos	IDEAL, mapa guiado	SPICE 4° Parte

Nota: Obtenido de <https://goo.gl/t24uvy>

### **Metodologías de Desarrollo.**

Según Álvarez & Taype (2015) en su publicación “Modelos De Aceptación De Metodologías De Desarrollo De Software” afirmaron:

“Una metodología de desarrollo de software, es definido como un conjunto de documentos de políticas, procesos y procedimientos que forman parte de un marco de trabajo usado por los equipos de desarrollo para estructurar, planificar y controlar el proceso de desarrollo de software, optimizándolo mediante el aumento de la productividad del personal de tecnologías de información y una mejora de la solución o producto de software.”

Los criterios de una metodología de desarrollo de software podrían considerarse un factor crítico en el desarrollo y la transformación gradual de las empresas desarrolladoras que no solo obtienen beneficios en la estructuración y organización de sus procesos de desarrollo, sino que logran optimizarlos para aumentar la producción y el nivel de calidad de los productos desarrollados.

Los dos tipos de metodologías que existen y son utilizadas por la mayoría de desarrolladores de software son el desarrollo tradicional y el desarrollo ágil.

En la tabla 12 se observa de manera general las características de cada metodología y cuáles son sus diferencias.

Tabla 12. Comparación de enfoque ágil vs el tradicional.

Tópico	Enfoque tradicional	Enfoque ágil
Ciclo de vida de desarrollo	Lineal, modelo ciclo de vida (cascada, espiral o alguna variación)	Iterativo, modelo evolutivo
Estilo de desarrollo	Anticipado	Adaptativo
Requerimientos	Conocidos tempranamente, estables, claramente definidos y documentos	Emergentes, de cambios rápidos, desconocidos (Se descubren durante el desarrollo del proyecto)
Arquitectura	Arquitectura pesada para requerimientos actuales y futuros.	Percepción YAGNI (—No lo vas a necesita)
Gestión	Centrado en procesos	Centrado en las personas
	Comandos y control	Liderazgo y colaboración
Documentación	Compleja y detallada	Ligera (Reemplazada por comunicación directa)
	Conocimiento explícito	Conocimiento tácito.
Objetivos	Predicciones y optimización	Exploración o adaptación
Cambio	Tienden a rechazar el cambio	Aceptan el cambio
Miembros de equipo	Equipo distribuido de especialistas, orientado al plan	Ágil, conocimiento enfocado, colaborativo
Organización del equipo	Equipo pre-estructurados	Equipos auto organizados
Participación del cliente	Participación baja y pasiva	Cliente activo y proactivo. Es considerado parte del equipo
Cultura organizacional	Cultura de control y mandato	Cultura de liderazgo y colaboración
Proceso de desarrollo de software	Enfoque y solución universal para proporcionar predicciones y calidad.	Enfoque flexible adaptado a la comprensión colectiva de las necesidades para proporcionar un desarrollo más rápido.
Indicadores de éxito	Conformidad con el plan	Valor de negocio entregado

Nota: Obtenido de <https://goo.gl/t24uvy>

### Metodologías de Desarrollo Tradicionales.

Según Brito (2009) en su publicación “Metodologías tradicionales y metodologías ágiles”:

Las metodologías tradicionales, imponen una disciplina con el fin de conseguir un software más eficiente. Para esto, se hace énfasis en la planificación total del trabajo a realizarse, una vez detallado, comienza el ciclo de desarrollo del producto software”.

Las metodologías tradicionales, se enfocan en el control del proceso mediante la definición de roles, actividades, artefactos, herramientas, notaciones para el modelado y documentación detallada.

Las metodologías tradicionales no logran adaptarse de manera adecuada a los cambios; por lo cual, no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse y pueden variar.

Entre las metodologías tradicionales se encuentran: Rup y Msf.

### **Rup.**

Es un proceso de ingeniería de software, para satisfacer las necesidades de los usuarios, mediante la asignación de tareas y responsabilidades dentro de una organización, lo cual tienen que cumplirse dentro de los límites presupuestales y de tiempo. Asimismo, tiene como objetivo asegurar la producción de software de mayor calidad.

Es una metodología de desarrollo iterativo, se enfoca al manejo de arquitectura, diagramas de casos de uso y riesgos.

Mejora la productividad del equipo, incluyendo conocimientos; permitiéndoles que cada miembro del grupo pueda acceder a la misma base de datos sin importar su responsabilidad específica.

Nos permite que todos los miembros compartan la misma visión y lenguaje de cómo desarrollar un software, dándonos una fortaleza en la organización.

En la figura 11 muestra las fases de flujo de trabajo del proceso y el esfuerzo en actividades según fase del proyecto.

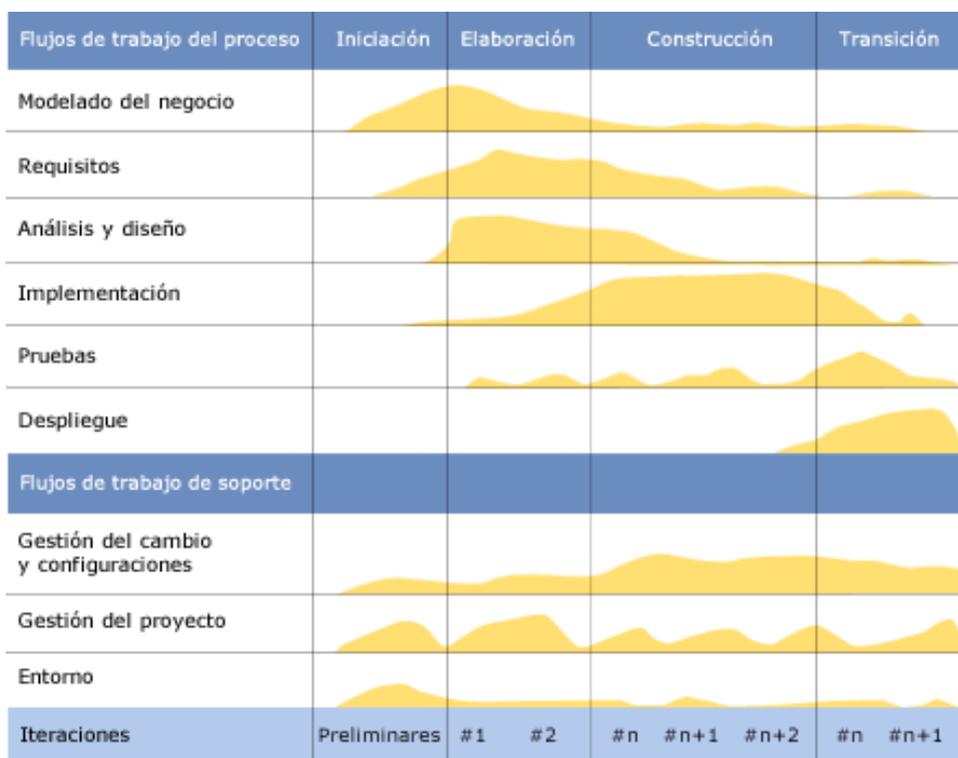


Figura 11. RUP-Esfuerzo en actividades según fase del proyecto.

Fuente: Wikipedia *Proceso unificado racional*. Recuperado de <https://goo.gl/T4QEL7>

### ***Principales características.***

Es iterativo e incremental, está conformada en los casos de uso; en otras de sus características verifica de manera seguida la calidad del software y administra los requisitos y el proceso de desarrollo que tiene tantos artefactos como roles.

#### *Estructura y elementos específicos de la metodología.*

*A continuación la descripción de las fases de desarrollo del software:*

##### *Fase de inicio.*

En esta etapa, se define el objetivo para determinar la “visión del proyecto”. Se realiza un plan de fases, en donde se identifican los principales casos de uso y los riesgos.

##### *Modelado del negocio.*

Primeramente, el equipo de trabajo, deberá de conocer el funcionamiento de la empresa y sus procesos.

Entender la dinámica y estructura de la organización para la cual el sistema va a ser desarrollado; para focalizar el problema actual en la organización e identificar mejoras idóneas.

En la organización, los clientes, usuarios finales y desarrolladores, se asegurarán en tener un entendimiento en común como objetivo.

##### *Requisitos.*

Los usuarios finales tienen que aceptar y comprender los requisitos que se especifique, los cuales estarán indicados en el contrato que deberá de cumplirse.

El sistema puede lograr a disponer a los desarrolladores de un mejor entendimiento de los requisitos del sistema, los cuales sean captados sin mayores inconvenientes, esto facilitará estimar costo/tiempo en el desarrollo del sistema, precisar una interfaz de usuarios para el sistema, lo cual se debe de enfocarse como base a las necesidades y metas del usuario.

### *Fase de elaboración.*

El plan de proyecto, se realiza para completar los casos de uso y se mitigan los riesgos. Asimismo, se planifica las actividades, especificando las características y el diseño de la arquitectura (para determinar la arquitectura óptima según los objetivos).

### *Análisis y diseño.*

Se describe cómo se va a implementar el sistema, en base a especificaciones de los requerimientos. Asimismo, se transforman los requisitos de acuerdo al diseño del sistema y se desarrolla una arquitectura específica para esta.

Se adopta el diseño, para que sea el más acorde y consistente con el entorno de implementación.

### **Fase de construcción**

Se construye el producto, la arquitectura y los planes, hasta que el producto está listo para ser enviado a los usuarios. El objetivo es obtener la capacidad operacional inicial.

### *Implementación.*

Se logra implementar: clases y objetos en ficheros fuente, binarios, ejecutables y demás, dándose como resultado un sistema ejecutable.

Se da a conocer que cada implementador toma la decisión del orden que se implementara los elementos del subsistema.

El sistema se integra siguiendo el plan; en caso de encontrar errores de diseño, este deberá ser notificado.

### *Prueba.*

Mediante el flujo de trabajo se evaluará la calidad del producto que se está desarrollando. Para poder lograr la integración en "todo el ciclo de vida"; se deberá de hallar, encontrar, documentar fallas e inconstancias en la calidad del software.

Se realizara las demostraciones sólidas, mediante las validaciones de las suposiciones indicadas en el diseño y las descripciones de los requisitos.

En base a lo diseñado, se verificará las funciones del producto de software, para que se tengan una apropiada implementación.

#### *Etapa de transición.*

Teniendo como objetivo el logro del entregable del proyecto, se procederá a la instalación del producto en el cliente, posteriormente se realiza el entrenamiento de los usuarios.

Se busca que el cliente quede debidamente satisfecho, de acuerdo a la transición del producto a los usuarios; lo cual incorpora: entrenamiento, soporte, envío, manufactura, mantenimiento del producto.

#### *Despliegue.*

Se tiene como objetivo realizar la distribución de los productos con éxito y entregar a los usuarios, las siguientes actividades implicadas contienen los pasos indicados en la figura 12:



Figura 12. Actividades del despliegue

Fuente: Wikipedia *Proceso unificado racional*. Recuperado de <https://goo.gl/T4QEL7>

*Ciclo de Vida de desarrollo de software utilizado.*

### **Proceso Iterativo e Incremental**

El Modelo utilizado para el desarrollo de un proyecto de software (RUP), plantea la implementación del proyecto a realizarse en iteraciones, definiendo los objetivos por cumplir en cada iteración y así poder ir completando todo el proyecto iteración por iteración. Dándose varias ventajas; entre ellas se puede mencionar: tener pequeños avances del proyecto que son entregables al cliente el cual puede probar mientras se está desarrollando otra iteración del proyecto, con lo cual el proyecto va creciendo hasta ser completado en su totalidad.

La figura 11 muestra el ciclo iterativo e incremental de RUP.



*Figura 13. Iterativo e incremental (Rup).*

Fuente: Foro Activo *Metodología Rational Unified Process Rup*. Recuperado de <https://goo.gl/5ZoQPY>

*Roles que forman parte.*

En la tabla 13, se describe los roles.

Tabla 13. *Roles de RUP.*

Rol	Descripción
Analistas	Analista de procesos de negocio, diseñador del negocio, analista de sistema, especificador de requisitos.
Desarrolladores	Arquitecto de software, diseñador, diseñador de interfaz de usuario, diseñador de cápsulas, diseñador de base de datos, implementador, integrador.
Gestores	Jefes de proyecto, control de cambios, configuración, pruebas, despliegue. Ingeniero de procesos, revisor de gestión del proyecto, gestor de pruebas.
Apoyo	Documentador técnico, administrador de sistema, especialista en herramientas, desarrollador de cursos, artista gráfico.
Especialista en pruebas	Especialista en pruebas, analista de pruebas, diseñador de pruebas.
Otros roles	Stakeholders, revisores, coordinación de revisiones, revisor técnico.

Nota: Obtenido de <https://goo.gl/t24uvy>

#### *Ventajas.*

“Requerir de conocimientos del proceso y asimismo de UML”.

“Lograr el progreso visible en las etapas tempranas”.

“Uso de iteraciones”.

“Evaluar los riesgos en lugar de descubrir en la integración final del sistema”.

“Facilita la reutilización del código”.

#### *Desventajas.*

Es muy probable que no resulte muy adecuado y de no aplicación en el estilo cascada; debido a la complejidad.

#### ***Msf.***

Define una metodología de trabajo para construir e implantar sistemas empresariales distribuidos basados en herramientas y tecnologías de Microsoft para cualquier plataforma (Linux, Citrix, Microsoft, Unix).

“Msf, nos da un conjunto de principios, modelos, disciplinas, conceptos y lineamientos para la entrega de tecnología de la información utilizando soluciones Microsoft”.

Msf, permite decidir qué método utilizar no obligando al desarrollador a utilizar una determinada metodología (Waterfall, Agile).

Msf, es aplicable a otros proyectos de TI, así como proyectos de implementación de redes o infraestructura.

#### *Componentes de msf.*

##### *Principios.*

Los principios en que se fundamenta “El modelo de equipo de trabajo” se propone tomando como base en:

Fortaleciendo al equipo, brindándoles capacitación.

Asignaciones de responsabilidades y autoridad.

Que la comunicación sea abierta.

Tiene valor agregado.

Calidad.

Aprender de las experiencias.

##### *Disciplinas.*

Gestión de proyectos/control de riesgos/control de cambios.

##### *Modelos*

En la figura 14, podemos visualizar el modelo de equipo de trabajo de Msf.

### Modelo de equipo de trabajo.



Figura 14. Modelo de equipo de trabajo en msf.

Fuente: Metodología de desarrollo de software – msf. Recuperado de <https://goo.gl/aEQcpQ>

### Modelo del proceso.

Se compone de cinco etapas iterativas o fases que se muestra en la figura 14; cuando se finaliza cada fase se cuentan con alcances específicos los cuales son definidos por hitos y se logra generar los entregables específicos, los cuales agregan el valor al proyecto:

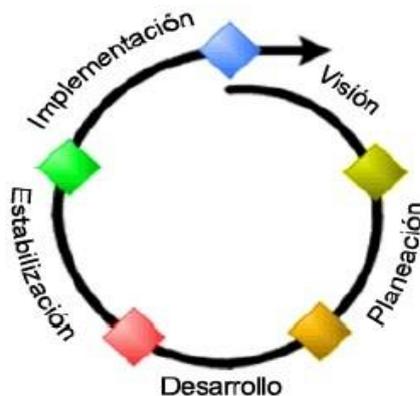


Figura 15. Etapas o fases del modelo de proceso de MSF.

Fuente: Metodología de desarrollo de software – MSF. Recuperado de <https://goo.gl/aEQcpQ>

### *Fases de la metodología.*

#### *Visión.*

De esta fase dependerá el éxito o su fracaso del proyecto. Para esto se debe de tener una visión a los objetivos del negocio; se debe de identificar los beneficios que traerá el proyecto, así como también sus restricciones y alcances.

Deberá corresponder ser el primer punto del propósito del proyecto, tener en cuenta los objetivos específicos que deben ser claros, medibles y alcanzables en un tiempo determinado.

Documentos a entregarse: “Documento visión, detalle de la visión, requerimientos funcionales, matriz de riesgos y acta de aprobación de visión”.

#### *Planeación.*

De acuerdo a un cronograma de trabajo que cumpla con lo planeado. Se realiza la planificación según el objetivo del proyecto y la arquitectura de la solución especificada en la fase de “Visión y Alcance”.

Se generará la lista de actividades que se deberán ejecutar como, los recursos asociados, responsabilidades y los costos.

De acuerdo a la planificación se prepara al proyecto para alcanzar el éxito, detectándose en forma temprana los riesgos y así tomar medidas para enfrentarlos y buscar siempre la solución óptima.

Documentos a entregarse: “Documento de cronograma, acta de aprobación de cronograma”.

#### *Desarrollo.*

“MSF, recomienda iniciar a construir código a partir de las funcionalidades básicas e ir haciendo entrega de cada funcionalidad desarrollada para someterse a pruebas unitarias y evaluaciones de experiencia de usuario; esto a su vez conlleva a realizar ajustes de cronograma necesarios”.

Documentos a entregarse: “Fuentes y ejecutables, documentos, manuales, acta de finalización de desarrollo, entre otros”.

### *Estabilización.*

Es la versión final o culminación del producto debidamente probada, ajustada y aprobada en su totalidad.

Documentos a entregarse: “Documento registro de pruebas, acta de aprobación de versión aprobada”.

### *Implantación.*

Entregar e instala el producto finalizado al cliente.

Documentos a entregarse: “Conjunto de archivos, ejecutables, directorios, base de datos, scripts, instaladores, manuales, licencias, entre otros; los cuales permitan su instalación y correcto funcionamiento, como son: acta de entrega y finalización de proyecto”.

### *Soporte.*

“Se brindara la garantía al producto durante un tiempo estipulado en el contrato, registrándose los reportes de soporte y mantenimiento recibido así como los ajustes que estén dentro de lo escrito en los documentos de la fase de “Visión”.

Documentos a entregarse: Documento de registro de reportes de mantenimiento, soporte y ajustes hechos.

### *Contextos en la aplicación.*

El MSF, se puede aplicar una gran diversidad de contextos en los “proyectos de TI”, utilizándose como desarrollo de software e “implementación de redes e infraestructura”.

### *Análisis comparativo.*

Lo que hay que tomar en cuenta, son los puntos fuertes en el MSF, como son: la gestión de proyectos, gestión de riesgos y la facilidad del desarrollo de aplicaciones web.

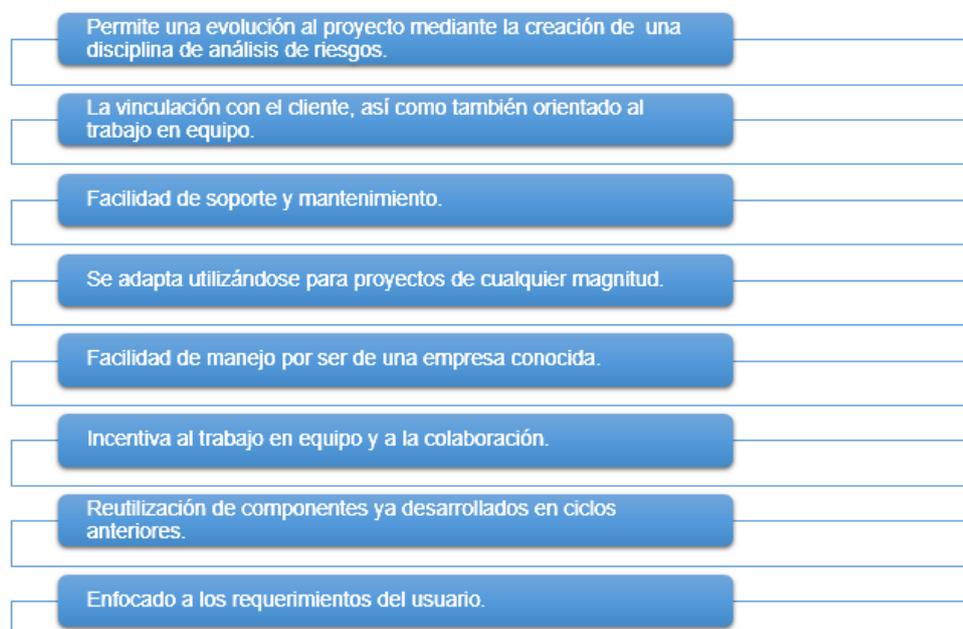
Esto facilitara la elección de la metodología de desarrollo, mediante ello la tabla 14 lista un conjunto de características que servirán para comparar las ventajas y desventajas de **rup** con **uml** y **msf**.

Tabla 14. *Análisis comparativo RUP y MSF.*

Característica	Metodología	
	RUP con UML	MSF
Desarrollo de aplicaciones web	Alto	Alto
Interacción del cliente con el equipo de desarrollo	Medio	Medio
Gestión de proyectos	Alto	Alto
Definición de requerimientos	Alto	Alto
Facilidad para desarrollos de corta duración	Alto	Medio
Uso de herramientas libres	Alto	Bajo
Experiencia de equipo de desarrollo	Medio	Medio

Nota: Obtenido de <https://goo.gl/aEQcpQ>

#### *Ventajas.*



*Figura 16.* Ventajas de RUP.  
Fuente: <https://goo.gl/aEQcpQ>

Es una metodología, que puede ajustarse a equipos de trabajo compuestos a partir de tres o más personas (Figura 16).

#### *Desventajas.*

Debido a tener la tecnología de Microsoft obliga a usar sus propias herramientas.

El proyecto se puede retardar, en el caso que el análisis de riesgos se haga muy extenuante, los elevados precios en las licencias, capacitación y soporte de Microsoft, se tiene alto grado de dependencias de tecnologías propietarias.

### **Metodologías de desarrollo ágil.**

Sepúlveda (2016) nos indica que “El concepto ágil definido como el conjunto de buenos valores y buenas prácticas para el desarrollo de proyectos de software, también se lo denomina metodologías ágiles o métodos ágiles. Las metodologías ágiles están basadas y fundamentadas en el manifiesto ágil, que es el resultado del trabajo de un grupo de expertos, siendo estos los mismos creadores de las metodologías ágiles. Teniendo como objetivo acordar y definir valores que ayuden a los equipos de trabajos a desarrollar software de manera eficiente, rápida y con adaptación a los cambios”.

El manifiesto ágil, presenta cuatro valores: individuos e interacciones: sobre procesos y herramientas, software funcionando: sobre documentación extensiva.

Colaboración con el cliente: sobre negociación contractual, respuesta ante el cambio: sobre un plan a seguir.

Los cuatro valores dieron origen a 12 principios ágiles que los podemos encontrar en el manifiesto ágil. Estos se definen de la siguiente forma:

De acuerdo a la entrega a tiempo y de manera continua del software, el cual tenga valor dando como la prioridad más importante es satisfacer al cliente.

Los procesos ágiles utilizan el cambio para aportar ventaja competitiva al cliente. Se puede aceptar que los requisitos se modifiquen, inclusive en etapas avanzadas en el desarrollo.

Se entrega el software funcional frecuentemente; los plazos están acordados entre dos semanas y dos meses preferentemente en el periodo de tiempo menos corto posible.

Los responsables del negocio, trabajan juntos en forma cotidiana durante todo el transcurso del proyecto.

En este tipo de proyectos se desarrollan en torno a los individuos que tienen motivación; dándoles el entorno necesario, apoyo antes sus necesidades y confianza en la ejecución del trabajo.

Entre otras metodologías más conocidas o llamados marcos de trabajo más usados se encuentran:

### **Scrum.**

Cuenta con roles y responsabilidades; poniendo mucho énfasis en el trabajo en equipo y la colaboración de todos los stakeholders, esto lográndose a través de la auto organización. Favoreciendo la visibilidad del producto. Se espera no tener asuntos, problemas ocultos u obstáculos que puedan poner en peligro el proyecto. Los integrantes del equipo, se guían por su experiencia y conocimiento, más que por planes de proyecto formalmente definidos.

La figura 17 muestra el marco de trabajo SCRUM.

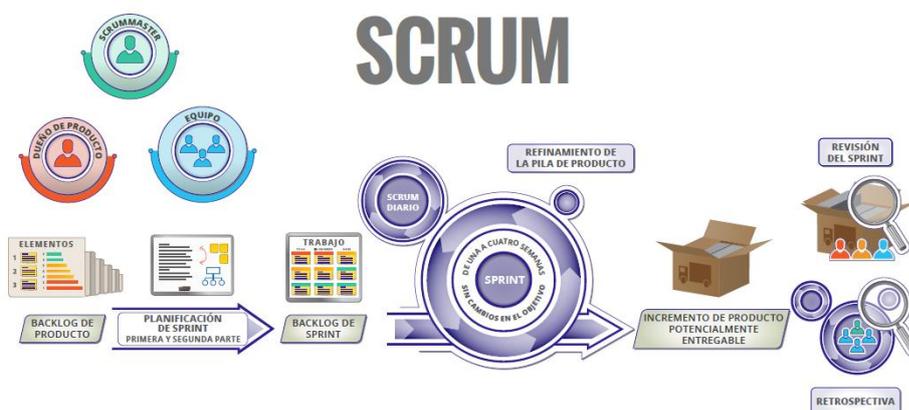


Figura 17. Marco de trabajo SCRUM.

Fuente: Blog vBote, en su artículo *Gestión de proyectos, metodología de desarrollo ágil: Scrum*. Recuperado de <https://goo.gl/JPgpva>

Según Guillermo Navarro (2017), indica cada componente del Scrum cuenta con un propósito y es importante para el éxito del proyecto. Las reglas de Scrum crean una relación entre los artefactos, roles y eventos a fin de gobernar las interacciones entre ellos”:

*Eventos.*

### **Scrum Diario.**

Es una reunión matutina (no mayor a 15 minutos) que se da en equipo y su objetivo es establecer las metas diarias de desarrollo y mostrando los avances logrados con respecto a las metas del Scrum diario anterior.

Los ítems a tratar en cada Scrum diario suelen ser:

¿Qué hicimos ayer para lograr el objetivo del sprint?

¿Qué haremos hoy para lograr el objetivo del sprint?

¿Cuáles son los problemas que impiden que logremos el objetivo del sprint?

El equipo utiliza el Scrum diario para evaluar el avance hacia el objetivo del sprint. Permitiendo optimizar las posibilidades de cumplimiento del sprint. El rol del Scrum Master dirige la realización de los scrum diarios, pero son los miembros del equipo quienes dirigen la reunión.

### **Sprint.**

Es un rango de tiempo definido (no mayor a un mes) en el cual se desarrollara un incremento del entregable funcional. Los sprint trabajan de manera secuencial es decir para empezar el siguiente sprint hay que terminar el anterior.

**Sprint Goal.** Es la meta u objetivo a cumplir en cada sprint, que será alcanzada mediante el desarrollo del product backlog. Sirve de guía para informar al equipo la razón del desarrollo del sprint. Es creado en la planificación del sprint.

*Artefactos.*

### **Product Backlog.**

Es un listado de todo lo necesario para el desarrollo del proyecto, incluye los requerimientos, características, funcionalidades, cambios, etc. es responsabilidad del Product Owner su mantenimiento y actualización.

## **Sprint Backlog.**

Son un conjunto de pendientes tomados del product backlog, incluyendo un plan para el desarrollo y el objetivo del sprint. Es un estimado elaborado por el equipo de desarrollo que indica que funcionalidad estará aplicada para el siguiente incremento y que será lo necesario para poder terminarlo.

*Roles.*

### **Scrum master.**

Personal encargado de liderar el equipo de desarrollo, haciendo cumplir las reglas y procesos de la metodología, actúa también como entrenador para los nuevos miembros del equipo.

### **Equipo de desarrollo (team).**

Está compuesto por un grupo multidisciplinario de personas que trabajaran en pro del desarrollo de un proyecto de software, cumpliendo con las historias que se han comprometido a desarrollar al comenzar mediante cada sprint.

### **Product owner.**

Es el Stakeholder designado a trabajar con el equipo el cual proporcionara los requerimientos, aprobar los objetivos del sprint, maximizar el valor del producto final.

### **Kanban.**

Sepulveda (2016), manifiesta que es una “Metodología de desarrollo que recoge la experiencia del Kanban de “Toyota”, la cual es llevada al mundo del desarrollo tecnológico, siendo su expresión más común un tablero dividido en columnas (ver figura 18) que señalan los estados de un flujo de trabajo e indicadores visualizadas en tarjetas que van señalizando como fluyen los requerimientos dentro de un equipo de software”.

Indica las siguientes propiedades fundamentales:

Visualiza el flujo de trabajo

Limita el trabajo en progreso.

Manejar y medir el flujo de trabajo.

Expone las políticas del proceso.

Usa modelos para mejorar las oportunidades de mejora.

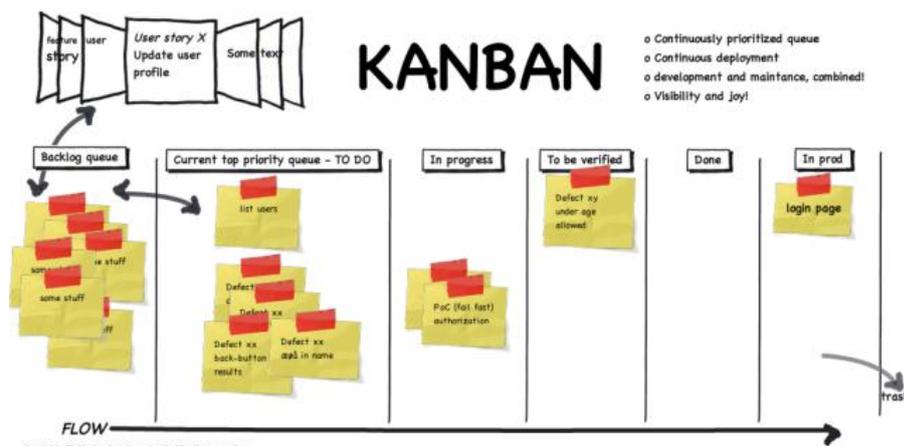


Figura 18. Tablero kanban.

Fuente: PMOinformatica.com, en su artículo *El sistema Kanban en el desarrollo de software*. Recuperado de <https://goo.gl/fKDc1V>

Podemos mencionar algunas de las características claves que hacen destacar a Kanban:

Puedes generar valor con poco aprendizaje: “La capacitación de una persona al uso del Kanban se reduce a enseñarle los códigos propios del equipo. Como en Kanban las políticas del proceso son explícitas el acceso y la comprensión de éstas se simplifica y disminuye los tiempos necesarios para que la persona entre en régimen”.

Es de simple construcción y evolución: “Al ser un tablero distribuido por zonas, armar un Kanban puede tomar menos de cinco minutos en su versión más simple”. Según Kniberg & Skarin (2010) nos indica que, “Cambiar la distribución del tablero es sencillo y permite que ésta evolucione ajustándose a las necesidades que el equipo desee implementar en su flujo de trabajo. El tablero Kanban físico posee la flexibilidad necesaria para adaptarse a las necesidades del equipo que lo utilizará”.

Tiene una baja inversión inicial y operacional: Para construir un tablero Kanban, resulta ser muy barato porque puede ser construido con materiales que se encuentran en cualquier oficina. Por decir, una versión sencilla puede construirse sobre una muralla con cinta adhesiva de color para crear la distribución de zonas. Para practicarlo un Kanban se necesita un lápiz y papeles de colores (post-it), que son de bajo costo en el mercado.

La orientación para la generación de valor: La filosofía tras el Kanban sostiene que el flujo de las tareas/actividades se debe hacer en base a “jalarlas” de un estado a otro, concepto que se conoce como modelo Pull.

Este modelo, nos da una visión clara de la priorización de las tareas/actividades basadas en la cantidad de valor generado al cliente, garantizando que sólo se construirá lo que se necesite evitando la acumulación de tareas/actividades y evitando el desperdicio de tiempo.

### ***Scrumban.***

Combina las estrategias de Scrum y Kanban, podemos decir, el desarrollo sigue un flujo de trabajo continuo a la vez que se llevan a cabo pequeñas iteraciones (sprints), se logra planificar y revisar, junto con iteraciones (sprints) más largas usadas para los lanzamientos de los productos; un sprint tiene una duración determinada por el equipo de trabajo; para este caso para los sprints cortos la duración es de una semana.

En Scrum, se definiría un conjunto de tareas a hacer acabadas en dicho sprint, en nuestro caso, al usar Scrumban dicho “compromiso” no se lleva a cabo, y se van desarrollando tareas bajo demanda y según vayan evolucionando. Los sprints semanales son en su gran mayoría de control, de esta manera se puede tener un seguimiento continuado y real sobre la evolución de las tareas y así ayudar a la priorización de las tareas pendientes a desarrollarse.

El motivo de disponer de estos cortos sprints, ayuda a conseguir un feedback continuo por parte del cliente, consiguiendo así que el producto se adapte al máximo a sus necesidades. Con esta metodología, nos permite que se apliquen cambios sobre ella.

A continuación se describen algunos de los conceptos más importantes y como Scrumban se ha aplicado al proyecto. Iteraciones (Sprints); El objetivo principal de estas divisiones es el tener contacto y revisión continua con el cliente; de esta manera cualquier tipo de problema se puede detectar y tratar con rapidez, y cualquier requisito puede ser fácilmente refinado con tal de adaptarse a las necesidades del cliente. Esta forma de revisiones semanales es ideal para el desarrollo de proyectos de investigación e innovación, ya que las tecnologías

evolucionan muy rápidamente y en cualquier momento las prioridades del proyecto pueden cambiar, asimismo, ésta facilita los cambios y adaptación.

Al inicio, se lleva a cabo un planning meeting con tal de definir el alcance del proyecto, se acuden todas las partes involucradas en el proyecto (negocio y desarrolladores). Las funcionalidades a desarrollar se dividen funcionalmente en tarjetas y se añaden al backlog (tareas pendientes) del board de trabajo (panel que contiene las tarjetas con tareas). Las reuniones se llevarán a cabo cada vez que el backlog quede vacío.

### **Daily Standup**

Al comenzar cada día, cada equipo de desarrollo con sus integrantes lleva a cabo una reunión de como máximo 10 minutos. Esta reunión recibe el nombre de Daily Standup, que viene del hecho de reunirse a diario estando de pie. Cada miembro del equipo hace un resumen de las actividades llevada a cabo el día anterior y de lo que se pretende hacer en el día actual. Asimismo, manifestar cualquier tipo de bloqueo que pueda haber y que este interfiera con la evolución de tareas/actividades se comenta a todos. Por tanto, este comentario es solo informativo, no se trata de desbloquear el problema en ese momento, tan solo de informar y si acaso, buscar ayuda por parte de algún compañero(a). Este tipo de reuniones ayudan a la motivación del equipo ya que se puede ver día a día como las tareas van evolucionando, a parte de su funcionalidad básica, que es la de que todo el equipo esté informado y al día de lo que están llevando a cabo sus compañeros generando así una gran coordinación entre el equipo.

Tablero de tareas: Es un tablero que permite ver el estado de cada tarea en el momento. Al estar todo en un mismo tablero permite la visualización del mismo completamente con tan solo una vista rápida. Nos permite ver qué tareas se están llevando a cabo en el momento, quién las está desarrollando, las tareas pendientes y las tareas ya completadas. Este tablero está formado por cuatro columnas:

Backlog: Tareas listas para ser realizadas.

Doing: Tareas que cada miembro está llevando a cabo.

Review: Tareas completadas pendientes de ser revisadas.

Done: Tareas completadas y revisadas.

## Historias de usuario

Los requisitos funcionales del proyecto se han creado/definido usando las historias de usuario. Es decir, cada uno de estos elementos define una funcionalidad que deberá de tener el sistema. Cada historia de usuario consiste en un par de frases descriptivas sobre la funcionalidad desde el punto de vista del usuario. El título de la misma ha de ser corto y conciso.

La historia de usuario, incluye los criterios de aceptación expresados desde lo que ve el usuario. En las historias de usuario ha de quedar claro el motivo y la funcionalidad que se espera lograr. Las historias de usuario pueden ser más o menos complejas, dependiendo de la granularidad o detalle con que se quiera tratar. Las historias de usuario, en Scrumban no se pre-asignan a determinados componentes del equipo, este se puede asignar o eligiendo sobre la marcha y según lo requiera el proyecto. El volumen de trabajo está enfocado a que se puede llevar a cabo paralelamente es delimitado determinando un número máximo de tareas que se pueden estar desarrollando a la vez.

A continuación un cuadro comparativo entre scrum, kanban, scrumban (Tabla 15):

Tabla 15. *Comparativa entre scrum, kanban, scrumban.*

	KANBAN	SCRUM	SCRUMBAN
<b>Tiempo para cada iteración recomendado</b>	Flujo constante. Se liberan entregables con base a eventos o necesidades	2 a 4 semanas	Flujo constante. Se liberan entregables con base a eventos o necesidades
<b>Tamaño del equipo</b>	Sin prescripción	Todos los tamaños (Scrum de Scrum)	No hay ScrumMaster
	Informal	Informal	Informal
	Cara a cara	Reuniones diarias de pie.	Reuniones diarias de pie.
<b>Comunicación en el equipo</b>	A través de tablero Kanban Cada equipo define sus procesos	Reuniones de retrospectiva.	A través de tablero Kanban  Reuniones de retrospectiva
<b>Tamaño del proyecto</b>	Proyecto de corta-media duración	Todo tipo de proyectos	Proyecto de corta-media duración
<b>Involucración del cliente</b>	Sería conveniente una participación activa del cliente	Cliente fuertemente involucrado a través del rol de ProductOwner	Cliente involucrado Comunicación fluida
<b>Documentación en el proyecto</b>	Evita la documentación Cada equipo define sus procesos	Solo documentación básica	Solo documentación básica
<b>Habilidades especiales</b>	Tablero Kanban	Sprint, product y sprint backlog, scrum board, scrum master, planning poker	Lo mejor de Scrum y de Kanban
<b>Ventajas</b>	Control del flujo de trabajo  • Fácil de aplicar  • Flexible ante cambios	• Facilita el que todo el mundo sepa hacer, quien hace qué y cómo va el proyecto  • Limita el número de tareas a hacer el mismo tiempo.  • El equipo sabe que tiene que hacer todo el día  • El cliente sabe lo que se le entrega en cada sprint <input type="checkbox"/> Flexible a cambios • Desarrolladores tienen autonomía • Minimiza el trabajo de gestión • Minimiza el síndrome del estudiante	Todas las de Kanban, más todo lo que aportan las reuniones de Scrum.
<b>Desventajas</b>	• Se deben definir las fases del ciclo de trabajo  • No define roles, ni fases, ni tampoco profundiza en el tablero Kanban • Algunos lo ven más como una técnica que como metodología	• Puede producir stress al sentirse el equipo en continuo sprint  • Requiere un equipo formado, motivado y con cierta experiencia • Necesidad de involucrar al cliente  • Problemas en equipos distribuidos geográficamente	• Algunos lo ven como Scrum solo que usando un tablero Kanban  • No hay una clara definición de roles y fases

Nota: Obtenido de <https://goo.gl/WjoM3Y>

## OBJETIVOS E HIPÓTESIS

### Objetivos

#### Objetivo general.

Determinar el impacto de la implementación de ISO/IEC 12207:2008 en el proceso de desarrollo de software de la empresa TCI.

#### Objetivos específicos.

1. Determinar el impacto de la implementación de ISO/IEC 12207:2008 en el número de incidencias de la empresa TCI.
2. Determinar el impacto de la implementación de ISO/IEC 12207:2008 en el número de quejas de la empresa TCI.
3. Determinar el impacto de la implementación de ISO/IEC 12207:2008 en la rotación de personal de la empresa TCI.

### Justificación de la investigación

#### Justificación teórica.

“El proceso de desarrollo de software”, es uno de los procesos más importantes de la empresa TCI debido a que brinda servicios que incluyen el desarrollo y mantenimiento de aplicaciones. Por lo tanto es importante aplicar un estándar o norma que nos brinde pautas para controlar y mantener todo lo relacionado referentemente al “ciclo de vida del software”, desde el inicio de la conceptualización de ideas, desarrollo, operación y mantenimiento de productos de software de lo cual se encarga la norma ISO/IEC 12207:2008, el objetivo principal es proporcionar una estructura común para que los desarrolladores, personal de mantenimiento, operadores, gestores y técnicos involucrados en el desarrollo de software utilicen un lenguaje en común conocido por el equipo.

La implementación de “ISO/IEC 12207:2008”, nos permitirá definir y mejorar los procesos asociados al “ciclo de vida del software” con el fin de desarrollar productos de alta calidad, lográndose en un tiempo adecuado, costos

competitivos, actividades de operación y mantenimiento menos complejas y así aumentar considerablemente la satisfacción de los clientes y usuarios finales.

### **Justificación práctica.**

Los resultados demostrarán una mejora en la productividad en las áreas de Desarrollo, Soporte y Mantenimiento entregando productos a tiempo y de alta calidad, lo cual se verá reflejado en la satisfacción del cliente.

La norma permite definir “como” un proceso será ejecutado lo cual brinda flexibilidad para implementarla con la tecnología que se considere pueda apoyarnos en esta implementación.

### **Justificación social.**

La población con la que se beneficiará el presente estudio son:

La empresa TCI, con un beneficio en la productividad de las áreas de Desarrollo, Soporte y Mantenimiento debido a que la implementación del estándar ISO/IEC 12207:2008 permitirá que se puedan brindar entregables con calidad y a tiempo, lo cual evitará las quejas de los clientes que se han venido presentando de manera recurrente y creciente en los últimos años.

Los empleados, mejorará la motivación de los empleados de las áreas de Desarrollo, Soporte y Mantenimiento al trabajar con un estándar que les permita trabajar de manera más ordenada, evitando la duplicidad de procedimientos para un mismo requerimiento.

Los clientes, la empresa mejorará la satisfacción al cliente porque se entregaran productos con alta calidad.

## **Hipótesis**

### **Hipótesis general.**

“La implementación de ISO/IEC 12207:2008 generará en TCI un impacto positivo en el proceso de desarrollo de software”.

**Hipótesis específica.**

1. La implementación de “ISO/IEC 12207:2008” reducirá el número de incidencias en la empresa TCI.
2. La implementación de “ISO/IEC 12207:2008” reducirá el número de quejas en la empresa TCI.
3. La implementación de “ISO/IEC 12207:2008” reducirá la rotación en la empresa TCI.

**Alcances y Limites****Alcances.**

El presente estudio abarca las áreas de “Desarrollo y Soporte y Mantenimiento” de la empresa TCI.

La implementación de “ISO/IEC 12207:2008”, para la mejora del proceso de desarrollo de software en las áreas Desarrollo y Soporte y Mantenimiento las cuales incluirán todos los grupos de “proceso del ciclo de vida”.

La implementación se realizará de la mano con apoyo de herramientas y marcos de trabajo que nos facilitaran la gestión de los nuevos proyectos y el mantenimiento de los mismos.

Se involucrará a todo el personal que conforme el proyecto, mediante la capacitación y definición de procedimientos con los líderes de equipo, quienes a su vez deberán hacer llegar la información a sus subordinados.

**Limitaciones.**

La presente investigación se vio limitada por el tiempo necesario para generar los datos pertinentes, al considerar el mes como unidad de medida, esto llevó a que nuestra muestra fuera reducida a un año, dividido en partes iguales según periodo de control y experimental.

Por otra parte, no se contaba con mecanismos de medición que relacionaran las variables intervenidas con ahorro de efectivo. Para superar este impase, se utilizó el método de estimación de Fermi.

## Matriz de Consistencia

IMPLEMENTACIÓN DE ISO/IEC 12207:2008 PARA LA MEJORA DEL PROCESO DE DESARROLLO DE SOFTWARE EN LA EMPRESA TCI							
Problema General	Objetivo General	Hipótesis General		Variables	Método		
¿Cuál es el impacto de la implementación de ISO/IEC 12207:2008 en los resultados del proceso de desarrollo de software de TCI?	Determinar el impacto de la implementación de ISO/IEC 12207:2008 en el proceso de desarrollo de software de la empresa TCI.	H1	La implementación de la ISO/IEC 12207:2008 generará en TCI un impacto positivo en el proceso de desarrollo de software.	Independiente: Implementación de ISO/IEC 12207:2008	<b>Tipo de Estudio:</b> Explicativo y Descriptivo.		
		H0	La implementación de la ISO/IEC 12207:2008 no generará en TCI un impacto positivo en el proceso de desarrollo de software.		<b>Diseño:</b> Experimental.		
Problemas Específicos	Objetivos Específicos	Hipótesis Específicas			Población: Resultados del sistema de indicadores de la empresa TCI.		
1	Determinar el impacto de la implementación de ISO/IEC 12207:2008 en el número de incidencias de la empresa TCI?	H1	La implementación de ISO/IEC 12207:2008 reducirá el número de incidencias en la empresa TCI.			Muestra: Resultados del sistema de indicadores 6 meses antes y después de la implementación del ISO.	
		H0	La implementación de ISO/IEC 12207:2008 no reducirá el número de incidencias en la empresa TCI.				
2	Determinar el impacto de la implementación de ISO/IEC 12207:2008 en el número de quejas de la empresa TCI?	H2	La implementación de ISO/IEC 12207:2008 reducirá el número de quejas en la empresa TCI.				Dependiente: Quejas, Incidentes, Rotación de Personal
		H0	La implementación de ISO/IEC 12207:2008 no reducirá el número de quejas en la empresa TCI.				
3	Determinar el impacto de la implementación de ISO/IEC 12207:2008 en la rotación de personal de la empresa TCI?	H3	La implementación de ISO/IEC 12207:2008 reducirá la rotación en la empresa TCI.	<b>Procedimiento recolección de datos:</b> La información fue recolectada por los departamentos administrativos de recepción y canalización de incidencias de la empresa TCI.			
		H0	La implementación de ISO/IEC 12207:2008 no reducirá la rotación en la empresa TCI.		<b>Análisis Estadístico:</b> Las quejas e incidencias fueron analizadas a un nivel descriptivos, evakando los indicadores de media, coeficiente de variación de medias y desviación estandar.		

Figura 19. Matriz de consistencia.

Fuente: Elaboración propia.

## METODO Y MARCO METODOLOGICO

### Tipo de Investigación

La investigación será de tipo cuantitativa básica, al contribuir al estado del conocimiento en la materia mediante la recolección y análisis de datos (Hernández, Fernández y Baptista, 2006).

### Diseño de la Investigación

En cuanto a su diseño, esta será experimental y de campo, al manipular una variable a criterio del investigador para analizar su impacto en una sola medida antes y después en las operaciones de una empresa de tecnologías de la información (Hernández, Fernández y Baptista, 2006).

### Variables

#### Variable Independiente.

Uso de ISO/IEC 12207:2008, manipulada mediante la implementación de los estándares de dicho ISO. Esta variable podrá tomar los valores de "Implementado" y "Sin implementar".

#### Variables Dependientes.

Se evaluará el impacto de la variable independiente en el *Número de quejas*, *Número de incidencias* y *rotación de personal*.

El número de incidencias se operacionaliza como el número de fallos en el software reportados por el cliente al departamento de soporte para su revisión.

El número de quejas se entiende como la cantidad de incidencias que causan suficiente incomodidad al cliente para ser escaladas a departamentos administrativos de la organización.

La rotación de personal se refiere al número de despidos o la renuncias de trabajadores cada mes.

## **Población y Muestra**

### **Población.**

Se consideraron las quejas e incidencias y rotación de las operaciones de la empresa TCI durante los meses de Enero – Diciembre 2017.

### **Muestra.**

Siendo el mes considerado como la unidad de análisis, se consideraron todos los meses comprendidos durante los meses de Enero – Diciembre 2017. Al final de la recolección, la distribución de las observaciones obtenidas se configuró como se presenta en la tabla 16.

Tabla 16. *Distribución de muestra.*

Variables	Sin Implementar	Implementado
	f	f
Número de Quejas	6	6
Número de Incidencias	6	6
Rotación de Personal	6	6

## **Técnicas e Instrumentos de Investigación**

### **Técnicas.**

Para la recolección del número de quejas e incidencias, se hizo uso del registro de indicadores de la empresa TCI, obteniendo los recuentos mensuales de cada variable.

### **Instrumentos.**

Registro de indicadores de la empresa TCI se utilizó al ser considerado un registro confiable de las variables evaluadas en cada unidad de análisis.

### **Procedimientos de recolección de datos.**

Para la recolección de incidencias, quejas y rotación, se generó un reporte de frecuencias mensuales desde la base de datos en excel de la empresa. Los resultados fueron digitados en una base de datos Excel para su posterior exportación, depuración y codificación en SPSS.

### **Método de análisis.**

Los datos obtenidos para el número de quejas, incidencias y rotación fueron resumidos descriptivamente en una tabla según el estado de la implementación. Estos fueron comparados de manera descriptiva para generar conclusiones.

Los resultados fueron analizados para la elaboración de conclusiones.

## RESULTADOS

Como se observa en la tabla 17, la media de incidencias tuvo una disminución de 25%, así como una disminución del 63% en su desviación estándar posteriores a la implementación del ISO. Por otra parte, encontramos que la distribución de datos de la variable quejas después de la implementación mostraría poca conglomeración de resultados.

Tabla 17. *Quejas, incidencias y rotación por implementación de ISO*

Variables	Control	Media	Desv. Est	TVM	TVDE	g1	g2
Incidencias	No implementado	37.67	16.16	-25%	-63%	0.82	-1.11
	Implementado	28.33	5.96			0.72	0.40
Quejas	No implementado	3.83	1.60	-30%	-24%	-0.04	-1.31
	Implementado	2.67	1.21			-0.08	-1.55
Rotación	No implementado	1.00	0.89	-17%	-16%	0.00	-1.88
	Implementado	0.83	0.75			0.31	-0.10

Nota: n=12; g1: Asimetría; g2: Curtosis; TVM: Tasa de variación de medias; TVDE: Tasa de variación de desviación estándar.

Con respecto a las quejas, se encontró que la media muestral disminuyó en un 30%, así como un 24% en su desviación estándar posterior a la implementación del ISO.

Finalmente, en cuanto a la rotación, se encontró una disminución de media del 17% y un 16% de dispersión.

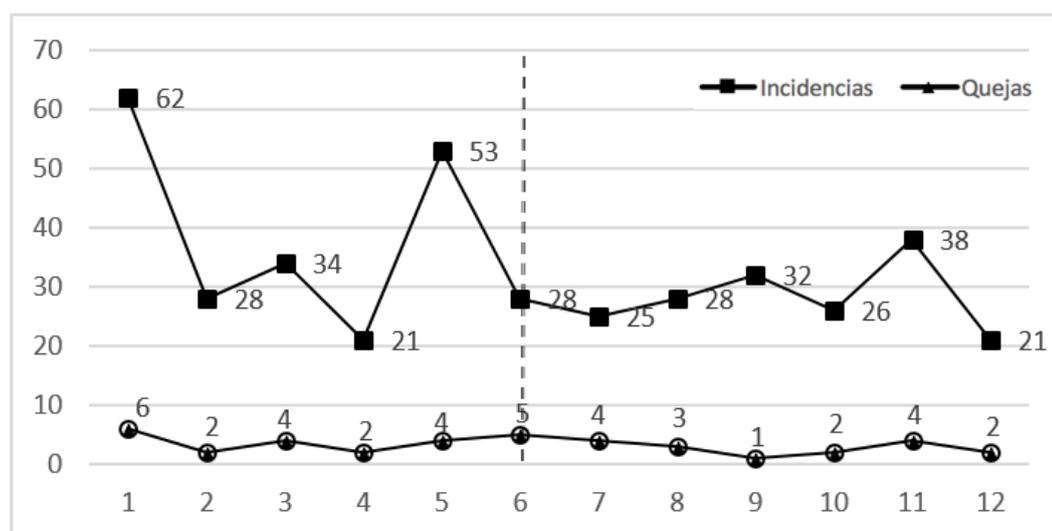


Figura 20. Dispersión de quejas e incidencias.  
Fuente: Elaboración propia.

Además, en la figura 20 se puede observar que, para las variables de quejas e incidencias después de la intervención, los resultados disminuyen su dispersión inmediatamente, aumentando una vez más cerca del cierre del año.

Finalmente, haciendo uso del método de la estimación de Fermi, se elaboró la siguiente ecuación para el cálculo estimado del ahorro en base a la modificación de variables fruto de la implementación.

$$(S/ 87.5 * \Delta \text{Incidencias} * 12 \text{ meses}) + (S/ 5250 * \Delta \text{Quejas} * 12 \text{ meses}) + (S/ 33600 * \Delta \text{Rotación})$$

$$(S/ 87.5 * -25\% * 12) + (S/ 5250 * -30\% * 12) + (S/ 33600 * -17\%) = -24874.5$$

Reemplazando las variables, se obtiene que el ahorro anual estimado por la implementación del ISO en estas variables es de S/ 24874.5. Considerando que el costo por la implementación fue de S/ 25384 (véase anexo 21) se tuvo un retorno a la inversión de -S/ 509.5 (véase anexo 22).

## DISCUSIONES

El objetivo de la presente investigación fue determinar el impacto de la implementación del “ISO/IEC 12207:2008” en las incidencias, quejas y rotación de personal de una empresa de tecnología de la información.

Si bien el estudio se enfoca en el estudio de la influencia desde un enfoque experimental, resulta pertinente añadir que el proceso de implementación presentó limitaciones e inconvenientes similares a las presentadas en otros estudios antecedentes.

En congruencia con Gallegos y Ortiz (2011), la implementación total del ISO resultó demasiado impactante en las operaciones como para ejecutarla de manera inmediata, fue necesaria una implementación de procesos de manera gradual para la aceptación de los nuevos procedimientos.

Por otra parte, en congruencia con Pacheco (2016), se evidenciaron mejoras en las variables de negocio evaluadas, no solamente disminuyendo la media mensual, si no la dispersión de los resultados. El mejoramiento de la media y la disminución de la dispersión son prueba de que el proceso efectivamente tuvo un impacto en la calidad del desarrollo de software, al tener un rango de resultados más predecible. Tanto las quejas como las incidencias y la rotación tuvieron un impacto similar en sus parámetros, lo cual brinda evidencia del efecto aislado de la intervención sobre las variables.

Finalmente, si bien la metodología de estimación de Fermi es susceptible al error humano, resultó útil para poder otorgar valores monetarios al impacto de la intervención; considerando que no se contaba con los sistemas de medición adecuados para contar con dicha información. Sin embargo, se encontró un retorno a la inversión negativo en la evaluación de estas variables, esto se debe a que la evaluación del ahorro consideró un espectro de variables limitado, dejando de considerar otras fuentes de ahorro importantes como la percepción del mercado, la marca empleadora, la optimización del uso de los recursos de la empresa o la mejora en el clima laboral.

## CONCLUSIONES

1. La implementación del ISO tuvo un impacto positivo en las medias incidencias, quejas y rotación, disminuyendo las mismas a razón de 25%, 30% y 17%. Se puede concluir que la implementación del ISO tuvo un impacto favorable sobre las variables evaluadas en el presente estudio.
2. La implementación del ISO requiere una implementación gradual para no afectar la operatividad del área. Se debe priorizar los objetivos de la empresa para poder elegir las fases o procesos del ISO a implementar, buscando no afectar la continuidad de las operaciones de la organización.
3. La implementación del ISO disminuyó la dispersión de los resultados de incidencias, quejas y rotación mejorando la predictibilidad de su aparición. La variable con el mayor impacto luego de la implementación fue la de incidencias, disminuyendo su dispersión en un 63%. Se puede concluir que la implementación del ISO tuvo un impacto favorable sobre la dispersión de los resultados de las variables evaluadas.
4. La implementación del ISO generó un ahorro estimado de S/ 24874.5 en el primer año. Considerando que el costo de implementación fue de S/ 25384, el retorno de la inversión será positivo a partir del segundo año.

## RECOMENDACIONES

En relación a las conclusiones presentadas, se recomienda lo siguiente.

1. Al encontrar un efecto positivo sobre variables de negocio que repercuten en el margen de utilidad del proyecto, se recomienda la implementación del ISO para mejorar el rendimiento de los equipos de desarrollo de software. Sin embargo, debido al tamaño de la muestra de incidencias y quejas no se pudieron aplicar pruebas inferenciales, por lo que además se recomienda mayor investigación para dar predictibilidad a los resultados.
2. Se debe prestar atención a la adecuada aceptación e integración del nuevo procedimiento en las operaciones del área de software intervenida. Una inadecuada gestión del cambio puede conllevar a que los nuevos procesos sean implementados, pero no utilizados para su verdadero propósito. La implementación se puede ver fragmentada según las necesidades del área.
3. Al igual que el impacto en el rendimiento de los equipos de desarrollo, la implementación funcionó como una manera de estandarizar los resultados. De esta manera, se recomienda continuar con la implementación del ISO para mejorar la calidad de las operaciones del área de desarrollo.
4. Si bien el uso del método de estimación de Fermi tiene un uso habitual en la investigación científica, es necesario realizar mayor investigación en diferentes empresas y promover el desarrollo de sistemas de inteligencia de negocios que permitan el cálculo del impacto económico de la variación de indicadores de negocio.

## REFERENCIAS

- Alegre, J. (2015). *Mejora de proceso software en una pequeña organización desarrolladora de software: caso PROCAL-PROSER-LIM. Nu-1er ciclo*. Pontificia Universidad Católica del Perú. Recopilado de <http://tesis.pucp.edu.pe/repositorio/handle/123456789/6365>
- Aparcana Ramos, L. M. & Zavala Quintana, A. C. (2014). *Modelo de mejora de procesos para la calidad del software basado en CMMI para una entidad financiera*. Universidad San Martín de Porres – Lima, Perú. Recopilado de [http://www.repositorioacademico.usmp.edu.pe/bitstream/usmp/1151/1/aparcana\\_r.pdf](http://www.repositorioacademico.usmp.edu.pe/bitstream/usmp/1151/1/aparcana_r.pdf)
- Blog vBote (2014), *en su artículo Gestión de proyectos, metodología de desarrollo ágil: Scrum*. Recuperado de <http://www.vbote.com/vbote-solutions-academy-blog/86-gestion-de-proyectos-metodologia-de-desarrollo-agil-scrum.html>
- Brito, A. K. (2009), *Metodologías tradicionales y metodologías ágiles: Eumet.net*. Recuperado de <http://www.eumed.net/libros-gratis/2009c/584/Metodologias%20tradicionales%20y%20metodologias%20agiles.htm>
- Cardona Velásquez, C. (2009), *Propuesta Metodológica para la Realización de Pruebas de Software en un Ambientes Productivos*. Universidad Nacional de Colombia. Recuperado de: [http://www.bdigital.unal.edu.co/930/1/8357252\\_2009.pdf](http://www.bdigital.unal.edu.co/930/1/8357252_2009.pdf)
- Collantes Chules, H. (2008), *NTP-ISO/IEC 12207 como un marco de referencia a los procesos del ciclo de vida del software*. Recuperado de: <http://www.monografias.com/trabajos99/articulo-ntp-iso-iec-12207/articulo-ntp-iso-iec-12207.shtml>
- Díaz Peña, G. A. (2015). *Mejora de proceso software en una pequeña organización desarrolladora de software: caso PROCAL-PROSER LIM. Epsilon-1er ciclo*. Pontificia Universidad Católica del Perú. Recopilado de: <http://tesis.pucp.edu.pe/repositorio/handle/123456789/6365>

Gallegos Velásquez A. & Ortiz Rodríguez P. A. (2011), *Elaboración del estándar de aplicación de la norma ISO/IEC 12207, al desarrollo de aplicaciones de software para la UTIC de la ESPE*. Sangolquí, Ecuador. Escuela Politécnica del Ejército. Recopilado de: <https://repositorio.espe.edu.ec/bitstream/21000/4275/1/T-ESPE-032633.pdf>

García-Paucar, L. (2009), *MEDESOFTE: Metodología de Desarrollo de Software en Entidades de Educación Superior*. Universidad de Piura, Perú. Recopilado de: [https://pirhua.udep.edu.pe/bitstream/handle/11042/1823/MAS\\_DET\\_003.pdf?sequence=1](https://pirhua.udep.edu.pe/bitstream/handle/11042/1823/MAS_DET_003.pdf?sequence=1)

GrupNADD (2012), *Metodología RUP y Ciclo de Vida*. Altagracia de Orituco, Estado Guárico, Venezuela. Recuperado de: <http://rupmetodologia.blogspot.pe/>

Godoy Alvarez, D. A. & Taype Calderón, R.S. (2015), *Modelos de aceptación de Metodologías de desarrollo de software*. Universidad Peruana de Ciencias Aplicadas (UPC) Lima, Perú. Recuperado de [http://repositorioacademico.upc.edu.pe/upc/bitstream/10757/528193/1/Tesis+Godoy\\_AD++Taype\\_CR.pdf](http://repositorioacademico.upc.edu.pe/upc/bitstream/10757/528193/1/Tesis+Godoy_AD++Taype_CR.pdf)

Guillermo Navarro, R. A. (2017), *Implementación de un Sistema Web Para Las Ventas En La Empresa One To One Contact Solutions*, Universidad San Ignacio de Loyola (USIL) Lima, Perú. Recuperado de: [http://repositorio.usil.edu.pe/bitstream/USIL/2764/1/2017\\_Guillermo\\_Implementacion-de-un-sistema-web.pdf](http://repositorio.usil.edu.pe/bitstream/USIL/2764/1/2017_Guillermo_Implementacion-de-un-sistema-web.pdf)

Hernández Sampieri R. & Fernández Collado C. & Baptista Lucio P. (2006), *Metodología de la investigación*. México. Editorial McGraw Hill. Recuperado de [https://mega.nz/#!yINTFD4B!NkPy70g0vKbXjHn\\_cBqj3Ta2a3cuTFyg0Y2EoDXl3ew](https://mega.nz/#!yINTFD4B!NkPy70g0vKbXjHn_cBqj3Ta2a3cuTFyg0Y2EoDXl3ew)

Hernández (2016), *Diseños no experimentales de investigación*. Recuperado de: [eva.fcs.edu.uy/mod/resource/view.php?id=6915](http://eva.fcs.edu.uy/mod/resource/view.php?id=6915)

Horna Merino, L. (2014). *Implementación de la ISO/IEC 12207:2008 para mejorar los procesos asociados al ciclo de vida de software en una micro empresa peruana cuyo objeto social es el desarrollo de sistema de información*. Pontificia Universidad Católica del Perú. Recopilado de <http://tesis.pucp.edu.pe/repositorio/handle/123456789/6298>

Jiménez, B. & Peña, S. & Valverde, Y. & Aramayo, A. & Salazar, Iver (2012), *Metodología de Desarrollo de Software - MFS*. Universidad Autónoma Gabriel Rene Moreno. Santa Cruz, Bolivia. recuperado de: <https://www.google.com.pe/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=0ahUKEwjqsZfs6IHUAhUDRSYKHQilBhQQFggnMAE&url=https%3A%2F%2Fbeymarjimenez.files.wordpress.com%2F2012%2F10%2Fmsf-documento.docx&usg=AFQjCNH2sCBIMTJ3mePrWaqTwhivRiSUbG&sig2=gNC9JKryzvKnX1FDgZ7TfQ>

Lorenzo, S. & Alonso, G. (2011). *Mejora del proceso software de una pequeña empresa desarrolladora de software: caso COMPETISOFT-Perú Tau*. Pontificia Universidad Católica del Perú. Recopilado de <http://tesis.pucp.edu.pe:8080/repositorio/handle/123456789/357>

Luna Villagrana, A. (2014), *RUP (Rational Unified Process) Proceso Unificado Racional*. Recuperado de: <http://proceso-unificado-racional.blogspot.pe/>

Llontop Chang, J. A. (2015). *Mejora en el proceso de desarrollo de nuevas línea de productos basado en la metodología QFD asegurando la calidad del producto a lo largo de su ciclo de vida*. Universidad Peruana de Ciencias Aplicadas (UPC) Lima, Perú. Recopilado de <http://repositorioacademico.upc.edu.pe/upc/bitstream/10757/581458/2/Tesis+Llontop+Chang.pdf>

Marroquín Ortiz, K. O. (2013) *Gestión de la seguridad informática en procesos de desarrollo de software a medida. Facultad de Ciencias Económicas*. Universidad de Buenos Aires, Argentina. Recuperado de: [http://bibliotecadigital.econ.uba.ar/download/tpos/1502-0548\\_MarroquinOrtizKX.pdf](http://bibliotecadigital.econ.uba.ar/download/tpos/1502-0548_MarroquinOrtizKX.pdf)

Mesquida Calafat (2012), *Un Modelo para Facilitar la Integración de Estándares de Gestión de TI en Entornos Maduros*. Universitat de les Illes Balears - Palma, Islas Baleares, España. Recuperado de: <http://www.tdx.cat/bitstream/handle/10803/84137/talmc1de1.pdf;sequence=1>

Pacheco Tejada, R. G. (2016), *Propuesta Metodológica para la documentación de desarrollos de software basado en el ISO 12207:2008 en instituciones financieras de la ciudad de Arequipa*. Universidad Católica Santa María – Arequipa, Perú. Recuperado de <https://tesis.ucsm.edu.pe/repositorio/bitstream/handle/UCSM/6026/71.0579.IS.pdf?sequence=1&isAllowed=y>

Psyma (2015), *Como determinar el tamaño de una muestra*. Recuperado de: <http://www.psyma.com/company/news/message/como-determinar-el-tamano-de-una-muestra>

Ramírez Chira, F. (2012), *Spice/Iso/lec 15504*. Recuperado de <http://seispice.blogspot.pe/>

Sánchez Rojas, N. (2016), *Metamodelo para la definición e implantación de los procesos de planificación y control de proyectos en pequeños equipos de desarrollo*. Universidad Tecnológica de la Mixteca - Oaxaca, México. Recuperado de [http://jupiter.utm.mx/~tesis\\_dig/12998.pdf](http://jupiter.utm.mx/~tesis_dig/12998.pdf)

Sepulveda Castaño, J. M. (2016), *Propuesta de aplicación de Scrumban para gestionar el proceso de generación de proyectos de I+D+i con el modelo Canvas: Estudio Preliminar*. Recuperado de [https://repository.eafit.edu.co/bitstream/handle/10784/11355/JorgeMauricio\\_SepulvedaCasta%C3%B1o\\_2016.pdf?sequence=2](https://repository.eafit.edu.co/bitstream/handle/10784/11355/JorgeMauricio_SepulvedaCasta%C3%B1o_2016.pdf?sequence=2)

Villa, C. & Ruiz, M. & Ramos, I. (2004), *Modelos de evaluación y mejora de procesos: análisis comparativo*. Universidad de Huelva, Cádiz y Sevilla, España. Recuperado de [http://rabida.uhu.es/dspace/bitstream/handle/10272/8773/Modelos\\_de\\_evaluacion.pdf?sequence=2](http://rabida.uhu.es/dspace/bitstream/handle/10272/8773/Modelos_de_evaluacion.pdf?sequence=2)

## ANEXOS

### ANEXO 1. Cronograma y recursos de la implementación de los procesos.

Para la implementación de los dos (2) procesos, se ha estimado 50.61 días de trabajo que corresponde a 665.5 horas, el trabajo fue realizado por 5 recursos. A continuación la información de los recursos y cronograma.

#### Recursos.

Nombre del recurso	Trabajo
▶ Líder de Desarrollo/Mantenimiento	158.4 horas
▶ Líder de Soporte FE	193.6 horas
▶ Analista Programador 1 (Área de Desarrollo)	282.6 horas
▶ Analista Programador 3 (Área de Soporte y mantenimiento)	281.6 horas

### Cronograma.

Nombre de tarea	Duración	Comienzo	Fin
<b>“Implementación de iso/iec 12207:2008 para la mejora del proceso de desarrollo de software”.</b>	<b>45.83 días</b>	<b>30/03/17 3:00 p. m.</b>	<b>12/06/17 10:30 a. m.</b>
<b>Inicio</b>	<b>9.06 días</b>	<b>30/03/17 3:00 p. m.</b>	<b>13/04/17 4:30 p. m.</b>
Reunión para explicar las bondades de ISO/IEC 12207 a la Gerencia	2 horas	30/03/17 3:00 p. m.	30/03/17 5:00 p. m.
Reunión para explicar las bondades de ISO/IEC 12207 a los Jefes y equipo	2 días	3/04/17 9:00 a. m.	5/04/17 11:00 a. m.
Elaboración de encuestas sobre el uso de proceso ISO/IEC 122207	6 horas	6/04/17 12:00 p. m.	7/04/17 10:00 a. m.
Realización de encuestas sobre el uso de proceso ISO/IEC 122207	3 horas	10/04/17 11:00 a. m.	10/04/17 3:00 p. m.
Análisis del resultado de las encuestas	4 horas	11/04/17 4:00 p. m.	12/04/17 11:00 a. m.
Elaboración del Acta de Constitución del Proyecto	3.5 horas	13/04/17 12:00 p. m.	13/04/17 4:30 p. m.
<b>Planificación</b>	<b>0.78 días</b>	<b>13/04/17 4:30 p. m.</b>	<b>14/04/17 3:30 p. m.</b>
Elaborar Enunciado de Alcance del Proyecto	4 horas	13/04/17 4:30 p. m.	14/04/17 11:30 a. m.
Elaboración del Cronograma del Proyecto	2 horas	14/04/17 12:00 p. m.	14/04/17 3:00 p. m.
<b>Ejecución</b>	<b>35.5 días</b>	<b>14/04/17 3:30 p. m.</b>	<b>9/06/17 3:00 p. m.</b>
<b>Mejoras en los procesos seleccionados</b>	<b>31.83 días</b>	<b>14/04/17 3:30 p. m.</b>	<b>5/06/17 2:00 p. m.</b>
<b>Proceso de Calificación del software.</b>	<b>31.83 días</b>	<b>14/04/17 3:30 p. m.</b>	<b>5/06/17 2:00 p. m.</b>
<b>Reuniones</b>	<b>15.17 días</b>	<b>14/04/17 3:30 p. m.</b>	<b>9/05/17 4:00 p. m.</b>
R1 – Proceso de Calificación del software.	3 horas	14/04/17 3:30 p. m.	17/04/17 9:30 a. m.
R2 – Proceso de Calificación del software.	3 horas	8/05/17 9:00 a. m.	8/05/17 12:00 p. m.
R3 – Proceso de Calificación del software.	3 horas	9/05/17 12:00 p. m.	9/05/17 4:00 p. m.
<b>Documentación</b>	<b>16.67 días</b>	<b>9/05/17 4:00 p. m.</b>	<b>5/06/17 2:00 p. m.</b>
Elaboración de: Procedimiento de Calificación del sw.	40 horas	9/05/17 4:00 p. m.	16/05/17 4:00 p. m.
<b>Proceso de Resolución de Problemas de Software.</b>	<b>5.56 días</b>	<b>18/05/17 4:00 p. m.</b>	<b>26/05/17 6:00 p. m.</b>
<b>Reuniones</b>	<b>0.44 días</b>	<b>18/05/17 4:00 p. m.</b>	<b>19/05/17 11:00 a. m.</b>
R1 – Proceso de Resolución de Problemas de sw.	4 horas	18/05/17 4:00 p. m.	19/05/17 11:00 a. m.
<b>Documentación</b>	<b>5.11 días</b>	<b>19/05/17 11:00 a. m.</b>	<b>26/05/17 6:00 p. m.</b>
Elaboración de Procedimiento de Calificación del sw.	30 horas	19/05/17 11:00 a. m.	24/05/17 6:00 p. m.
<b>Implementación</b>	<b>2.33 días</b>	<b>7/06/17 9:00 a. m.</b>	<b>9/06/17 3:00 p. m.</b>
<b>Proceso de Calificación del software.</b>	<b>1.33 días</b>	<b>7/06/17 9:00 a. m.</b>	<b>8/06/17 2:00 p. m.</b>
Capacitación del procedimiento del Proceso de calificación del software al equipo correspondiente	6 horas	7/06/17 9:00 a. m.	7/06/17 4:00 p. m.
Capacitación de los formatos al equipo correspondiente	6 horas	7/06/17 4:00 p. m.	8/06/17 2:00 p. m.
<b>Proceso de Resolución de Problemas de Software.</b>	<b>1 día</b>	<b>8/06/17 2:00 p. m.</b>	<b>9/06/17 3:00 p. m.</b>
Capacitación del procedimiento del Proceso de resolución del software al equipo correspondiente	6 horas	8/06/17 2:00 p. m.	9/06/17 11:00 a. m.
Capacitación de los formatos al equipo correspondiente	3 horas	9/06/17 11:00 a. m.	9/06/17 3:00 p. m.
<b>Seguimiento y Control</b>	<b>25.89 días</b>	<b>2/05/17 9:00 a. m.</b>	<b>12/06/17 10:00 a. m.</b>
<b>Monitoreo y Control</b>	<b>25.89 días</b>	<b>2/05/17 9:00 a. m.</b>	<b>12/06/17 10:00 a. m.</b>
Conformidad de avance de elaboración del procedimiento del Proceso de calificación del software.	4 horas	16/05/17 4:00 p. m.	17/05/17 11:00 a. m.
Conformidad de avance de formatos del Proceso de calificación del software.	12 horas	5/06/17 2:00 p. m.	6/06/17 6:00 p. m.
Conformidad de avance de elaboración del procedimiento del Proceso de Resolución de Problemas de Software.	2 horas	25/05/17 9:00 a. m.	25/05/17 11:00 a. m.
Conformidad de avance de formatos del Proceso de Resolución de Problemas de Software.	4 horas	29/05/17 9:00 a. m.	29/05/17 2:00 p. m.
Conformidad sobre la implementación de los procesos de calificación Sistema y Resolución de Problemas de software	4 horas	9/06/17 3:00 p. m.	12/06/17 10:00 a. m.
<b>Cierre</b>	<b>0.06 días</b>	<b>12/06/17 10:00 a. m.</b>	<b>12/06/17 10:30 a. m.</b>
Elaboración de documento de cierre	0.5 horas	12/06/17 10:00 a. m.	12/06/17 10:30 a. m.

**ANEXO 2.** Actividades y plan de acción para la implementación del Proceso de pruebas de calificación de software.

El cuadro que se muestra a continuación describe el propósito del Proceso Resolución de Pruebas de Calificación de Software, las actividades que se deben realizar en este proceso según la norma “ISO/IEC 12207”, un indicador sobre el cumplimiento en la empresa de cada actividad y finalmente el plan de acción especificando los documentos que se elaboraron para el cumplimiento de este proceso.

Proceso de Pruebas de Calificación del Software			
Propósito	El propósito del Proceso de Pruebas de Calificación del Software es confirmar que el producto software integrado satisface sus requisitos definidos.		
El proyecto debe implementar las siguientes actividades de acuerdo con las políticas y procedimientos de la organización aplicables con respecto al Proceso de Pruebas de Calificación del Software.			
Actividades y tareas Según ISO 12207	Cumplió la empresa	Plan de Acción	
		Procedimiento	Documento
7.1.7.3.1 Prueba de calificación del software. Para cada elemento de software (o elemento de configuración, si está identificado) esta actividad consta de las siguientes tareas:			
7.1.7.3.1.1 El implementador debe realizar la prueba de calificación según los requisitos de calificación para el elemento de software. Se debe asegurar que se somete a prueba de conformidad la implementación de cada requisito. Los resultados de la prueba de calificación deben estar documentados.	NO	Se elaborará un procedimiento para la Gestión de Pruebas que incluirá lo siguiente:	- Procedimiento del Proceso de Pruebas de Calificación del Software. (Anexo 3)
7.1.7.3.1.2 El implementador debe actualizar la información del usuario, según sea necesario.		- Como fases: Análisis del Requerimiento, Diseño de pruebas, Ejecución de pruebas y Cierre de pruebas.	
		- Como entrada se solicitará la elaboración del documento de Alcance y Requisitos del sistema por parte del equipo de desarrollo.	-Enunciado de Alcance del Proyecto (Anexo 4)
		-Se considera dos tipos de estimaciones, una para pruebas del equipo de Soporte y Mantenimiento y otro para el equipo de Desarrollo.	-Estimación de pruebas. (Anexo 5)
		- Se elaborará el plan de pruebas.	-Cronograma. (Anexo 6)
		-Uso de herramientas para la gestión de incidencias.	-Plan de pruebas. (Anexo 7)
		-Informe de pruebas, con este informe se documentaran los resultados.	-Informe de pruebas. (Anexo 8)
7.1.7.3.1.3 El implementador debe evaluar el diseño, el código, las pruebas, los resultados de prueba y la documentación del usuario considerando los criterios que se enumeran a continuación. Los resultados de las evaluaciones deben estar documentados.			
a) Cobertura de prueba de los requisitos del elemento de software.	NO	En el plan de pruebas se encontrarán los casos de prueba que son elaborados en base a los requisitos del sistema.	-Plan de pruebas. (Anexo 7)
b) conformidad con los resultados esperados; y	NO	Parte del procedimiento es la gestión de incidencias, las cuales serán reportadas mediante una herramienta a los desarrolladores para que subsanen las incidencias de manera inmediata.	
c) Factibilidad de la integración del sistema y de la prueba, si se realiza.	NO	Una vez finalizadas las pruebas y correcciones se realizará una prueba de regresión que garantizará que el software se encuentra listo para la puesta en operación.	
d) Factibilidad de la operación y el mantenimiento.	NO		
7.1.7.3.1.4 El implementador debe soportar las auditorías de acuerdo con el apartado 7.2.7. Los resultados de las auditorías deben estar documentados. Si tanto el hardware como el software están en proceso de desarrollo o integración, las auditorías se pueden posponer hasta la prueba de calificación del sistema.	NO	En el caso corresponda auditoría, porque no será parte de todos los proyectos, los resultados de la auditoría se documentaran y entregaran a la parte auditada. La parte auditada debe reconocer a la parte que realiza la auditoría todos los problemas hallados en la auditoría y las soluciones planificadas para los problemas relacionados.	-Resultado de la auditoría. (Anexo 9)
7.1.7.3.1.5 Después de la finalización exitosa de las auditorías, si se realizan, El implementador debe actualizar y preparar el producto software entregado para la Integración del Sistema, Prueba de Calificación del Sistema, Instalación del Software o Soporte de Aceptación del Software, según se aplique.	NO	Una vez finalizada la etapa auditoría en caso se realice, se subsanarán los problemas encontrados y el software quedará listo para los procesos de instalación y soporte de aceptación de software	

**ANEXO 3.** Procedimiento establecido para el cumplimiento del Proceso de Pruebas de Calificación del Software.

El siguiente procedimiento indica los pasos que se seguirán en la empresa para el cumplimiento del “Proceso Pruebas de Calificación de Software”, de acuerdo a la norma “ISO/IEC 12207:2008”, para asegurar que el producto software satisface los requisitos definidos.

## HISTORIA DE CAMBIOS

### CONTROL DE CONFIGURACIÓN

<b>Título:</b>	Procedimiento de Pruebas de Calificación del Sistema
<b>Autor:</b>	John Solar
<b>Fecha:</b>	09 de Mayo 2017

### Histórico de Versiones

Versión	Fecha	Estado	Responsable	Descripción del Cambio
1.0	09/05/2017	Creación	John Solar	Elaboración del documento

### Introducción

#### Objetivo

Establecer los lineamientos de Control de Calidad para proporcionar un software de Calidad al cliente interno y externo de TCI.

#### Alcance

Este procedimiento es aplicable al Proceso de Control de Calidad de Software de TCI. Las pruebas se realizan a los aplicativos que conforman los Sistemas de Información, considerando los ámbitos para el cliente interno y externo.

La revisión abarca la verificación de los requisitos especificados en los documentos de alcance entregados en cada requerimiento o proyecto de pruebas.

## Referencias

Norma de calidad ISO 12207:2008

## Responsabilidades:

### Gerencia de TI

Priorizar y entregar los requerimientos y/o proyectos a verificar.

Validar artefacto de alcance

Aprobar Estimación de las pruebas

Revisar el Plan de pruebas

### Líder de Proyecto

Entregar los artefactos (Alcance, Especificación Funcional, Manuales, etc.), necesarios para el análisis de la pruebas a realizar.

Gestionar con el área de desarrollo

### Analista de Calidad

Realizar la gestión de las pruebas para verificar la calidad del software entregado.

Elaborar y distribuir los artefactos para las pruebas.

### Desarrollador

Efectuar los cambios solicitados por el usuario.

Atender las incidencias reportadas.

## Entregables y criterios de aceptación

Artefacto	Responsable del Artefacto	Criterios de Aceptación
Alcance de Requerimiento o Proyecto	Líder de Proyecto	El artefacto esta revisado por el Jefe de Proyecto
Estimación de pruebas	Equipo de Calidad	Validación de Jefe de Proyecto
Cronograma	Equipo de Calidad	Validación de Jefe de Proyecto
Plan de pruebas	Equipo de Calidad	Validación de Jefe de Proyecto
Informe de pruebas	Equipo de Calidad	Validación de Jefe de Proyecto

## **Descripción del procedimiento**

### **Descripción general**

Las pruebas efectuadas para el control de la calidad del software se basan en la verificación del cumplimiento de las especificaciones y/o requisitos indicados por el usuario y que ha sido construido por el área de desarrollo.

La verificación de las funcionalidades se lleva a cabo bajo condiciones definidas explícitamente, con el objetivo de mejorar los niveles de calidad del software y detectar posibles errores, fallos o debilidades en los entregables con la finalidad de garantizar un producto consistente y adecuado a las necesidades planteadas por el Usuario. El sistema de gestión de pruebas es el Testlink.

Todos los errores, fallos u observaciones son registrados en el sistema de gestión de Incidencias (BugTracker) Mantis, para su atención y posterior levantamiento de las mismas.

El personal del Área de Control de Calidad en base a su experiencia, formación, habilidades y educación es capaz de realizar las tareas que se requieren en el Control de Calidad. Todas estas actividades y sus resultados son documentados.

Se definen dos tipos de flujo de pruebas: Pruebas de un Requerimiento (RQ) y Pruebas de un Proyecto (PR), diferenciados por la magnitud del cambio solicitado. Siendo las pruebas RQ pruebas puntuales y con alcance funcional acotado, y pruebas PR las que abarcan una serie de funcionalidades, aplicativos nuevos o módulos completos, siendo impacto de cambio más alto y más complejo.

Para cada requerimiento o proyecto de pruebas se realizan las siguientes actividades:

### **Planificación de las pruebas**

El equipo de control de calidad recibe de parte del Líder de Proyectos la documentación referente al alcance del requerimiento o proyecto a verificar

según la prioridad en la atención que ha indicado el Gerente de TI, en función a la información de dicho análisis el Área de Control de Calidad realiza las siguientes actividades:

### **Análisis de requerimiento**

El alcance de las pruebas se determina en base al análisis de requerimientos. Este se basa en los artefactos sobre el alcance y análisis funcional ya realizado, y que ha sido previamente entregado. Por lo que no todas las opciones serán probadas con la misma importancia y es posible que algunas queden fuera del alcance de las pruebas.

El área de control de calidad presenta como entregables de esta etapa 3 artefactos, dependiendo si es un requerimiento o proyecto el que verificara.

### **Elaborar estimación**

El tiempo de duración de ejecución de las pruebas se estima en base a la información recibida y de la experiencia del Área de Control de Calidad, la cual debe ser entregada al Jefe de Proyectos para su aprobación. Esta estimación puede ser de dos tipos:

Estimación Básica: Estimación realizada a Pruebas RQ, la cual debe ser entregada en un periodo menor a 1 día al Gerente de TI

Estimación Completa: Estimación realizada a Pruebas PR, la cual debe ser entregado al Gerente de TI.

La aprobación de la estimación la realiza el Jefe de Proyectos y en caso de observaciones esta será devuelta para su revisión y modificación al equipo de calidad.

### **Elaborar cronograma**

La elaboración del cronograma de actividades está sujeta a las pruebas de Proyecto (PR), el cual se especifica las responsabilidades de cada tarea y/o actividad a una analista. Para la elaboración del cronograma se hace uso la herramienta Microsoft Project.

La entrega del cronograma se realiza en conjunto con la estimación.

### **Elaborar plan de pruebas**

Con la información de los dos puntos anteriores (Estimación y Cronograma) el equipo de control de Calidad establece el Plan de Pruebas para las pruebas de tipo Proyecto (PR) de acuerdo al Formato Plan de Pruebas del anexo A.

El periodo de entrega del Plan de Pruebas no está sujeto al periodo de la estimación. Sin embargo este debe ser entregado para su revisión antes del diseño de casos de prueba.

El equipo de control de calidad puede iniciar la planeación de las actividades anteriores con el documento y/o información previa referente al requerimiento o proyecto a verificarse, sin embargo, es hasta que recibe el Documento Final de Alcance o Análisis Funcional que puede formalizar dichas actividades.

### **Diseño de las pruebas**

Una vez aprobado los artefactos presentados en la etapa de planeación el equipo de control de calidad procederá a elaborar los casos de pruebas de acuerdo al tipo de pruebas a realizar.

Pruebas de Requerimiento (RQ): El diseño de casos será realizado directamente en la herramienta de gestión de pruebas, TestLink, el cual solo especificara el caso de prueba (cabecera) mas no el detalle del paso a paso.

Pruebas de Proyecto (PR): El diseño de casos será realizado en Excel bajo una estructura ya definida (plantilla), que será convertido a en formato xml para su importación en Testlink.

### **Ejecución de las pruebas**

El área de trabajo de Desarrollo entrega al equipo de Control de Calidad el Desarrollo realizado. La gestión de este entregable es realizado por el Líder de Proyecto el cual velara por que se realice según el cronograma ya presentado.

El equipo de control de Calidad realiza unas pruebas humo (Smoke Test) al desarrollo realizado, esto con el propósito de verificar la estabilidad del producto para ser testado. El Smoke Test es prerequisite en la ejecución de Pruebas de Proyecto (PR) como paso inicial antes de la ejecución de casos de la 1ra

Iteración. En caso se presente observaciones en la ejecución del Smoke Test, se devolverá el desarrollo realizado para su corrección hasta contar con una versión estable para probar.

### **Ejecutar casos de pruebas**

Cada miembro del equipo de Control de Calidad se encargará de realizar las pruebas definidas según sus tareas asignadas.

Las pruebas se realizan de acuerdo a los puntos indicados en el Plan de Pruebas. Las cuáles serán registradas (resultado) en el TestLink.

La ejecución de casos de pruebas se da en dos etapas:

**Ejecución de 1ra Iteración:** Ejecución de casos registrados en TestLink, enfocados en la funcionalidad.

**Ejecución de Regresión:** Ejecución de pruebas de verificación ya levantadas todas las incidencias presentadas en la 1ra iteración, teniendo como requisito un congelamiento de código del área de desarrollo.

### **Gestionar incidencias**

Las observaciones que resultan de las pruebas registradas en el Sistema de Gestión de Incidencias (BugTracker) Mantis y se muestran inicialmente como:

**Estado Asignado:** El programador atenderá las mismas para la verificación y conformidad del equipo de Control de Calidad, dándolas por resuelta si corresponde. En caso de no proceder se definirán como Anuladas y si quedan pendientes de atención se indican como Siguiete Versión.

El equipo de control de calidad puede dar inicio a la ejecución de pruebas sin tener como prerrequisito la finalización 100% del diseño de casos de pruebas. Esta coordinación debe ser realizada con el Gerente de TI y líder de proyecto responsable.

### **Cierre de pruebas**

#### **Elaborar informe de pruebas**

La elaboración del Informe de Pruebas será elaborada por el equipo de pruebas en el cual se registra los resultados de las pruebas realizadas.

El cierre de pruebas no está sujeto a la ejecución del 100% de casos de pruebas, esto en caso de una suspensión de ejecución de pruebas. La suspensión de ejecución debe ser indicada por el Gerente de TI.

**ANEXO 4.** Plantilla de enunciado de alcance del proyecto.**ENUNCIADO DEL “ALCANCE DEL PROYECTO”.**

ENUNCIADO DEL ALCANCE DEL PROYECTO versión 1.0			
PROYECTO:			
JEFE DE PROYECTOS:			
PREPARADO POR:		FECHA	
REVISADO POR:		FECHA	
APROBADO POR:		FECHA	
REVISIÓN	REALIZADA POR	FECHA	
01			
02			
03			
04			

**Objetivos del proyecto.**

[Descripción de los objetivos del proyecto para el negocio.]

**Descripcion del Alcance del Producto**

[Breve descripción del alcance del proyecto]

**Gestión de Requerimientos****Requerimientos funcionales**

[Requerimientos funcionales que describen los procesos de la empresa, cómo se interactúa con el producto, cuáles son las características que tendrá el producto, cuáles son los servicios que el producto deberá proporcionar, cuáles son las casuísticas funcionales del producto, la trazabilidad de los requerimientos se hará en la matriz de trazabilidad]

Distribuir los requerimientos por cada módulo, tratar de colocar requerimientos lo más detallado posible, dependerá del conocimiento y experiencia.

Modulo: [Nombre del módulo]

ID	Requerimiento-Funcional	Descripción
[Identificador-requerimiento]	[Nombre del-requerimiento-funcional]	[Descripción-detallada-del-requerimiento]
[Identificador-requerimiento]	[Nombre del-requerimiento-funcional]	[Descripción-detallada-del-requerimiento]

### Reglas del Negocio por Requerimiento Funcional

[Identificador del requerimiento]	[Nombre del requerimiento]
[Descripción general del requerimiento]	
Id	Reglas del Negocio
[Identificador de la regla de negocio]	[Descripción de la regla de negocio]
[Identificador de la regla de negocio]	[Descripción de la regla de negocio]
[Identificador de la regla de negocio]	[Descripción de la regla de negocio]

### Requerimientos No Funcionales

[Requerimientos particulares o emergentes tales como fiabilidad, tiempo de respuesta, almacenamiento, nivel del servicio, rendimiento, seguridad, cumplimiento, soporte, legislativos, éticos, retención, etc.]

ID	Requerimiento No Funcional	Observaciones
[Identificador requerimiento]	[Nombre del requerimiento no funcional]	[Descripción detallada del requerimiento]
[Identificador requerimiento]	[Nombre del requerimiento no funcional]	[Descripción detallada del requerimiento]

### Requerimientos de Soporte y Entrenamiento

[Describir requerimientos relativos a capacitaciones, manuales, soporte]

### Modelo de Procesos

[Diagrama de modelo de procesos]

Asunciones relativos a los requerimientos

### Limites del Proyecto

[Que trabajo está incluido y excluido del proyecto]

Entregables del Proyecto

[Listado de Entregables del Proyecto]

## ANEXO 5. Plantilla de estimación de pruebas.

Historial de Cambios		Versión	Acción	Fecha acción	Resumen cambios	Responsable de la acción	Aprobado por	Distribuido a
		1.0	--- Seleccione - --			--- Seleccione - --	--- Seleccione -- -	--- Seleccione ---
Datos de Proyecto		Cliente	--- Seleccione - --					
		Línea	--- Seleccione - --					
		Aplicativo	--- Seleccione - --					
		Código en el Cliente						
		Nombre en el Cliente						
		Código SGP (Testlink)						
		Nombre SGP (Testlink)						
Número de horas planeadas por día		9.00		Factor de Ajuste	50%		Cantidad Casos de Pruebas	0
Actividad					Total Esfuerzo Estimado (horas)	Total Esfuerzo Más Probable (horas)	Recursos	Total Días
Planeación					1	1	1.00	0.11
Análisis de Requerimiento					1.00	1.00	1.00	0.11
Inducción					0.00	0.00	1.00	0.00
Estimación de Tiempos					0.00	0.00	1.00	0.00
Diseño					1	1	1.00	0.17
Funcionales					0.00	0.00	1.00	0.00
Automatizadas					0.00	0.00	1.00	0.00
No Impacto					0.00	0.00	1.00	0.00
Aprobación de Casos					0.50	0.50	1.00	0.08
Modificación de Casos					0.50	0.50	1.00	0.08
Ejecución					0.00	0	1.00	0.00
Creación de Set de datos					0.00	0.00	1.00	0.00
1ra Iteración					0.00	0.00	1.00	0.00
Regresión					0.00	0.00	1.00	0.00
Cierre					0.5	0.5	1.00	0.08
Informe de pruebas					0.50	0.50	1.00	0.08
Total Tiempo Estimado de Pruebas					2.50	2.50	1.00	0.36

## ANEXO 6. Plantilla del Cronograma.

Nombre de tarea	Duración	Trabajo	Comienzo	Fin
<b>Proyecto ISO/IEC 12207:2008</b>	<b>7 días</b>	<b>0 horas</b>	<b>jue 09/06/16</b>	<b>mar 21/06/16</b>
<b>Inicio</b>	<b>1,5 días</b>	<b>0 horas</b>	<b>jue 09/06/16</b>	<b>vie 10/06/16</b>
Reunión para definición inicial de requerimiento	1,5 horas	0 horas	jue 09/06/16	jue 09/06/16
Análisis de Requerimiento	2 horas	0 horas	vie 10/06/16	vie 10/06/16
Revisión para definición de estrategia	4 horas	0 horas	vie 10/06/16	vie 10/06/16
<b>Hito 1: Fin de Inicio</b>	<b>0 días</b>	<b>0 horas</b>	<b>vie 10/06/16</b>	<b>vie 10/06/16</b>
<b>Planificación</b>	<b>0,72 días</b>	<b>0 horas</b>	<b>vie 10/06/16</b>	<b>lun 13/06/16</b>
Elaboración de Estimación / Cronograma	3 horas	0 horas	vie 10/06/16	lun 13/06/16
Revisión de Estimación / Cronograma	1 hora	0 horas	lun 13/06/16	lun 13/06/16
Elaboración de Plan de Pruebas	1 hora	0 horas	lun 13/06/16	lun 13/06/16
Revisión de Plan de Pruebas	1 hora	0 horas	lun 13/06/16	lun 13/06/16
<b>Hito 2: Fin de planificación</b>	<b>0 días</b>	<b>0 horas</b>	<b>lun 13/06/16</b>	<b>lun 13/06/16</b>
<b>Diseño</b>	<b>2,67 días</b>	<b>0 horas</b>	<b>lun 13/06/16</b>	<b>mié 15/06/16</b>
Escenarios y Casos de Prueba	4,5 horas	0 horas	lun 13/06/16	mar 14/06/16
Diseño de Pruebas Automatizadas	1,5 días	0 horas	mar 14/06/16	mié 15/06/16
<b>Hito 3: Fin de Diseño</b>	<b>0 días</b>	<b>0 horas</b>	<b>lun 13/06/16</b>	<b>lun 13/06/16</b>
<b>Ejecución</b>	<b>2,78 días</b>	<b>0 horas</b>	<b>jue 16/06/16</b>	<b>mar 21/06/16</b>
Generación de Data	2,5 horas	0 horas	jue 16/06/16	jue 16/06/16
Ejecución de Casos Funcionales	10 horas	0 horas	jue 16/06/16	vie 17/06/16
Ejecución de Casos Automatizados	2 horas	0 horas	vie 17/06/16	vie 17/06/16
Ejecución de Casos No Impacto	1 día	0 horas	vie 17/06/16	lun 20/06/16
Regresión	1,5 horas	0 horas	lun 20/06/16	mar 21/06/16
<b>Hito 4: Fin de Ejecución</b>	<b>0 días</b>	<b>0 horas</b>	<b>vie 17/06/16</b>	<b>vie 17/06/16</b>
<b>Cierre</b>	<b>0,22 días</b>	<b>0 horas</b>	<b>vie 17/06/16</b>	<b>vie 17/06/16</b>
Elaboración de documento de cierre	1 hora	0 horas	vie 17/06/16	vie 17/06/16
Revisión de Documento de Cierre	1 hora	0 horas	vie 17/06/16	vie 17/06/16
<b>Hito 5: Fin de cierre</b>	<b>0 días</b>	<b>0 horas</b>	<b>vie 17/06/16</b>	<b>vie 17/06/16</b>

**ANEXO 7.** Plantilla del plan de pruebas.**Historia de cambios**

## Control de configuración

<b>Título:</b>	Plan De Pruebas De Software Para Proyecto De Testing
<b>Autor:</b>	Juan Soler
<b>Fecha:</b>	99 de XXXX 9999

## Histórico de versiones de plan de pruebas

Versión	Fecha	Estado	Responsable	Descripción del Cambio

**Sección 1: introducción****Objetivo:**

[Colocar el objetivo para este proyecto].

**Aplicativos / componentes involucrados**

IdAC	Aplicativo – Componente / Interfaz	Plataforma	Nuevo / Modificado	Responsable
1				

**Hitos de pruebas (tiempo más probable)**

Id	Tarea	Lista de Id AC [Para saber Aplicativos – Componentes / Interfaces a probar]	Fecha de Inicio (DD/MM/AAAA)	(/DD/ Fecha de Inicio MM/AAAA)	Recurso
1	Inicio	1	24/11/2015	05/01/2016	2
2	Planificación	1	05/01/2016	07/01/2016	1
3	Diseño	1	08/01/2016	12/01/2016	1
4	Ejecución	1	15/01/2016	10/02/2016	1
4.1	Set de Datos	1	15/02/2016	20/02/2016	1
4.2	Ejecución Smoke Test	1	21/02/2016	22/02/2016	1
4.3	Ejecución 1ra Iteración	1	22/02/2016	10/02/2016	1
5	Cierre	1	11/01/2015	11/01/2015	1
Fecha Pase a Producción:					

### Dependencias de Otros Proyectos

[En caso existan dependencias con otros proyectos colocar el detalle en esta sección]

### Restricciones y supuestos

“Las actividades necesarias para alcanzar los objetivos del proyecto (costo, calendario, personas, objetivos de calidad) se planean y llevan a cabo bajo los siguientes supuestos:”

#### Tabla de supuestos

#	Supuestos
1	
2	
3	
4	
5	

### Criterios de Entrada y Salida

Criterios de Entradas:

[Colocar los criterios de entradas]

#### Criterios de Salida:

[Colocar los criterios de salidas]

## SECCIÓN 2: PRUEBAS DEL SOFTWARE

### Alcances

#### Alcance Específico

Verificación Funcional:

La verificación funcional se realizará para escenarios con:

[Especificar escenarios]

#### Fuera del alcance

No se realizarán pruebas de versiones anteriores.

No se considerarán casos de pruebas de performance.

[Especificar escenarios fuera de alcance]

### Estrategia de pruebas

Tipos de prueba a simular

[Especificar tipos de prueba a simular]

### Procedimientos para las pruebas.

[Especificar los procedimientos que se realizaran en las pruebas]

ID	Ítems Lista Requerim.	Funcionalidad a probar	Escenario	Uso de la funcionalidad	Impacto si falla	Total casos prueba
1		Smoke Test	Verificar la estabilidad del ambiente y la factibilidad que el sistema funciona correctamente con todas sus funcionalidades.	Alto	Alto	6
		Funcionales				...
						...
						...
			Total casos			50

### Riesgos del “Proceso de Pruebas”.

#	Evento (riesgo identificado)	Mitigación (acciones para minimizar prob. del evento)	Contingencia (acciones si sucede el evento)
1			
2			
3			
4			
5			
6			
7			

### Casos de prueba y escenarios

Los escenarios y casos de prueba que se realizaran para este proyecto se encuentran en el siguiente documento:

(Adjuntar Escenarios)

Nota: El detalle de los casos de prueba se visualizará en la carpeta de Preparación del Repositorio donde se encontrarán ubicados los casos de prueba realizados.

### Ambiente de pruebas

### Requerimientos de Hardware y Software

[Especificar requerimientos de hardware y software]

### Requerimientos de accesos a sistemas

[Especificar accesos a sistemas]

**Requerimientos de datos de prueba**

[Especificar requerimientos de datos de prueba]

**Requerimientos para contingencias de la prueba**

[Especificar requerimientos para contingencia]

**Aplicaciones**

Las aplicaciones a instalar son las siguientes:

**Aplicaciones del proceso de pruebas**

Id	Aplicación	Versión	Ambiente Certificación	Ambiente UAT/	Ambiente Producción
				Preproducción	
1					
2					
3					
4					
5					

**Datos**

La data requerida que será suministrada es:

[Especificar tipo de data requerida]

La data que se realizará por el equipo de pruebas será:

[Especificar data que generará el equipo de pruebas]

**Dispositivos externos**

[Especificar dispositivos externos necesarios para las pruebas]

**Pruebas especiales**

[Especificar si es necesario realizar pruebas especiales]

**Pruebas Integrales**

[Especificar la necesidad de realizar pruebas integrales]

**Pruebas de Stress/Concurrencia**

[Especificar la necesidad de pruebas de stress o concurrencia]

**Pruebas de Carga/Volumen**

[Especificar la necesidad de pruebas de carga o volumen]

**Pruebas de Ancho de Banda**

[Especificar la necesidad de pruebas de ancho de banda]

**Pruebas de Seguridad de Información**

[Especificar la necesidad de pruebas de seguridad de información]

**Pruebas de Instaladores**

[Especificar la necesidad de pruebas de instaladores].

**Otras Pruebas**

[Especificar otras pruebas]

**Documentación de respaldo****Evidencias**

Se creará una carpeta dentro del repositorio de certificación, con el nombre del proyecto, en donde se guardará todas las evidencias de las pruebas.

**Conformidad de pruebas**

Los informes de resultados de pruebas deberán ser aprobados por el Usuario.

**Reporte de estado final de pruebas**

“El reporte final de las pruebas deberá ser aprobado por el Usuario.

**Plan de Conformidad final de pruebas en el ambiente de producción**

No se han identificado pruebas finales en ambiente de producción

**Tiempos y recursos requeridos (resumen de cronograma)**

Los tiempos requeridos para este proyecto está calculado con 1 recurso para las etapas de planificación, diseño, ejecución y entrega.

Esta sección resume información del cronograma de la fase de Certificación:”

## Cuadro resumen de casos y horas más probables por recurso de certificación

Cuadro resumen de casos y horas más probables por recurso de certificación			Estimación preliminar	
Id	Nombre del Recurso – Rol – Dedicación	Función a probar	# Casos Planificados	# Horas Estimadas
		(Área – Aplicación)		
1	<> - Testing Project Leader- 100%	Etapa de Inicio		
2	<> - Testing Project Analyst - 100%	Etapa de Planeación		
3	<> - Testing Project Analyst - 100%	Etapa de Diseño		
4	<> - Testing Project Analyst - 100%	Etapa de Ejecución		
5	<> - Testing Project Analyst - 100%	Etapa de Cierre		
		<b>Total General</b>		

A continuación se detalla las actividades a realizar durante todo el proceso de certificación del proyecto:

Cronograma Total más Probable de Pruebas del Proyecto			
Tarea	Recurso	Duración	
1. Checklist documentación	1		
2. Análisis de Requerimiento	1		
3. Estimación Inicial	1		
4. Planificación de Actividades	1		
5. Creación del Plan de Certificación	1		
6. Preparación de Casos de Prueba	1		
7. Ejecución de Pruebas	2		
8. Reporte Final	1		
Duración total			

## ANEXO 8. Plantilla del informe de pruebas.

Conformidad de pruebas



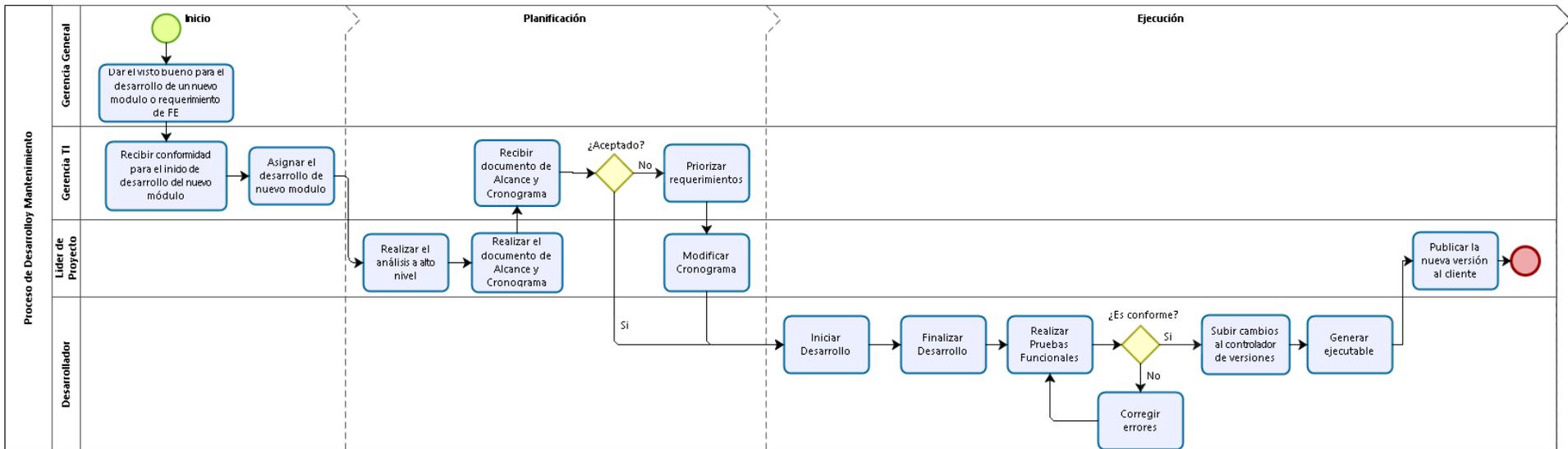
CONFORMIDAD DE PRUEBAS																								
PROYECTO																								
CODIGO N°	001																							
AREA SOLICITANTE:																								
ACCION DE PRUEBA																								
RESULTADOS ESPERADOS																								
OBSERVACIONES DE LA PRUEBA	<p>Etapa de Planeación:</p> <p>Etapa de Diseño</p> <p>Etapa de Ejecución:</p> <p><b><u>Temas que impactaron en los tiempos</u></b></p> <p>Ver Anexo Lista de Errores</p>																							
RECOMENDACIONES																								
RESULTADO FINAL																								
ESTADISTICAS	<table border="1"> <thead> <tr> <th>Estado Casos de Pruebas</th> <th>Cantidad Casos de Pruebas</th> <th>%</th> </tr> </thead> <tbody> <tr> <td>Ok</td> <td></td> <td></td> </tr> <tr> <td>Fallidos</td> <td>0</td> <td></td> </tr> <tr> <td>Pendiente</td> <td>0</td> <td></td> </tr> <tr> <td>Anulado</td> <td>0</td> <td></td> </tr> <tr> <td>Bloqueado</td> <td>0</td> <td></td> </tr> <tr> <td>No Completado</td> <td>0</td> <td></td> </tr> </tbody> </table>	Estado Casos de Pruebas	Cantidad Casos de Pruebas	%	Ok			Fallidos	0		Pendiente	0		Anulado	0		Bloqueado	0		No Completado	0			
	Estado Casos de Pruebas	Cantidad Casos de Pruebas	%																					
	Ok																							
	Fallidos	0																						
	Pendiente	0																						
	Anulado	0																						
	Bloqueado	0																						
No Completado	0																							
TOTAL CASOS DE PRUEBAS			<table border="1"> <thead> <tr> <th>Estado Defectos</th> <th>Cantidad</th> <th>%</th> </tr> </thead> <tbody> <tr> <td>Rechazados</td> <td></td> <td></td> </tr> <tr> <td><b>Solucionado</b></td> <td></td> <td></td> </tr> <tr> <td>Hallazgo</td> <td></td> <td></td> </tr> <tr> <td>No Corresponde</td> <td></td> <td></td> </tr> <tr> <td>Así están en producción</td> <td></td> <td></td> </tr> </tbody> </table>	Estado Defectos	Cantidad	%	Rechazados			<b>Solucionado</b>			Hallazgo			No Corresponde			Así están en producción					
Estado Defectos	Cantidad	%																						
Rechazados																								
<b>Solucionado</b>																								
Hallazgo																								
No Corresponde																								
Así están en producción																								
TOTAL DEFECTOS ENCONTRADOS																								
ANALISTAS DE PRUEBAS																								
JEFE DE PROYECTOS																								
GERENTE DE TI																								
FECHA	29 de Marzo del 2017																							

## ANEXO 9. Resultado de la auditoría.

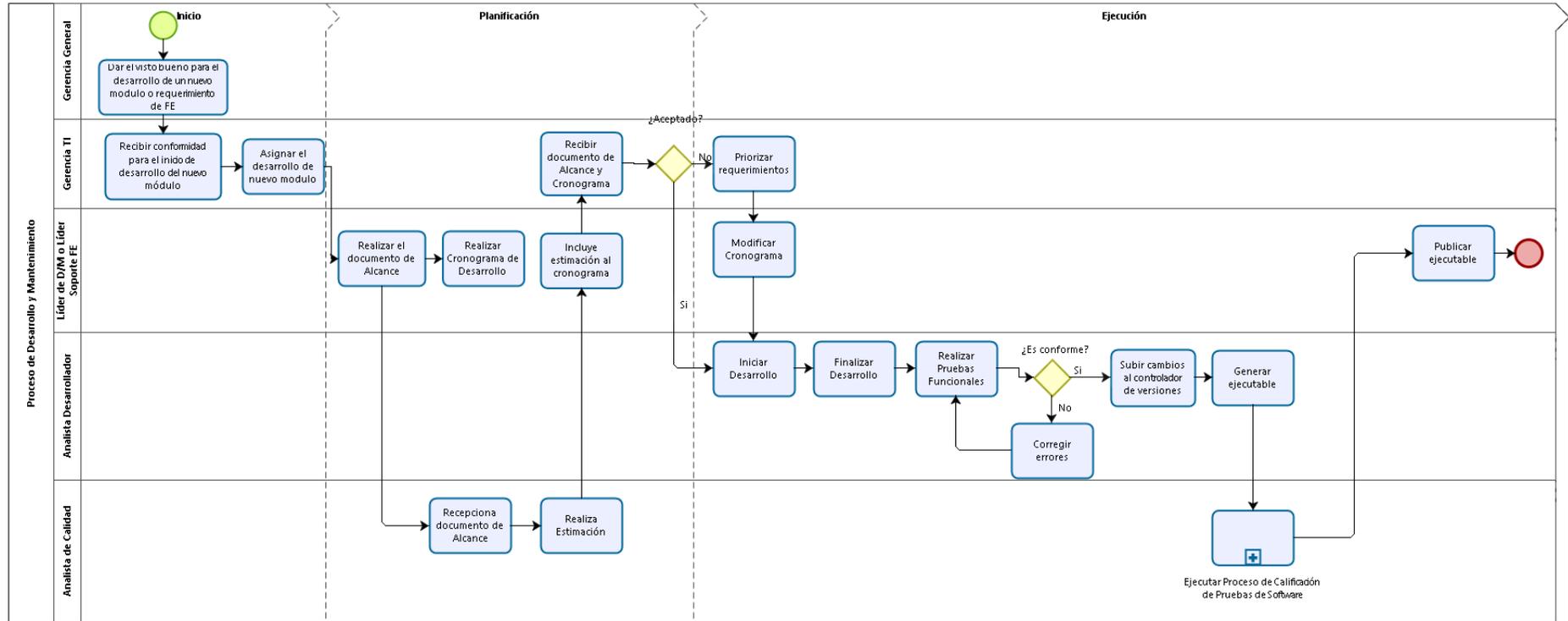
DEPARTAMENTO, ÁREA, O PROCESO AUDITADO		N° HOJA	
Punto 7.2.7 Proceso de Auditoría del Software del ISO 12207:2008		1	
7.2.7.3.1 Implementación del proceso.			
<b>ASPECTOS A VERIFICAR (NOTAS INICIALES DEL AUDITOR)</b>		N° Auditoría:	
7.2.7.3.1.1 Realizar auditorías en puntos clave predeterminados según se especifique en el plan o planes del proyecto.		Fecha inicio:	
7.2.7.3.1.2 El personal de auditoría no debe tener ninguna responsabilidad directa en los productos software ni las actividades que audita.			
7.2.7.3.1.3 Las partes deben estar de acuerdo sobre los recursos requeridos para realizar las auditorías. Estos recursos incluyen personal de soporte, lugar, instalaciones, hardware, software y herramientas.		Hora inicio:	
7.2.7.3.1.4 Las partes deberían acordar sobre los siguientes elementos en cada auditoría: agenda, productos software (y resultados de una actividad) que se van a revisar, alcance y procedimientos de auditoría, y criterios de entrada y salida para la auditoría.		Fecha final:	
7.2.7.3.1.5 Los problemas detectados durante las auditorías se deben registrar e ingresar en el Proceso de Resolución de Problemas del Software (apartado 7.2.8), según se requiera.		Hora final:	
7.2.7.3.1.6 Después de terminar una auditoría, los resultados de ésta se deben documentar y entregar a la parte auditada. La parte auditada debe reconocer a la parte que realiza la auditoría todos los problemas hallados en la auditoría y las soluciones planificadas para los problemas relacionados.		Auditor/es:	
7.2.7.3.1.7 Las partes deben llegar a un acuerdo sobre el resultado de la auditoría y las responsabilidades del elemento de acción y los criterios de cierre.			
OBSERVACIONES / NOTES / COMENTARIOS DEL AUDITOR			
NO CONFORMIDADES (INDICAR LAS EVIDENCIAS DEL INCUMPLIMIENTO)			
ITEM (número)	CALIFICACIÓN (1)	PUNTO NORMA	DESCRIPCIÓN NO CONFORMIDAD Y EVIDENCIAS
--	--	---	---
(1) Calificar N.C. según:		CALIFICACIÓN TOTAL N.C.	
1. Desviación menor: afecta poco al resultado de los procesos.		(Auditor/es)	
2. Desviación moderada. En ciertas condiciones puede afectar a los procesos			
3. Desviación importante. Puede provocar defectos o errores que afecten a la satisfacción del cliente.			

DEPARTAMENTO, ÁREA, O PROCESO AUDITADO		Nº HOJA	
Punto 7.2.7 Proceso de Auditoria del Software del ISO 12207:2008 7.2.7.3.2 Auditoria del software		2	
ASPECTOS A VERIFICAR (NOTES INICIALES DEL AUDITOR)		Nº Auditoría:	
7.2.7.3.2.1 Las auditorías del software se deben realizar para asegurar que: a) productos software reflejan la documentación del diseño, b) De acuerdo a documentación la revisión de aceptación y los requisitos de prueba prescritos son adecuados para la aceptación de los productos software, c) datos de prueba cumplen con la especificación, d) productos software se prueban exitosamente y satisfacen sus especificaciones, e) informes de prueba son correctos y se han resuelto las discrepancias entre los resultados reales y los esperados, f) documentación del usuario cumple con los estándares especificados, g) las actividades se han llevado a cabo de acuerdo con los requisitos, planes y contrato aplicables, h) los costos y cronogramas cumplen con los planes establecidos.		Fecha inicio:	
		Hora inicio:	
		Fecha final:	
		Hora final:	
		Auditor/es:	
OBSERVACIONES / NOTES / COMENTARIOS DEL AUDITOR			
NO CONFORMIDADES (INDICAR LAS EVIDENCIAS DEL INCUMPLIMIENTO)			
ITEM (número)	CALIFICACION <sup>(1)</sup>	PUNTO NORMA	DESCRIPCIÓN NO CONFORMIDAD Y EVIDENCIAS
(1) Calificar N.C. según:		CALIFICACION TOTAL N.C.	
1. Desviación menor: afecta poco al resultado de los procesos.			
2 Desviación moderada. En ciertas condiciones puede afectar a los procesos			
3 Desviación importante. Puede provocar defectos o errores que afecten a la satisfacción del cliente.			
		(Auditor/es)	

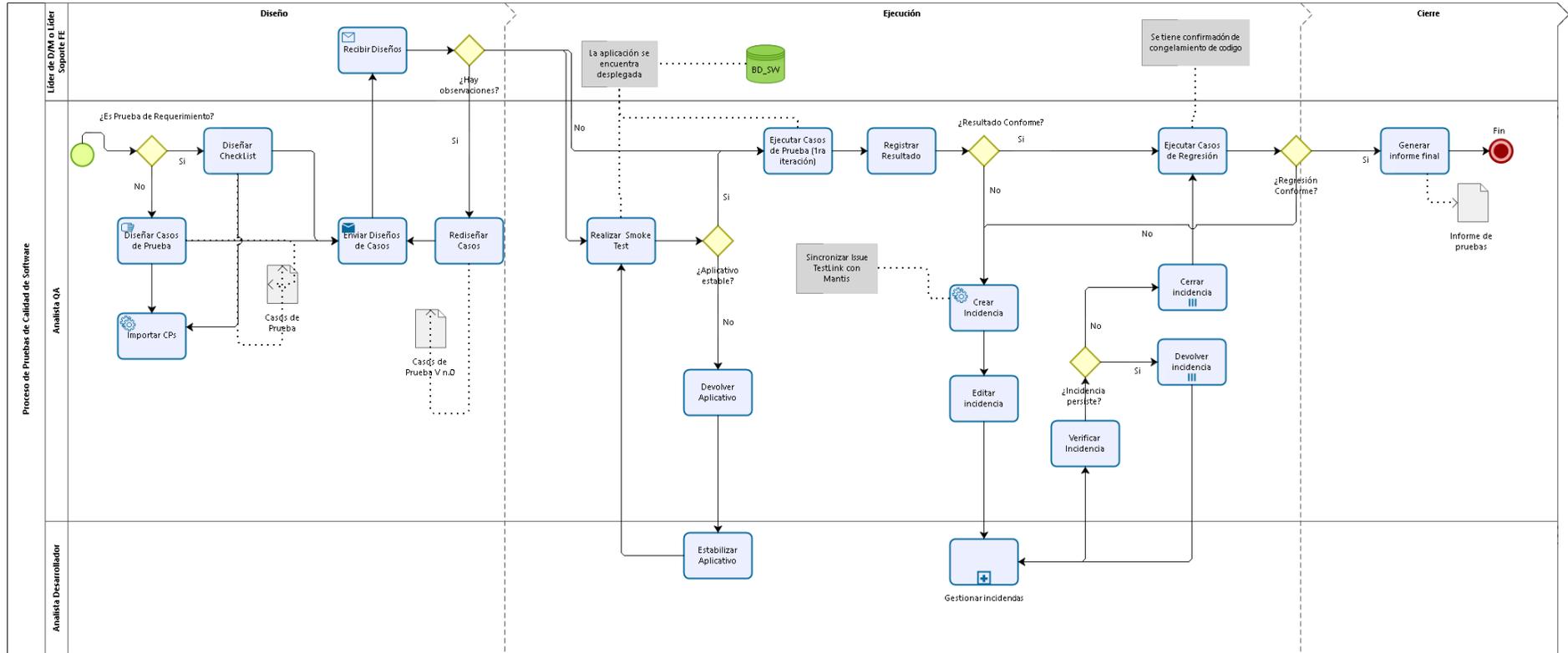
**ANEXO 10.** Diagrama AS-IS del Proceso de Pruebas de Calificación del Software.

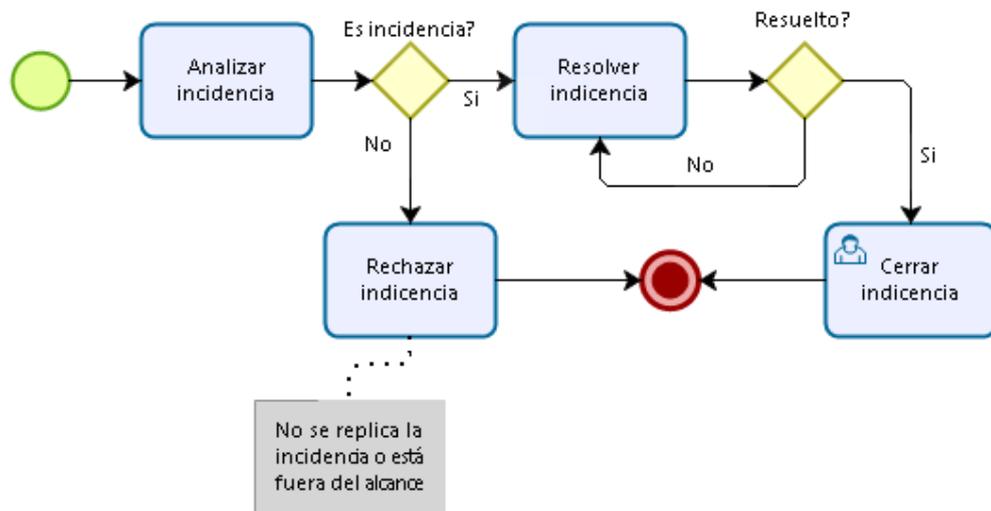


ANEXO 11. Diagrama TO-BE del Proceso de Pruebas de Calificación del Software.



ANEXO 12. SubProceso del Proceso Calificación de Pruebas de Software.



**ANEXO 13.** SubProceso Gestionar Incidencias.

#### ANEXO 14. Proceso de Resolución de Problemas de Software.

El siguiente cuadro describe el propósito del Proceso Resolución de Problemas de Software, las actividades que se deben realizar en este proceso según la norma ISO/IEC 12207, indica por actividad si antes de la implementación se cumplía con este proceso y finalmente el plan de acción especificando los documentos que se elaboraron para el cumplimiento de este proceso.

Proceso de Resolución de Problemas del Software			
<b>Propósito</b>	El propósito del Proceso de Resolución de Problemas del Software es asegurar que todos los problemas descubiertos se identifiquen, analicen, administren y controlen hasta su solución.		
El proyecto debe implementar las siguientes actividades de acuerdo con las políticas y procedimientos de la organización aplicables con respecto al proceso de solución de problemas del software.			
Actividades y tareas Según ISO 12207	Cumplió la empresa	Plan de Acción	
7.2.8.3.1 Implementación del proceso. Esta actividad consta de la siguiente tarea:		Procedimiento	Documento
7.2.8.3.1.1 Se debe establecer un proceso de resolución de un problema para manejar todos los problemas (incluyendo las no conformidades) detectados en los productos y las actividades del software. El proceso debe cumplir con los siguientes requisitos:	NO	Se elaborará un procedimiento para la resolución de problemas que incluirá las actividades y tareas del proceso.	-Procedimiento del Proceso de Resolución del problema (Anexo 15)
a) El proceso debe ser de bucle cerrado, asegurando que: todos los problemas detectados se reporten oportunamente e ingresen en el Proceso de Resolución del Problema; se inicia la acción para ellos; se notifica a las partes pertinentes sobre la existencia de los problemas, según corresponda; se identifican y analizan las causas y, cuando sea posible, se eliminan; se logra la solución y se retroalimenta; se reporta y se realiza seguimiento al estado; y los registros de los problemas son mantenidos según lo estipulado en el contrato.	NO		-Registro de Problemas categorizados por aplicación, prioridad, fecha y estado. (Anexo 16)
b) El proceso debería tener un esquema para clasificar y priorizar los problemas. Se recomienda que cada problema se clasifique por categoría y prioridad para facilitar el análisis de tendencia y solución del problema.	NO		
c) Se debe llevar a cabo el análisis para detectar las tendencias de los problemas reportados.	NO		
d) Se debe evaluar la solución de problemas y las disposiciones para evaluar que los problemas han sido solucionados, que las tendencias adversas han retrocedido y que se han implementado correctamente los cambios en los productos y las actividades del software correspondientes; y para determinar si se han introducido problemas adicionales.	NO		
7.2.8.3.2 Solución de problema. Esta actividad consta de la siguiente tarea:			-Informe del Problema. (Anexo 17)
7.2.8.3.2.1 Cuando se hayan detectado problemas (incluyendo no conformidades) en un producto software o una actividad, se debe elaborar un informe del problema para describir cada problema detectado. El informe del problema se debe utilizar como parte del proceso de bucle cerrado que se describió anteriormente: desde la detección del problema a través de la investigación, el análisis y la solución del problema y su causa, hasta la detección de tendencias de todos los problemas.	NO		

**ANEXO 15.** Procedimiento del Proceso de Resolución de Problemas de Software.

El siguiente procedimiento indica los pasos que se seguirán en la empresa para asegurar el registro de los problemas descubiertos. Asimismo, se identifiquen, analicen, administren y controlen hasta su solución.

### HISTORIA DE CAMBIOS

#### Control de configuración

#### Control de configuración de “resolución de problemas de software”.

**Título:** “Procedimiento del Proceso de Resolución de Problemas de Software”.

**Autor:** John Solar

**Fecha:** 12 de Mayo 2017

#### Histórico de versiones

Versión	Fecha	Estado	Responsable	Descripción del Cambio
1.0	09/05/2017	Creación	John Solar	Elaboración del documento

#### Sección 1. Introducción

##### Objetivo

“Establecer los lineamientos de la Gestión de Resolución de Problemas de Software para asegurar que todos los problemas en el software se identifiquen, analicen, administren y controlen hasta su solución”.

##### Alcance

“Este procedimiento es aplicable al Proceso de Gestión de Resolución de Problemas de Software, específicamente a los problemas identificados en el software en el área de Soporte FE que se encarga de corregir las incidencias presentadas en el software del servicio Factura Electrónica.

Teniendo en cuenta la diferencia entre incidente y problema. El incidente es cualquier evento que no forma parte de la operación estándar de un servicio y que causa, o puede causar, una interrupción del servicio o una reducción de su calidad.

Mientras que problema es la causa desconocida de uno o más incidentes, o sea, un incidente que no tiene su causa raíz identificada acaba transformándose un problema”.

### **Referencias**

Norma ISO/IEC 12207:2008

### **Responsabilidades**

#### **Gerente de TI**

Aprobar Estimación de la corrección del problema

Revisar el Informe del Problema

#### **Líder de Desarrollo y Mantenimiento**

Asignar problema a Analista Desarrollador.

Revisar Informe del problema elaborado por Analista Desarrollador.

Administración del problema.

#### **Líder de Soporte FE**

Identificar si se trata de un problema

Registrar el problema

Comunicar el problema

#### **Analista Desarrollador**

Analizar el problema, identificar las causas, evaluar posibles soluciones.

Elaborar Informe del problema

Realizar la corrección

### **Entregables y Criterios de Aceptación**

<b>Artefacto</b>	<b>Responsable del Artefacto</b>	<b>Criterios de Aceptación</b>
Registro de Problemas	Líder de Soporte FE	El artefacto esta revisado por el Líder de Desarrollo y Mantenimiento.
Informe del Problema	Analista Desarrollador	Validación del Líder de Desarrollo y Mantenimiento y Gerente de TI

## **Sección 2. Descripción del procedimiento**

### **Descripción general**

La corrección a los problemas de software se realiza con el objetivo de mejorar la calidad del software, atacando la causa raíz de los problemas presentados, encontrando soluciones permanentes, lo cual permitirá brindar a los clientes un software estable.

Todos los problemas identificados son registrados en el Registro de Problemas, para su posterior análisis, estimación, elaboración del informe del problema y corrección.

Para cada Problema se realizan las siguientes actividades:

### **Identificación del Problema**

El Líder de Soporte de FE recibe por correo electrónico del Líder de Soporte de primera línea las incidencias que no pueden ser solucionadas por el equipo de Soporte de primera línea. El Líder de Soporte de FE que lleva el control de las incidencias en la herramienta MyhelpDesk que es una donde soporte de primera línea registra las incidencias, identifica que la incidencia recibida ya no es una incidencia, sino un problema.

“El Líder de Soporte de FE identifica que se trata de un problema cuando se trata de una incidencia que se ha presentado anteriormente y que ya fue solucionada pero persiste”.

El Líder de Soporte de FE registra el problema en el artefacto Registro de Problemas que es un archivo en formato Excel que se encuentra en un directorio compartido con acceso solo por los Líderes de Soporte de FE y Desarrollo y Mantenimiento.

El problema se registra con el estado “Pendiente” y se indica la prioridad, detalle del problema, pasos para reproducir el problema, resultado esperado, resultado real, información del entorno, fecha, prioridad y el nombre del Líder de Soporte de FE o encargando por si el Líder de Soporte de FE se encuentra de vacaciones.

### **Análisis de Problema**

El Líder de Desarrollo y Mantenimiento revisa el problema y según su prioridad que puede ser Alta, Media, Baja y disponibilidad de Analistas Programadores asigna el caso a un Analista Programador.

Una vez que el Analista Programador inicia la revisión del caso el Líder de Desarrollo y Mantenimiento actualiza el estado del problema al estado "En análisis.

El Analista Programador analiza el problema con la información proporcionada por el Líder de Soporte de TI, identifica las causas y evalúa posibles soluciones. Elabora el Informe del Problema y estimación del desarrollo de la solución el cual es presentado en un cronograma haciendo uso de la herramienta Project.

Entrega al Líder de Desarrollo y Mantenimiento el Informe del Problema y Cronograma para su revisión y se reúne con el Analista Desarrollador para repasar detalles de la solución propuesta, en el caso de necesitar ajustes a la propuesta el Analista Desarrollador las realiza.

### **Presentación de la Propuesta de Solución**

El Líder de Desarrollo y Mantenimiento y Analista Desarrollador presentan el Informe del Problema y el Cronograma al Gerente de TI, quien decide si se realizará o no se realizará el desarrollo, en el caso que dé su aprobación se ejecuta el proceso de Desarrollo y Mantenimiento de Software y el problema cambia el estado **En Desarrollo** Una vez finalizado el desarrollo y pruebas el Líder de Desarrollo y Mantenimiento cambia el estado del problema a **Corregido**. **En el caso que el Gerente de TI desapruébe el inicio del desarrollo el problema cambia al estado "Desarrollo Estimado.**



## ANEXO 17. Informe del problema del Proceso de Resolución de Problemas de Software

### Informe del problema



Informe del problema  
99/99/9999

#### PROPORCIONE UN RESUMEN

(Describa en unas pocas palabras los problemas que ha encontrado. Normalmente es una buena idea usar elementos existentes del gestor como ejemplo si es la primera vez que informa de un fallo.)

#### PROPORCIONE DETALLES SOBRE LA INCIDENCIA

(Para proporcionar la mayor información posible, el gestor rellena el campo de descripción con una plantilla que contiene 5 subsecciones:)

##### Pasos para reproducir la incidencia:

(detallar sobre cómo otra persona puede reproducir su incidencia.)

##### Resultado esperado:

(lo que cree que debería ocurrir al realizar los pasos anteriores.)

##### Resultado real:

(lo que realmente sucede al realizar los pasos anteriores)

##### • Información del sistema:

(información acerca del entorno en el que está configurado su sistema.)

##### Comentarios adicionales:

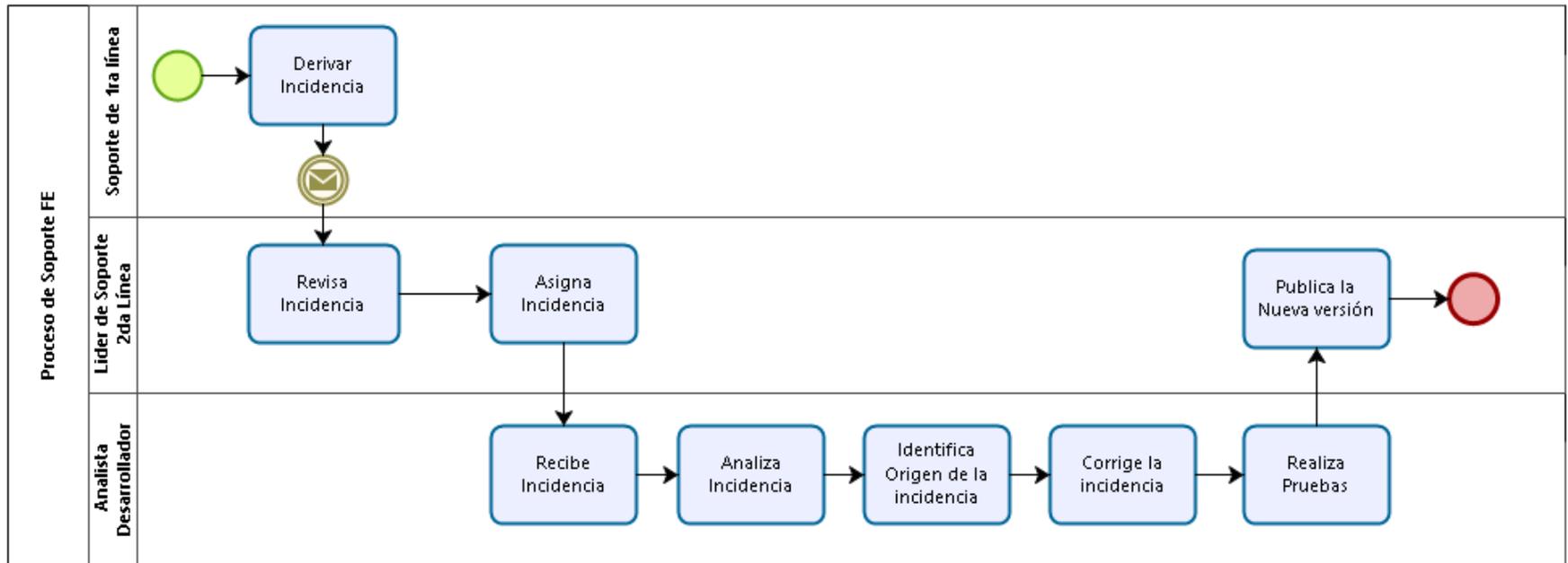
##### Esto es exactamente lo que hice

##### Esto es lo que pasó

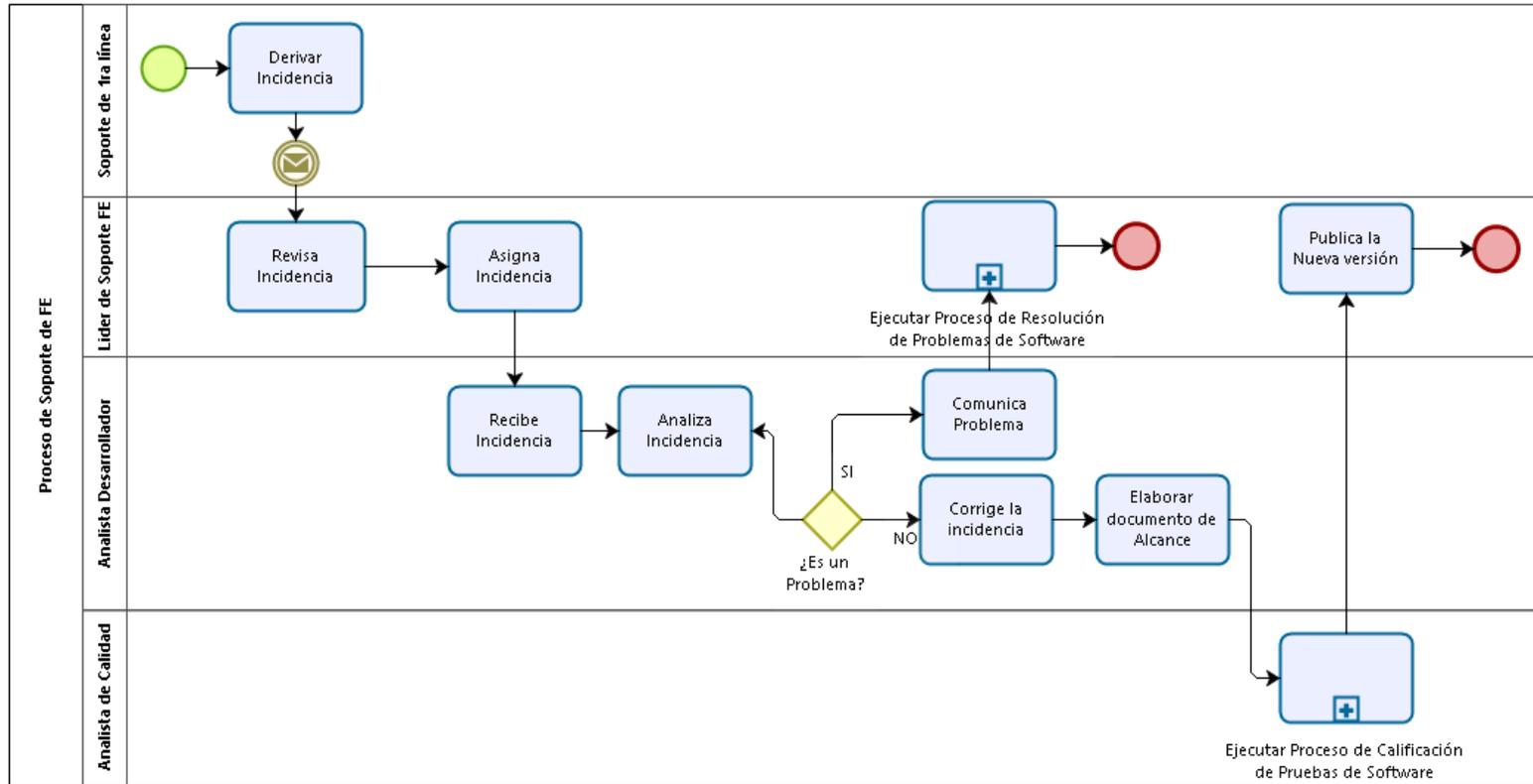
##### Esto es lo que creo que debería haber pasado

##### Otra información, posible solución, parche de código propuesto

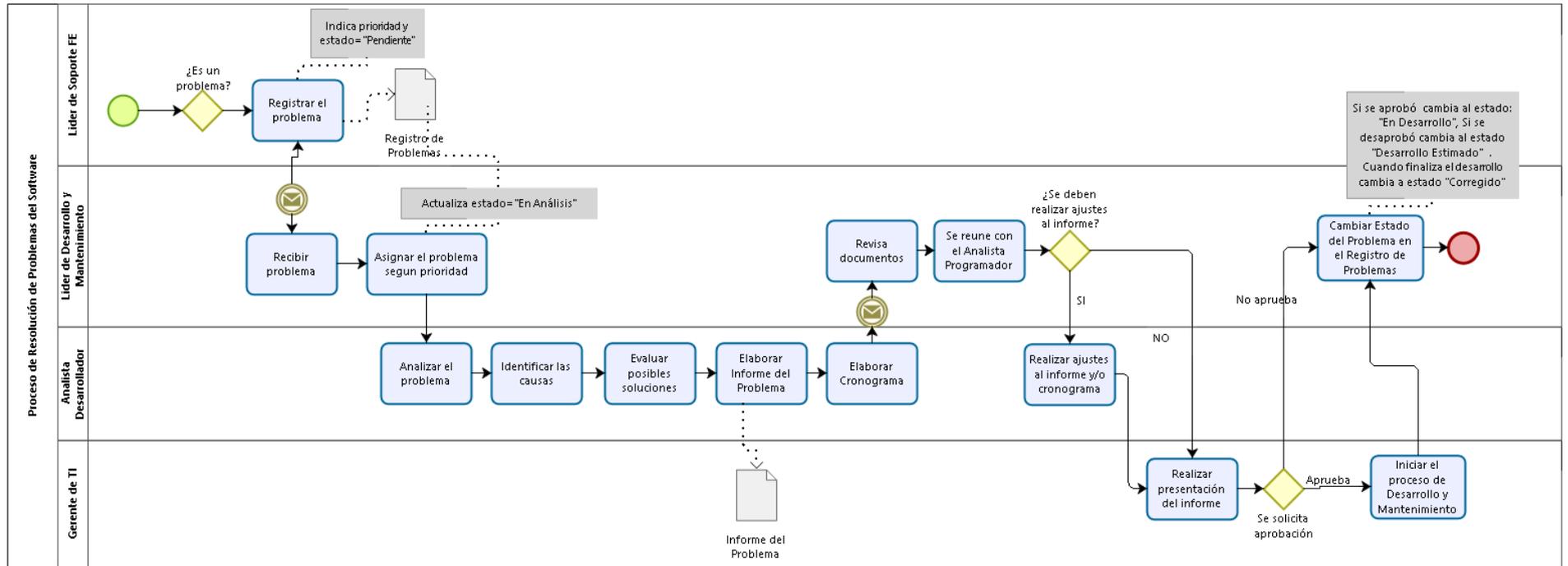
**ANEXO 18.** Diagrama AS-IS del Proceso de Resolución de Problemas de Software.



**ANEXO 19.** Diagrama TO-BE del Proceso de Resolución de Problemas de Software.



ANEXO 20. Diagrama SubProceso del Proceso de Resolución de Problemas de Software.



**ANEXO 21. Valorización del proyecto.**

<b>DATOS DE PROYECTO</b>	
Horas al día	8
Días al mes	22
Tarifa por hora del Líder de Desarrollo/Mantenimiento	S/.31,25
Tarifa por hora del Líder de Soporte FE	S/.28,41
Tarifa por hora del Analista Programador 1	S/.19,89
Tarifa por hora del Analista Programador 2	S/.18,75
Tarifa por hora del Analista Programador 3	S/.17,05

<b>Inversión para el proyecto</b>	<b>Total</b>
Servidor para aplicación web	S/.4.620
Electricidad	S/.5.000
Mantenimiento, vigilancia, agua	S/.1.500
Americatel	S/.1.320
Infointernet – telefónica	S/.3.000
Telefónica- telefonía fija	S/.700
Telefónica –móviles	S/.3.000
<b>TOTAL</b>	<b>S/.19.140</b>

<b>FUNCION EN EL PROYECTO</b>	<b>ABRIL</b>			<b>MAYO</b>			<b>JUNIO</b>			<b>TOTAL</b>
	<b>Particip.</b>	<b>Hrs/H</b>	<b>T. Mes</b>	<b>Particip.</b>	<b>Hrs/H</b>	<b>T. Mes</b>	<b>Particip.</b>	<b>Hrs/H</b>	<b>T. Mes</b>	
Jefe de Proyecto	50%	88	2,500	30%	52,8	1,500	10%	17,6	500	4,500
Jefe de Soporte y Mantenimiento	50%	88	2,750	50%	88	2,750	10%	17,6	550	6,050
Analista Programador 1	50%	88	1,650	100%	176	3,300	10%	18	330	5,280
Analista Programador 3	50%	88	1,500	100%	176	3,001	10%	17,6	300	4,801
										<b>S/ 20,631</b>

**Cálculo**

Total mes = Hrs/H x Tarifa x Hora

**Presupuesto del proyecto**

<b>Resumen General</b>	<b>Total</b>
Equipo de trabajo	S/.20.631
Inversión tecnología y servicios básicos	S/.2.393
Reserva de contingencia (5%)	S/.1.151
<b>Total Línea Base</b>	<b>S/.24.175</b>
Reserva de gestión (5%)	S/.1.209
<b>Presupuesto del proyecto</b>	<b>S/.25.384</b>

**ANEXO 22.** Estimación del ahorro utilizando el método de Fermi.

Para realizar el cálculo del ahorro por incidencia, se consideró el número de horas promedio requeridas para resolver una incidencia y el sueldo por hora del trabajador encargado de la resolución de incidencias. Dicho resultado dio un total de 87.5 soles por incidencia resuelta.

<p>40 horas semanales 160 horas al mes</p> <p>S/. 3500 (sueldo) /160 (horas) = S/. 21.875 (hora) S/. 21.875 * 3 (horas) = S/. 87.5</p>
------------------------------------------------------------------------------------------------------------------------------------------------

En cuanto a las quejas, el departamento administrativo informa que en promedio son necesarias 3 quejas para que un cliente decida no renovar su contrato de servicio, los cuales tienen un monto promedio de S/ 15750. De esta manera, por cada queja, se estaría perdiendo un total de S/. 5250.

Finalmente, la empresa terceriza su selección de personas con una consultora que solicita S/ 4200 en promedio por posición seleccionada. Anualmente, se informa que se realizan 8 requerimientos de este tipo. Así es como para el cálculo del ahorro por la rotación, se considera la multiplicación del precio por selección por el número de requerimientos por la variación en la rotación.

La suma de estos tres conceptos representa el ahorro total por la intervención.

$$(S/ 87.5 * \Delta \text{ Incidencias} * 12 \text{ meses}) + (S/ 5250 * \Delta \text{ Quejas} * 12 \text{ meses}) + (S/ 4200 * 8 * (\Delta \text{ Rotación}))$$