

# **ESCUELA POLITÉCNICA NACIONAL**

## **FACULTAD DE INGENIERÍA DE SISTEMAS**

### **ELABORACIÓN DE UNA GUÍA DE PROCEDIMIENTOS PARA EL PROCESO DE MANTENIMIENTO DE SOFTWARE**

#### **PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

**BALSECA RODRÍGUEZ TANIA MARCELA**  
namavi7\_88@hotmail.com

**CAIZA CHACÓN GABRIELA ALEXANDRA**  
gabyalex6@hotmail.com

**DIRECTOR: ING. MARCOS RAÚL CÓRDOVA BAYAS**  
raul.cordova@epn.edu.ec

**Quito, Agosto 2014**

## DECLARACIÓN

Nosotros, Tania Marcela Balseca Rodríguez y Gabriela Alexandra Caiza Chacón, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

---

**Tania Balseca Rodríguez**

---

**Gabriela Caiza Chacón**

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue desarrollado por Tania Marcela Balseca Rodríguez y Gabriela Alexandra Caiza Chacón bajo mi supervisión.

---

**Ing. Raúl Córdova**  
**DIRECTOR DE PROYECTO**

## AGRADECIMIENTO

A Dios por darme la fuerza, sabiduría y valor para realizar todo lo que he logrado hasta ahora de la mejor manera, sin decaer y así seguir luchando por alcanzar mis metas.

A mis padres Gabriel e Hipatia por su apoyo incondicional en todo momento de mi vida, por la motivación para seguir adelante sin dejarme vencer por cualquier obstáculo. Su compañía en los malos y buenos momentos me ha dado la fuerza necesaria para continuar. Agradecida por los valores que me inculcaron desde niña, todo lo que soy es gracias a mis padres.

A mi hijito Ángel Gabriel, el amor de mi vida, quién es el motor de mi vida y de mis logros de ahora en adelante. Es una bendición en mi vida y estoy tan agradecida por haberme elegido como madre.

A mis hermanas Gabriela y Valeria que sin duda han sido un apoyo muy importante en mi vida que me ha impulsado a no estancarme cuando he atravesado problemas. Con sus locuras, alegría y consejos han hecho de mí una mejor persona.

A mis sobrinitas Nathalia y Victoria, que son un pilar fundamental en mi vida.

Al Ingeniero Raúl Córdova, por guiarnos en este largo y paciente proyecto, brindándonos sus conocimientos una vez más para culminar con éxito.

**Marcela Balseca R**

## DEDICATORIA

A mis padres, sin ellos no hubiera logrado obtener este gran triunfo y ahora culminarlo es la mejor satisfacción en mi vida.

A mi hijito, mi muñequito Ángel Gabriel quién es el que me da fuerza para vivir y ser mejor cada día, seguiré luchando y ahora las enseñanzas y valores que mis padres inculcaron en mí, lo haré con mi hijito, ser mejor por él. Te amo mucho mi bebito.

A la Escuela Politécnica Nacional, mi querida Universidad por ser el reto más importante que se presentó en mi vida y ahora cumplirlo me hace sentir muy orgullosa de mí misma y de todas las experiencias que para mal o para bien me hicieron una mujer más fuerte.

A los Ingenieros(as), por brindar sus conocimientos, su ayuda, sus experiencias.

“Un triunfador es aquel que se levanta y busca las circunstancias que desea y si no las encuentra las fabrica - George Bernard Shaw”

**Marcela Balseca R**

## AGRADECIMIENTO

*A Dios, por darme la vida, por guiar mis pasos por el sendero del bien y sobre todo por permitirme culminar con éxito tan anhelado sueño. Gracias, por brindarme tu apoyo cuando me sentí derrotada por muchas dificultades que se presentaron durante mi vida universitaria, sin embargo, jamás me dejaste afrontar sola mis problemas y hoy gracias a ti he cumplido uno de mis grandes sueños.*

*A mis padres, por sus enseñanzas y apoyo constante, sobre todo por el sacrificio que hicieron por mí, sin ellos no hubiera podido lograr esta meta. Mil gracias por sus consejos, esfuerzo y dedicación que han realizado día a día por mí para poder cumplir este sueño.*

*A mi madre, por el amor, la comprensión y el apoyo que me ha dado durante toda mi vida y más aún durante mi carrera universitaria, gracias por creer en mí y por estar siempre a mi lado cuando lo necesité.*

*A mi hermana, que a pesar de la distancia siempre estuvo pendiente de mí, brindando su cariño y apoyo incondicional. Gracias por estar conmigo en los momentos más importantes de mi vida.*

*A mi sobrina Jocelyne, que ha sido una de mis mayores inspiraciones para lograr esta meta, que siempre tiene una sonrisa o una travesura para alegrar mi vida*

*Al Ing. Raúl Córdova por su ayuda y guía en el desarrollo de este proyecto, gracias por su paciencia y enseñanza.*

*A mis profesores porque han sido parte fundamental para alcanzar este reto académico y a quienes les debo gran parte de mis conocimientos.*

*A mis amigos, por mostrarme lo maravillosa que es la vida cuando se cuenta con personas tan especiales que celebran mis triunfos y me tienden una mano para levantarme en mis tropiezos.*

*Gracias a todas las personas que hicieron esto posible, a la empresa LogiCiel y a todas las personas que nos abrieron las puertas para poder realizar este trabajo.*

**Gabriela Caiza**

## DEDICATORIA

*Dedico este trabajo a Dios, por ser mi guía en cada paso que doy, por ser quién ha estado a mi lado cuidándome y dándome fortaleza necesaria para continuar luchando día tras día.*

*En especial dedico este trabajo a mi madre, por su abnegación, sabiduría y compañía en cada momento de mi vida, su luz de amor me ha guiado por el camino del bien. Ella es mi mayor inspiración para seguir adelante y lograr mis metas. Por brindarme siempre su apoyo incondicional en este extenso y arduo camino.*

*Por acompañarme en este sueño que empezó unos años atrás, por estar conmigo cuando he llorado y he reído; pero sobre todo por disfrutar conmigo el dulce sabor que deja la satisfacción de una meta cumplida.*

*Gabriela Caiza*

## CONTENIDO

<b>1 CAPÍTULO 1 ESTUDIO SOBRE LOS FUNDAMENTOS DEL MANTENIMIENTO DE SOFTWARE .....</b>	<b>1</b>
1.1 DESCRIPCIÓN DE LOS ASPECTOS FUNDAMENTALES DEL MANTENIMIENTO DE SOFTWARE.....	1
1.1.1 DEFINICIONES Y TERMINOLOGÍA.....	1
1.1.2 NECESIDADES DE MANTENIMIENTO .....	2
1.1.3 COSTOS MAYORITARIOS DE MANTENIMIENTO.....	2
1.1.4 EVOLUCIÓN DEL SOFTWARE .....	3
1.1.4.1 Proceso de evolución .....	4
1.1.5 FACTORES CLAVES EN EL MANTENIMIENTO DE SOFTWARE .....	6
1.1.5.1 Factores técnicos.....	6
1.1.5.2 Factores administrativos.....	8
1.1.5.3 Estimación del esfuerzo. ....	9
1.1.6 SEGUIMIENTO AL PROCESO DE MANTENIMIENTO .....	10
1.1.6.1 Gestión de la Configuración .....	11
1.1.6.2 Aseguramiento de la Calidad .....	12
1.1.7 TÉCNICAS PARA EL MANTENIMIENTO .....	12
1.1.7.1 Comprensión del programa.....	12
1.1.7.2 Reingeniería.....	13
1.1.7.3 Ingeniería inversa .....	14
1.1.8 ACTORES Y ROLES EN EL MANTENIMIENTO DEL SOFTWARE.....	15
1.2 DESCRIPCIÓN DEL PROCESO DE MANTENIMIENTO DE SOFTWARE .....	15
1.2.1 ESTÁNDARES .....	16
1.2.1.1 Estándar IEEE 1219 .....	16
1.2.1.2 Estándar ISO/IEC 12207 .....	22
1.2.2 METODOLOGÍAS .....	25
1.2.2.1 Metodología Mantema .....	25
1.2.2.2 Metodología MÉTRICA V3 .....	29
1.3 DESCRIPCIÓN DE LOS TIPOS DE MANTENIMIENTO DE SOFTWARE .....	30
1.3.1 MANTENIMIENTO CORRECTIVO .....	31
1.3.2 MANTENIMIENTO DE EMERGENCIA.....	31
1.3.3 MANTENIMIENTO ADAPTATIVO .....	32
1.3.4 MANTENIMIENTO PREVENTIVO .....	32
1.3.5 MANTENIMIENTO PERFECTIVO .....	32
1.4 ELABORACIÓN DE GUÍAS TÉCNICAS .....	34

1.4.1	CARACTERÍSTICAS Y FUNCIONES DE UNA GUÍA.....	34
1.4.2	ESTRUCTURA DE UNA GUÍA TÉCNICA .....	34
<b>2</b>	<b>CAPÍTULO 2 ELABORACIÓN DE LA GUÍA DE PROCEDIMIENTOS PARA EL PROCESO DE MANTENIMIENTO DE SOFTWARE.....</b>	<b>37</b>
2.1	INTRODUCCIÓN.....	37
2.1.1	VISIÓN GENERAL DE LA GUÍA.....	38
2.1.2	ESTRUCTURA DE LA GUÍA.....	40
2.1.3	RESUMEN DE LA GUÍA.....	40
2.2	OBJETIVOS Y ALCANCE.....	43
2.2.1	OBJETIVOS .....	43
2.2.2	ALCANCE.....	43
2.3	CONTENIDO DE LA GUÍA .....	44
I.	DESCRIPCIÓN DEL SISTEMA .....	45
II.	PROCEDIMIENTOS.....	45
1.	ACTIVIDADES Y TAREAS INICIALES .....	45
1.1.	Fase 1: Identificación, clasificación y priorización del problema .....	45
2.	ACTIVIDADES Y TAREAS PARA UN MANTENIMIENTO DE EMERGENCIA .....	51
2.1.	Fase 2: Análisis .....	51
2.2.	Fase 3: Diseño .....	53
2.3.	Fase 4: Implementación.....	54
2.4.	Fase 5: Pruebas del sistema .....	54
2.5.	Fase 6: Pruebas de aceptación.....	55
2.6.	Fase 7: Puesta en producción.....	56
3.	ACTIVIDADES Y TAREAS PARA UN MANTENIMIENTO CORRECTIVO, PERFECTIVO, PREVENTIVO Y ADAPTATIVO .....	60
3.1.	Fase 2: Análisis .....	60
3.2.	Fase 3: Diseño .....	66
3.3.	Fase 4: Implementación.....	68
3.4.	Fase 5: Pruebas del sistema .....	69
3.5.	Fase 6: Pruebas de aceptación.....	70
3.6.	Fase 7: Puesta en producción.....	71
4.	FLUJOS DE APOYO AL PROCESO.....	78
4.1.	Aseguramiento de la Calidad.....	78
4.2.	Gestión de la Configuración .....	78
<b>3</b>	<b>CAPÍTULO 3 APLICACIÓN DE LA GUÍA DE PROCEDIMIENTOS DE MANTENIMIENTO DE SOFTWARE A UN CASO DE ESTUDIO.....</b>	<b>80</b>
3.1	DESCRIPCIÓN DEL CASO DE ESTUDIO .....	80

3.2	APLICACIÓN DE LA GUÍA .....	81
I.	DESCRIPCIÓN DEL SISTEMA .....	82
II.	PROCEDIMIENTOS.....	83
3.3	ANÁLISIS DE RESULTADOS .....	148
<b>4</b>	<b>CAPÍTULO 4 CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>149</b>
4.1	CONCLUSIONES.....	149
4.2	RECOMENDACIONES.....	150
<b>5</b>	<b>GLOSARIO .....</b>	<b>151</b>
<b>6</b>	<b>BIBLIOGRAFÍA.....</b>	<b>156</b>
<b>7</b>	<b>ANEXOS.....</b>	<b>157</b>

## ÍNDICE DE TABLAS

### CAPITULO 1

Tabla 1.1 Leyes de Lehman .....	4
Tabla 1.2 Entradas, controles y salidas de las distintas etapas del proceso de mantenimiento del software (IEEE 1219-1999) .....	21
Tabla 1.3 Cuadro comparativo entre la IEEE 1219 e ISO/IEC 12207 .....	25

### CAPITULO 2

Tabla 2.1 Resumen de las actividades y tareas de la fase 1.....	50
Tabla 2.2 Resumen de las actividades y tareas para un Mantenimiento de Emergencia.....	59
Tabla 2.3 Resumen de las actividades y tareas para un Mantenimiento Correctivo, Perfectivo, Preventivo y Adaptativo. ....	77

## ÍNDICE DE FIGURAS

### CAPITULO 1

Figura 1.1 Proceso de evolución.....	5
Figura 1.2 Proceso de reparación de emergencia .....	5
Figura 1.3 Modelo de proceso de la reingeniería del software .....	13
Figura 1.4 Proceso de ingeniería inversa .....	14
Figura 1.5 Fases del proceso de mantenimiento de software (IEEE 1219-1999) .....	16
Figura 1.6 Proceso de mantenimiento (Métrica V3, 2000) .....	29
Figura 1.7 Tipos de mantenimiento según la Metodología – MANTEMA .....	30
Figura 1.8 Esfuerzo dedicado a cada tipo de mantenimiento.....	33
Figura 1.9 Ubicación de cada elemento del formato de identificación de una guía técnica. ....	35

### CAPITULO 2

Figura 2.1: Esquema general del contenido de la guía.....	39
Figura 2.2: Estructura del contenido de la guía.....	41
Figura 2.3: Fases, actividades y entregables que contiene la guía.....	42
Figura 2.4 Clasificación de la solicitud de modificación por tipo de mantenimiento .....	46

### CAPITULO 3

Figura 3.1 Solicitud de modificación inicial del caso de estudio (PLANTILLA 1).....	83
Figura 3.2 Registro del estudio de la solicitud de modificación inicial (PLANTILLA 2). ....	84
Figura 3.3 Solicitud de modificación clasificada y validada (PLANTILLA 1). ....	85
Figura 3.4 Inventario de la documentación del sistema (PLANTILLA 3). ....	86
Figura 3.5 Registro de asignación de recursos (PLANTILLA 4). ....	87
Figura 3.6 Definición del impacto del cambio (PLANTILLA 5). ....	88
Figura 3.7 Lista de requerimientos funcionales del sistema LogiHistoricos.....	92
Figura 3.8 Resumen de los requerimientos del sistema LogiHistoricos.....	92
Figura 3.9 Matriz de rastreabilidad (PLANTILLA 10). ....	93
Figura 3.10 Estimación de costo y esfuerzo (PLANTILLA 11). ....	94
Figura 3.11 Listado de elementos a modificar (PLANTILLA 12). ....	95
Figura 3.12 Plan de pruebas del sistema LogiHistoricos (PLANTILLA 13). ....	98
Figura 3.13 Plan de implementación inicial (PLANTILLA 14). ....	99
Figura 3.14 Listado de elementos a modificar actualizado y revisado (PLANTILLA 12). ....	100
Figura 3.15 Listado de documentos a modificar (PLANTILLA 15). ....	101
Figura 3.16 Caso de Prueba 01 (PLANTILLA 16-CP01). ....	102
Figura 3.17 Caso de Prueba 02 (PLANTILLA 16-CP02). ....	103
Figura 3.18 Caso de Prueba 03 (PLANTILLA 16-CP03). ....	104
Figura 3.19 Caso de Prueba de Estrés 01 (PLANTILLA 16-CPR01). ....	105
Figura 3.20 Caso de Prueba de Estrés 02 (PLANTILLA 16-CPR02). ....	106
Figura 3.21 Caso de Prueba de Estrés 03 (PLANTILLA 16-CPR03). ....	107
Figura 3.22 Caso de Prueba de Aceptación 01 (PLANTILLA 16-CPA01). ....	108
Figura 3.23 Caso de Prueba de Aceptación 02 (PLANTILLA 16-CPA02). ....	109
Figura 3.24 Caso de Prueba de Aceptación 03 (PLANTILLA 16-CPA03). ....	110
Figura 3.25 Plan de implementación actualizado y revisado (PLANTILLA 14). ....	111

Figura 3.26 Pantalla principal de LogiHistoricos. ....	112
Figura 3.27 Menú principal de LogiHistoricos antes del cambio. ....	112
Figura 3.28 Solución del sistema LogiHistoricos.....	113
Figura 3.29 Gestores que contiene el sistema LogiHistoricos.....	113
Figura 3.30 Controles que contiene el sistema LogiHistoricos.....	114
Figura 3.31 Vistas que contiene el sistema LogiHistoricos.....	114
Figura 3.32 Control de la pantalla Métricas por proyecto .....	115
Figura 3.33 Código generado para obtener las métricas por proyecto .....	116
Figura 3.34 Vista de la pantalla Métricas por Proyecto.....	116
Figura 3.35 Código para crear los elementos de la pantalla métricas por proyecto.....	117
Figura 3.36 Gestor de la pantalla Métricas por Responsable.....	118
Figura 3.37 Código creado para calcular las métricas por responsable .....	119
Figura 3.38 Control de la pantalla Métricas por Responsable.....	119
Figura 3.39 Código generado para obtener las métricas por responsable .....	120
Figura 3.40 Vista de la pantalla Métricas por Responsable.....	121
Figura 3.41 Código para crear los elementos de la pantalla métricas por responsable .....	122
Figura 3.42 Control de la pantalla Métricas Estándar .....	122
Figura 3.43 Código generado para obtener las métricas estándar .....	123
Figura 3.44 Vista de la pantalla Métricas Estándar .....	124
Figura 3.45 Código para crear los elementos de la pantalla métricas estándar.....	125
Figura 3.46 Pantalla principal de LogiHistoricos .....	125
Figura 3.47 Menú principal de LogiHistoricos después del cambio .....	126
Figura 3.48 Menú de la opción Reportes.....	126
Figura 3.49 Pantalla de Métricas por Proyecto.....	127
Figura 3.50 Pantalla de Métricas por Responsable .....	127
Figura 3.51 Pantalla de Métricas Estándar.....	128
Figura 3.52 Listado de elementos a modificar actualizado (PLANTILLA 12).....	129
Figura 3.53 Listado de documentos a modificar actualizado (PLANTILLA 15). ....	130
Figura 3.54 Registro del estudio de la solicitud de modificación actualizado (PLANTILLA 2) .....	131
Figura 3.55 Resultado del Caso de Prueba 01 (PLANTILLA 16-CP01).....	132
Figura 3.56 Resultados del Caso de Prueba 02 (PLANTILLA 16-CP02).....	133
Figura 3.57 Resultado del Caso de Prueba 03 (PLANTILLA 16-CP03).....	134
Figura 3.58 Resultado del Caso de Prueba de Estrés 01 (PLANTILLA 16-CPR01).....	135
Figura 3.59 Resultado del Caso de Prueba de Estrés 02 (PLANTILLA 16-CPR02).....	136
Figura 3.60 Resultado del Caso de Prueba de Estrés 03 (PLANTILLA 16-CPR03).....	137
Figura 3.61 Nombres de los reportes de pruebas de funcionalidad generados .....	138
Figura 3.62 Nombres de los reportes de pruebas de estrés generados .....	138
Figura 3.63 Resultado del Caso de Prueba de Aceptación 01 (PLANTILLA 16-CPA01). ....	139
Figura 3.64 Resultado del Caso de Prueba de Aceptación 02 (PLANTILLA 16-CPA02). ....	140
Figura 3.65 Resultado del Caso de Prueba de Aceptación 03 (PLANTILLA 16-CPA03). ....	141
Figura 3.66 Notificación de puesta en producción del sistema. ....	142
Figura 3.67 Registro del estudio de la solicitud de modificación actualizado (PLANTILLA 2) .....	143
Figura 3.68 Documentación del sistema LogiHistoricos .....	144
Figura 3.69 Registro de las revisiones de aseguramiento de calidad (PLANTILLA 18). ....	145
Figura 3.70 Registro del control de versiones del sistema (PLANTILLA 19). ....	146
Figura 3.71 Registro de la auditoría de gestión de la configuración (PLANTILLA 20). ....	147

## RESUMEN

**El proyecto de titulación que se presenta contiene** una guía de procedimientos para el proceso de mantenimiento de Software, que permite a los desarrolladores seguir un conjunto de pasos para realizar el mantenimiento de sistemas software, preservando la calidad del producto y haciendo que durante este proceso el sistema no sufra un deterioro de sus características.

La estructura del presente proyecto se resume de la siguiente manera:

**Capítulo I:** Contiene los fundamentos del mantenimiento de Software, incluyendo sus definiciones, necesidades, costes, evolución del software, factores claves, técnicas, estándares, metodologías y tipos de mantenimiento que existen. El capítulo concluye con una descripción de los fundamentos para elaborar guías técnicas.

**Capítulo II:** Contiene la guía de procedimientos para el proceso de mantenimiento de software, con la descripción de la secuencia de pasos a realizar para cada una de las fases del proceso de mantenimiento.

**Capítulo III:** En este capítulo se aplica la guía desarrollada al proceso de mantenimiento de un sistema de registro de información de medidas y generación de métricas de proyectos llamado "LogiHistoricos".

**Capítulo IV:** Este capítulo contiene las Conclusiones y Recomendaciones obtenidas en el desarrollo del proyecto de titulación.

Al final del documento se incluye la Bibliografía, el Glosario y dos anexos que se adjuntan al trabajo presentado.

## PRESENTACIÓN

En el presente proyecto se presenta la propuesta de una “Guía de procedimientos para el proceso de mantenimiento de software”. El trabajo comienza con un conjunto de definiciones básicas usadas para el proceso de mantenimiento de software. Luego se definen los pasos a seguir durante este proceso, y se describen diferentes estándares y metodologías utilizadas con este fin. A continuación, se presentan las definiciones de cada uno de los tipos de mantenimiento de software que existen.

Posteriormente se presenta la introducción, objetivos, alcance y contenido de la guía desarrollada, así como la descripción de la secuencia de pasos a seguir para realizar el proceso de mantenimiento de software que han sido seleccionados para este trabajo.

Una vez elaborada la guía, se la aplica a un caso de estudio que consiste en realizar el mantenimiento de un producto software que se encarga del registro de información de medidas y generación de métricas de proyectos, llamado “LogiHistoricos”, lo que permite evaluar la guía propuesta.

Finalmente se presentan las conclusiones y recomendaciones obtenidas a lo largo del proyecto de titulación.

# CAPÍTULO 1

## ESTUDIO SOBRE LOS FUNDAMENTOS DEL MANTENIMIENTO DE SOFTWARE

### 1.1 DESCRIPCIÓN DE LOS ASPECTOS FUNDAMENTALES DEL MANTENIMIENTO DE SOFTWARE

#### 1.1.1 DEFINICIONES Y TERMINOLOGÍA

Pressman menciona que *"el cambio es algo inevitable cuando se construyen sistemas basados en computadoras"* [1]. El cambio es una de las características fundamentales del software, y quien se encarga de lidiar con esta evolución es el mantenimiento. Un sistema no puede existir por mucho tiempo si no evoluciona.

El estándar IEEE 1219 define el mantenimiento del software como *"la modificación de un producto de software después de su entrega para corregir las fallas, mejorar el rendimiento o para adaptarlo a un nuevo entorno"* [2].

A su vez, el estándar ISO/IEC 12207 determina el mantenimiento como *"uno de los procesos principales en el ciclo de vida del software"* [3], el objetivo es modificar el software existente preservando su integridad; también lo hace en estos mismos términos la ISO/IEC 14764 ésta enfatiza en las entregas previas para la planeación del mantenimiento de software [4].

Analizando las definiciones, podemos concluir que el mantenimiento del software es la modificación de un sistema después de su liberación, para: corregir errores, mejorar el desempeño y otras propiedades, prevenir futuras fallas o adaptar el producto a un nuevo ambiente.

Por lo tanto, el objetivo fundamental del mantenimiento es preservar la funcionalidad y operabilidad, optimizar el rendimiento y aumentar la vida útil del sistema.

### **1.1.2 NECESIDADES DE MANTENIMIENTO**

Las necesidades de mantenimiento se dan para garantizar que el software cumpla satisfactoriamente con los requerimientos solicitados y se aplica a cualquier desarrollo independiente del modelo de ciclo de vida utilizado. El mantenimiento deber ser realizado con el fin de [4] :

- Corregir fallas.
- Mejorar el diseño.
- Implementar mejoras.
- Definir interfaces con otros sistemas.
- Adaptar programas a diferentes tipos de hardware, software, características del sistema y capacidad de las redes de telecomunicaciones.
- Migrar software heredado.
- Retirar el software.

### **1.1.3 COSTOS MAYORITARIOS DE MANTENIMIENTO**

Los costos de mantenimiento de software son los más caros en todo el ciclo de vida del software. A continuación se presentan algunos aspectos técnicos y no técnicos que afectan los costos de mantenimiento [4] :

- Tipo de aplicación.
- Disponibilidad del mantenimiento de software.
- Ciclo de vida del software.
- Características del hardware.
- Calidad de diseño del software, construcción, documentación y pruebas.

Los costos de mantenimiento son altos porque es más caro añadir nuevas funcionalidades cuando el sistema está en funcionamiento que implementar la misma funcionalidad durante el desarrollo. Los factores claves que conducen a costos de mantenimiento más elevados son [5] :

1. **Estabilidad del equipo:** Una vez entregado el sistema, el equipo de desarrollo se disuelve y la gente trabaja en nuevos proyectos. El nuevo equipo de mantenimiento no comprende dicho sistema, por lo que dedican demasiado esfuerzo durante el proceso de mantenimiento en comprender el sistema antes de implementar nuevos cambios.
2. **Práctica de desarrollo deficiente:** El contrato de mantenimiento normalmente está separado del contrato de desarrollo de un sistema. Este factor, junto con la inestabilidad del equipo, implica que no exista ningún compromiso por parte del equipo de desarrollo para construir un software que sea fácil de modificar, lo que ocasiona que se incrementen los costos de mantenimiento.
3. **Habilidad del personal:** El equipo de mantenimiento a menudo no está muy familiarizado con el dominio de la aplicación, por lo que debe tener la habilidad de entender y aprender lo que se requiera para comprender el sistema y poder realizar el mantenimiento.
4. **Antigüedad y estructura del programa:** Con el tiempo, la estructura de los programas tienden a degradarse por los cambios, por lo que los sistemas se vuelven más difíciles de comprender y modificar.

Estos problemas principalmente surgen porque la mayoría de las organizaciones todavía consideran el desarrollo y el mantenimiento como actividades independientes. Las organizaciones deben tener presente que los sistemas evolucionan durante su ciclo de vida a través de un proceso de desarrollo continuo.

#### 1.1.4 EVOLUCIÓN DEL SOFTWARE

*"El software evoluciona a través del tiempo, sin importar su dominio de aplicación, tamaño o complejidad"* [1]. El Mantenimiento de Software dirige este proceso que se presenta cuando se corrigen errores, cuando se adapta a un nuevo ambiente y

cuando el cliente solicita características o funciones nuevas. La mayor parte del trabajo en esta área ha sido realizada por Lehman y Belady. En los últimos 30 años, Manny Lehman y otros investigaron los procesos de evolución para desarrollar una teoría unificada para la evolución del software [1].

Estas leyes deberían tenerse en cuenta al momento de planificar el proceso de mantenimiento del software. A partir de los estudios realizados, propusieron un conjunto de leyes las cuales se presentan en la Tabla 1.1 [5]:

LEY	DESCRIPCIÓN
<b>Cambio continuo</b>	Un sistema que se utiliza en un entorno real debe cambiar o se volverá obsoleto con el pasar el tiempo.
<b>Complejidad creciente</b>	A medida que un programa en evolución cambia, su estructura tiende a ser cada vez más compleja.
<b>Evolución prolongada del programa</b>	La evolución de los programas es un proceso autorregulativo. Los atributos de los sistemas, como el tamaño, el tiempo entre entregas y el número de errores documentados, son aproximadamente invariantes para cada entrega del sistema.
<b>Estabilidad organizacional</b>	Durante el tiempo de vida de un programa, su velocidad de desarrollo es aproximadamente constante e independiente de los recursos dedicados al desarrollo del sistema.
<b>Conservación de la familiaridad</b>	Durante el tiempo de vida de un sistema, el cambio incremental en cada entrega es aproximadamente constante.
<b>Crecimiento continuo</b>	La funcionalidad de los sistemas tiene que crecer continuamente para mantener la satisfacción de los usuarios.
<b>Decremento de la calidad</b>	La calidad de los sistemas comenzará a disminuir a menos que dichos sistemas se adapten a los cambios en su entorno de funcionamiento.
<b>Sistema de retroalimentación</b>	Los procesos de evolución incorporan sistemas de realimentación, para lograr una mejora significativa del producto.

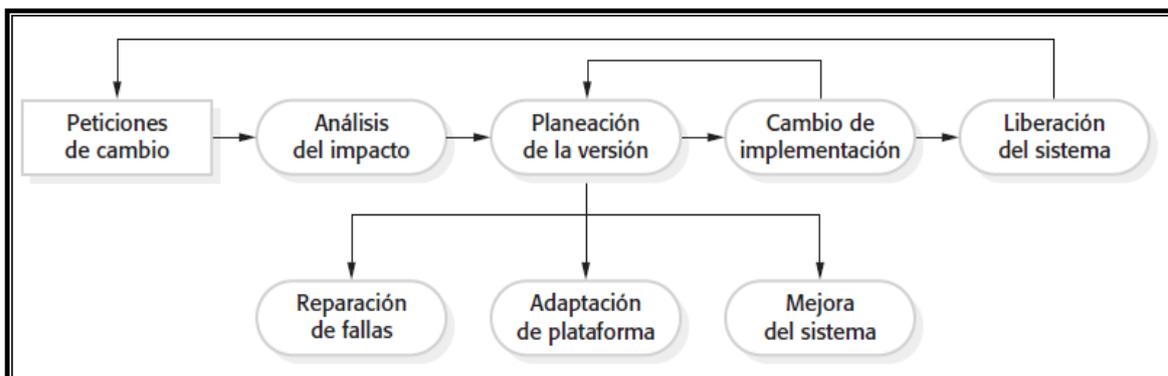
**Tabla 1.1 Leyes de Lehman**

Fuente: I. Sommerville, Ingeniería del Software, Novena Ed, Addison Wesley, 2011

#### 1.1.4.1 Proceso de evolución

Los procesos de evolución del software varían dependiendo del tipo de software, de los procesos de desarrollo utilizados y del personal implicado en el proceso. Estos procesos son cíclicos y continúan durante toda la vida del sistema. Las

actividades fundamentales de un proceso de evolución, se las puede ver en la Figura 1.1 [5] :



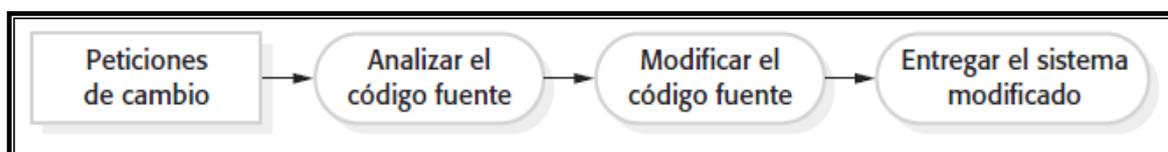
**Figura 1.1 Proceso de evolución**

Fuente: I. Sommerville, Ingeniería del Software, Novena Ed, Addison Wesley, 2011

En algunas ocasiones, las peticiones de cambio están relacionadas con problemas que deben solucionarse de manera urgente. Los cambios urgentes pueden darse por tres razones:

1. Sí ocurre un defecto serio en el sistema que tenga que ser reparado para permitir el funcionamiento normal.
2. Si los cambios en el entorno del sistema operativo tienen efectos inesperados que impiden el funcionamiento normal.
3. Si hay cambios no anticipados en las empresas que utilizan el sistema, tales como la aparición de nuevos competidores o la introducción de una nueva legislación.

En estos casos, se realiza una reparación de emergencia en el sistema para resolver el problema inmediatamente, en la Figura 1.2 se puede ver el proceso de reparación [5]:



**Figura 1.2 Proceso de reparación de emergencia**

Fuente: I. Sommerville, Ingeniería del Software, Novena Ed, Addison Wesley, 2011

Cuando se realiza una reparación de emergencia, se presentan varios problemas, entre los cuales está que se elige una solución funcional rápida en lugar de la mejor solución, esto acelera el proceso de envejecimiento del software de forma que futuros cambios son progresivamente más difíciles y los costos de mantenimiento se incrementan; además, los requerimientos, el diseño del software y el código gradualmente se vuelven inconsistentes. Finalmente, el cambio original se olvida y la documentación del sistema y el código nunca vuelven a ser consistentes [5].

### **1.1.5 FACTORES CLAVES EN EL MANTENIMIENTO DE SOFTWARE**

Hay un gran número de factores claves que se debe tener en cuenta al momento de realizar el mantenimiento de software [4] :

- Factores técnicos.
- Factores administrativos.
- Estimación del esfuerzo.

#### **1.1.5.1 Factores técnicos**

##### **a) Comprensión limitada**

La comprensión limitada se refiere a la rapidez con que un ingeniero de software puede hacer un cambio o una corrección en el software que este individuo no desarrolló. Las investigaciones indican que aproximadamente del 40% al 60% del esfuerzo de mantenimiento está dedicado a comprender el software para ser modificado. Es más difícil comprender el software cuando los cambios no están documentados y los desarrolladores no están disponibles para explicarlo, que es a menudo el caso.

##### **b) Pruebas**

El costo de la repetición de las pruebas puede ser significativo en términos de tiempo y dinero. Las pruebas de regresión o la repetición de pruebas selectivas de

un componente de software para verificar que las modificaciones no han causado efectos no deseados, es importante para el mantenimiento. Además, encontrar tiempo para probar es a menudo difícil.

### **c) Análisis del impacto**

El análisis del impacto describe cómo llevar a cabo de manera rentable un análisis completo de los efectos de un cambio en el software existente. Las personas encargadas del mantenimiento de software deben poseer un conocimiento profundo de la estructura y del contenido del software para realizar el análisis de impacto que identifica a todos los sistemas y productos de software afectados, por una solicitud de cambio de software y desarrollan una estimación de los recursos necesarios para llevar a cabo el cambio.

Los objetivos de análisis del impacto son:

- Determinar el alcance de un cambio con el fin de planificar y ejecutar el trabajo.
- Desarrollar estimaciones precisas de los recursos necesarios para realizar el trabajo.
- Analizar los costos/beneficios de la modificación solicitada.
- Comunicar a otras personas la complejidad de un cambio dado.

### **d) Mantenibilidad**

La IEEE - 1219 define mantenibilidad como la facilidad con que el software puede ser mantenido, mejorado, adaptado o corregido para satisfacer los requerimientos especificados. La ISO/IEC 9126 define mantenibilidad como una de las características de la calidad del software.

Existe una relación directa entre los costos para mantener un producto de software y su mantenibilidad, a mayor mantenibilidad menores costos, la cuantificación de esta propiedad es muy útil para realizar presupuestos de los costos de mantenimiento.

En la actualidad las empresas que desarrollan software no construyen sus productos para que sean mantenibles. Es decir, los desarrolladores tienen otro enfoque y pierden el rumbo de hacer un producto mantenible y de alta calidad, por lo que es necesario que la mantenibilidad sea un objetivo primordial con el fin de reducir los costos del ciclo de vida del software.

#### **1.1.5.2 Factores administrativos**

##### **a) Alineación con los objetivos organizacionales**

Los objetivos organizacionales describen cómo demostrar el retorno de la inversión de las actividades de mantenimiento de software. Por el contrario, el mantenimiento del software a menudo tiene el objetivo de extender la vida del software para el mayor tiempo posible. En ambos casos, el retorno de la inversión es mucho menos claro, de modo que la vista a nivel de la alta gerencia es a menudo una de las principales actividades que consumen recursos significativos sin ningún beneficio cuantificable claro para la organización.

##### **b) Dotación del personal**

La dotación de personal se refiere a la manera de atraer y mantener al personal de mantenimiento. El mantenimiento con frecuencia no es visto como un trabajo atractivo por lo que el personal de mantenimiento de software no es bien visto dentro de la organización.

##### **c) Proceso**

El proceso de software es un conjunto de actividades, métodos, prácticas y transformaciones que el personal usa para desarrollar y mantener el software. A nivel de proceso, las actividades de mantenimiento de software comparten mucho en común con el desarrollo de software. El mantenimiento también requiere varias actividades que no se encuentran en el desarrollo de software.

#### **d) Aspectos organizativos de mantenimiento**

Los aspectos organizativos describen cómo identificar qué organización y / o función será responsable del mantenimiento de software. Al decidir quién se hará responsable del mantenimiento, las organizaciones de ingeniería de software pueden quedarse con el desarrollador original o ir a un equipo independiente. A menudo, se elige la opción de un equipo independiente para asegurarse que el software funcione correctamente y evolucione para satisfacer las necesidades cambiantes de los usuarios. Puesto que hay muchos pros y contras de cada una de estas opciones, la decisión debe ser tomada sobre una base fundamentada.

#### **e) Subcontratación**

La subcontratación de mantenimiento se está convirtiendo en una industria importante. Las grandes empresas están subcontratando portafolios completos de sistemas de software, incluyendo el mantenimiento del software. Frecuentemente, se selecciona la opción de subcontratación para software de misión crítica, ya que las empresas no están dispuestas a perder el control del software utilizado en su negocio.

##### **1.1.5.3 Estimación del esfuerzo.**

Existen pocos métodos para realizar la estimación del esfuerzo de mantenimiento de software; estos métodos ayudan a los responsables del proceso a mejorar la planificación de las actividades de mantenimiento de software y a realizar un análisis de costo/beneficio.

A continuación, se presentan dos métodos de estimación de esfuerzo de mantenimiento de software [6]:

### **a) Estimación por analogía**

La estimación por analogía se basa en datos históricos, es decir compara las características del sistema a modificar con un conjunto de características similares del mismo u otros sistemas que se hayan modificado. Primero se debe seleccionar el conjunto de características, luego se realiza algún tipo de análisis, bien matemático o subjetivo, que enfrente los valores de los atributos históricos con los valores del producto que se va a modificar.

### **b) Modelo COCOMO para mantenimiento**

El modelo tradicional COCOMO de estimación de costos fue modificado por Granja y Barraco para ajustarlo a una versión aplicable para la estimación de los costos de mantenimiento de software [5].

Algunas de sus características principales son:

- Está basado en modelos de estimación matemáticos.
- Es orientado al producto final, no a fases intermedias.
- Se basa en la cantidad de líneas de código del sistema.

## **1.1.6 SEGUIMIENTO AL PROCESO DE MANTENIMIENTO**

En este tiempo moderno donde el software es el corazón de la mayoría de los negocios, es vital que los cambios y la evolución en el software sean controlados haciendo que estos sistemas puedan crecer siendo más funcionales y útiles.

Es importante que se mantenga un seguimiento y control del proceso de mantenimiento, para ello debemos implementar dos procesos importantes que son el proceso de gestión de la configuración y de aseguramiento de la calidad, que se ejecutan paralelamente, durante todo el proceso de mantenimiento de software.

### 1.1.6.1 Gestión de la Configuración

La administración de la configuración es un proceso que utiliza todos los medios necesarios para mantener bajo control el cambio en el software y tener seguridad en la implementación del cambio. En la actualidad, la Gestión de la Configuración es un proceso básico en la administración y el mantenimiento de cualquier sistema [2].

Dentro de este proceso se distinguen las siguientes actividades:

- Identificar los componentes y los cambios.
- Controlar la forma en que los cambios son hechos.
- Auditar los cambios.
- Actualizar datos, grabar y documentar todas las actividades que se hicieron.

Uno de los principales objetivos en la administración de la configuración es reusar. Cuando se quiere reproducir una antigua versión de una aplicación debido a que una nueva versión está teniendo serios problemas o simplemente es que se necesita una funcionalidad específica con unas características especiales de un ambiente en particular. Para cumplir este objetivo es necesario agregar unos objetivos específicos que son: Control, coherencia y costo.

**Control:** La esencia del mantenimiento de software es la evolución natural de los sistemas de software. Si el proceso por el cual los sistemas evolucionan no es controlado, es muy probable que resulte en caos. La administración de la configuración es necesaria porque los sistemas de software tienen una vida útil larga y además están sujetos a cambios.

**Coherencia:** La administración de la configuración asegura que en producción haya productos de software coherentes. También, es preciso estandarizar la documentación del sistema para que se permita versionar y organizar, ya que la documentación de un sistema es muy importante. Realizar versiones

estandarizadas y liberaciones del producto bajo control es necesario para hacer un seguimiento del sistema.

**Costo:** Otro objetivo principal en la administración de la configuración es asegurar que el costo total de los cambios realizados se haya disminuido en lo posible. Esto aplica, tanto para los mantenimientos que se realizarán en el futuro como para los inmediatos.

### **1.1.6.2 Aseguramiento de la Calidad**

Se debe considerar el Aseguramiento de la Calidad del Software cuando se realiza cualquier modificación sobre un sistema existente. Durante el proceso de mantenimiento se deben llevar a cabo los tipos y niveles de aseguramiento de la calidad como: inspecciones, revisiones, auditorías, verificaciones, etc.

En el plan de Mantenimiento se indica el grado y la forma de realizar las revisiones. Una parte fundamental a cuidar será asegurar que la documentación del sistema original continúe describiendo el producto actual [2].

*"La documentación proporciona el fundamento para un buen desarrollo y lo que es más importante, proporciona guías para la tarea de mantenimiento de software"* [1]. A menudo, los responsables del mantenimiento se enfrentan al problema de tener que proporcionar mantenimiento a un sistema con poca o ninguna documentación, lo que provoca que se dificulte el mantenimiento. El proceso de Mantenimiento detallado debería documentarse de forma que todo el personal de mantenimiento siga el mismo proceso [7].

## **1.1.7 TÉCNICAS PARA EL MANTENIMIENTO**

### **1.1.7.1 Comprensión del programa**

Los programadores gastan mucho tiempo leyendo y entendiendo los programas para poder implementar cambios, una documentación clara y concisa puede

ayudar en la comprensión del programa. La comprensión de los programas es uno de los principales factores para alcanzar un Mantenimiento del Software efectivo [4].

### 1.1.7.2 Reingeniería

La reingeniería requiere mucho tiempo, dinero y recursos, por lo que el proceso de reingeniería no se logra en unos cuantos meses, ni siquiera en unos cuantos años. La reingeniería es una actividad de reconstrucción. Un modelo de proceso de reingeniería del software se define en seis actividades, como se muestra en la Figura 1.3 [1] :



Figura 1.3 Modelo de proceso de la reingeniería del software

Fuente: R.S. Pressman, Ingeniería de Software, Séptima Ed. New York: Mac Graw Hill, 2010

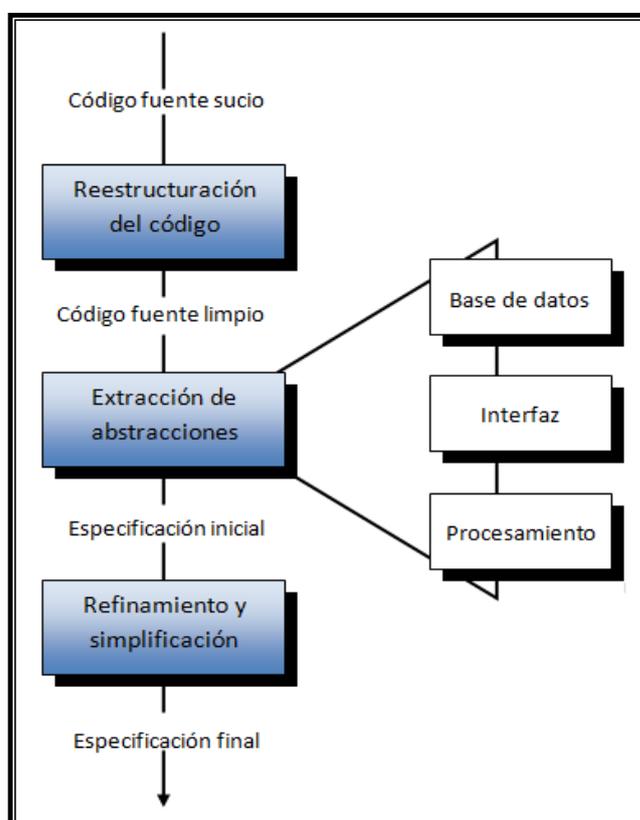
Hacer reingeniería de un sistema software tiene dos ventajas clave que son [5] :

1. **Reducción del riesgo:** Hay un alto riesgo en volver a desarrollar software crítico para los negocios. Pueden cometerse errores en la especificación y presentarse problemas en el desarrollo. Los retrasos en la introducción del nuevo software pueden significar pérdidas en el negocio e incurrir en costes adicionales.

**2. Reducción del costo:** Los costos de hacer reingeniería son significativamente menores que los costos de desarrollar un nuevo software.

### 1.1.7.3 Ingeniería inversa

La ingeniería inversa es un proceso en el cual se analiza un programa para comprender la estructura de su diseño interna en un mayor grado de abstracción que el código fuente. En la Figura 1.4 se representa el proceso de ingeniería inversa [1] :



**Figura 1.4** Proceso de ingeniería inversa

Fuente: R.S. Pressman, Ingeniería de Software, Séptima Ed. New York: Mac Graw Hill, 2010

En el proceso de ingeniería inversa la actividad más importante es la extracción de abstracción, en esta actividad el ingeniero debe evaluar el código fuente y desarrollar una especificación del procesamiento que realiza y de las estructuras de datos que utiliza el programa.

### 1.1.8 ACTORES Y ROLES EN EL MANTENIMIENTO DEL SOFTWARE

Los actores y los roles que juegan durante el proceso de mantenimiento pueden variar considerablemente dependiendo de cada organización. Se indicarán las tareas y responsabilidades de los cuatro roles principales que se deben definir en un proceso de mantenimiento de software [2]:

1. **Solicitante del mantenimiento:** Es quien presenta las solicitudes de modificación y establece los requerimientos necesarios para su implementación.
2. **Gestor de peticiones:** Es el responsable de aceptar o rechazar las peticiones de modificación y decidir el tipo de mantenimiento que se debe aplicar. Además, es el encargado de planificar la cola de peticiones de modificación aceptadas.
3. **Responsable de mantenimiento:** Es el encargado de preparar el proceso de mantenimiento, establecer las normas y procedimientos necesarios para aplicar la guía. Es quien interactúa con el solicitante y trabaja a la par con el equipo de mantenimiento.
4. **Equipo de mantenimiento:** Es el grupo de personas que implementan los cambios presentados en la solicitud de modificación y realizan todas las tareas indicadas por el Responsable de mantenimiento.

## 1.2 DESCRIPCIÓN DEL PROCESO DE MANTENIMIENTO DE SOFTWARE

El proceso de mantenimiento de software cuenta con varios estándares definidos para llevar a cabo el proceso. Entre los estándares más conocidos están los definidos por la ISO/IEC 12207 y la IEEE - 1219. Y a partir de estos estándares, empezaron a surgir, metodologías para mantenimiento de software.

A continuación, se mencionarán algunos de los estándares y metodologías más conocidas para el mantenimiento de software.

## 1.2.1 ESTÁNDARES

### 1.2.1.1 Estándar IEEE 1219

El estándar IEEE 1219 fue el primer estándar especializado en el mantenimiento de software. Este estándar propone una serie de actividades agrupadas en siete fases, tal como se presentan en la Figura 1.5:

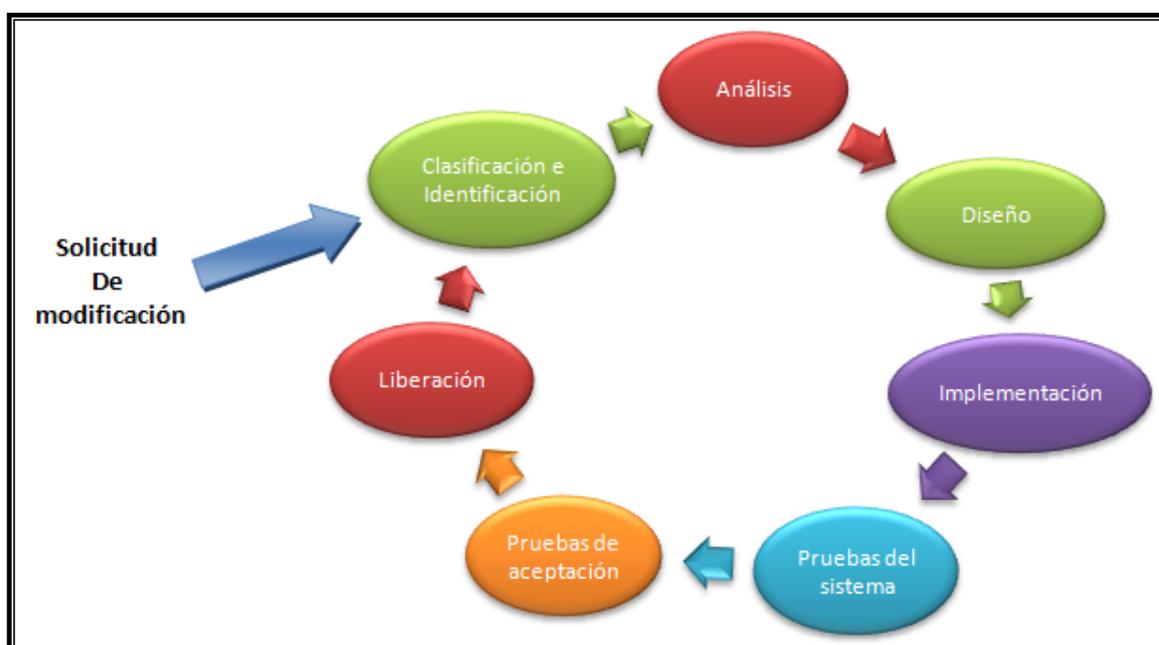


Figura 1.5 Fases del proceso de mantenimiento de software (IEEE 1219-1999)

Fuente: J.A. Martínez Párraga, Planificación y Gestión de Sistemas de Información, Estándar IEEE 1219 de Mantenimiento de Software, 1999

Cada una de estas fases tienen asociadas actividades que permiten gestionar el proceso de Mantenimiento del Software, las cuales son [2]:

#### a) **FASE 1: Identificación, clasificación y priorización del problema.**

En esta fase se identifican, clasifican y asignan prioridades a las solicitudes de modificación. Primero se identifica cual es el problema o solicitud de mejora que reporta el usuario que hizo la petición. Una vez que se ha identificado el

problema, en esta misma fase se clasifica a qué tipo de mantenimiento pertenece dicha petición. Por último, se le da una prioridad a la solicitud de modificación para conocer el orden de atención, en caso de que haya más solicitudes en cola.

Las actividades de esta fase son:

1. Asignación de un número de identificación.
2. Clasificación del tipo de mantenimiento.
3. Análisis de la modificación para determinar si se acepta, se deniega o se evalúa.
4. Realizar una estimación preliminar de la magnitud de la modificación.
5. Priorizar la modificación.
6. Asignar solicitudes de modificación a bloques de tareas planificadas para su implementación.

#### **b) FASE 2: Análisis**

En esta fase se estudia la viabilidad y el alcance de la solicitud de modificación validada y se inicia con la realización de planes preliminares de diseño, implementación, pruebas y liberación del software. El análisis es un proceso iterativo ya que la documentación del producto puede estar en constante cambio. Este proceso iterativo tiene dos componentes que son: el análisis de viabilidad y el análisis detallado.

Las actividades de esta fase son:

1. Realizar un análisis de viabilidad y un reporte de viabilidad con los siguientes datos:
  - Impacto de las modificaciones.
  - Soluciones alternativas.
  - Análisis de los requisitos de conversión.
  - Implicaciones de seguridad.
  - Implicaciones de robustez.
  - Factores humanos.
  - Costes a corto y largo plazo.
  - Beneficios de realizar las modificaciones

2. Realizar un análisis detallado, que incluye:

- Definición de los requisitos fijados para la modificación.
- Identificación de los elementos a modificar.
- Identificación de los elementos de seguridad.
- Identificación de los elementos de robustez.
- Definición de la estrategia de pruebas.
- Desarrollo de un plan de implementación.

**c) FASE 3: Diseño**

En esta fase se estudia la documentación generada en la fase de análisis, la documentación del sistema, el código fuente y la base de datos y se diseña la modificación. Además, se actualiza la documentación del sistema y se procede a crear los casos de prueba e identificar las pruebas de regresión necesarias.

Las actividades de esta fase son:

1. Identificar módulos del software afectados.
2. Modificar la documentación de los módulos de software.
3. Crear casos de prueba para el nuevo diseño.
4. Identificar y crear pruebas de regresión.
5. Identificar documentación y actualizar requerimientos.
6. Actualizar la lista de modificaciones.

**d) FASE 4: Implementación**

Esta fase consiste en realizar la codificación de los cambios validados en la fase de diseño, y a medida que se va codificando se realizan pruebas unitarias; luego de que el código implementado está probado se continúa con la integración del sistema. Después de hacer esta integración se realizan pruebas de regresión para asegurarse que todo el código en conjunto funciona correctamente.

Finalmente en esta fase se obtiene un nuevo software que debe estar bajo la Gestión de Configuración, acompañado de toda la documentación que se generó y se modificó.

Las actividades de esta fase son:

1. Codificación de modificaciones y realización de pruebas.
2. Integración.
3. Análisis de riesgo.
4. Revisión de pruebas preparadas.

**e) FASE 5: Pruebas del sistema**

El objetivo de esta fase consiste en probar todo el sistema totalmente integrado, realizando pruebas de funcionalidad, de interfaz y de regresión para validar que el código que se ha modificado no introduzca errores que no existían antes de la actividad de mantenimiento.

Las actividades de esta fase son:

1. Pruebas de funcionalidad del sistema.
2. Pruebas de las interfaces.
3. Pruebas de regresión.
4. Revisión y preparación de las pruebas de aceptación.

**f) FASE 6: Pruebas de aceptación**

El objetivo de esta fase consiste en realizar las pruebas de aceptación sobre un sistema completamente integrado, en las cuales deben intervenir el cliente y los usuarios finales.

Las actividades de esta fase son:

1. Realizar pruebas de aceptación a nivel funcional.
2. Realizar pruebas de interoperabilidad.
3. Realizar pruebas de regresión.

**g) FASE 7: Liberación**

El objetivo de esta fase consiste en poner en producción la nueva versión del software, para ello se debe realizar un plan de implementación.

Las actividades de esta fase son:

1. Conducir una auditoría de configuración física.
2. Notificar a la comunidad de usuarios.
3. Archivar una versión del sistema para respaldo.
4. Realizar la instalación y dar entrenamiento.

Además de las actividades indicadas, el estándar define una serie de elementos de entrada y salida, así como algunos controles para cada una de estas fases. En la Tabla 1.2 se presentan estos elementos.

Es importante tener en cuenta que es decisión de cada organización definir qué actividades y elementos son relevantes para aplicarlos en el proceso de mantenimiento dentro de la organización. Sin embargo, este estándar puede servir de guía para todas las organizaciones que deseen tener un proceso de mantenimiento bien definido.

ETAPA	ENTRADA	CONTROLES	SALIDAS
<b>Identificación del problema</b>	<ul style="list-style-type: none"> <li>✓ Solicitud de modificación.</li> </ul>	<ul style="list-style-type: none"> <li>✓ Identificador único de la solicitud</li> <li>✓ Entrada de la solicitud de modificación al repositorio</li> </ul>	<ul style="list-style-type: none"> <li>✓ Solicitud de modificación validada</li> <li>✓ Proceso determinado</li> </ul>
<b>Análisis</b>	<ul style="list-style-type: none"> <li>✓ Documentos del proyecto o sistema</li> <li>✓ Solicitud de modificación validada</li> </ul>	<ul style="list-style-type: none"> <li>✓ Revisión técnica</li> <li>✓ Verificación de estrategias de prueba y documentación actualizada</li> <li>✓ Identificación de aspectos de seguridad</li> </ul>	<ul style="list-style-type: none"> <li>✓ Reporte de factibilidad</li> <li>✓ Reporte de análisis detallado</li> <li>✓ Requerimientos actualizados</li> <li>✓ Lista de modificaciones preliminares</li> <li>✓ Plan de implementación</li> <li>✓ Estrategia de pruebas</li> </ul>
<b>Diseño</b>	<ul style="list-style-type: none"> <li>✓ Documentación del sistema</li> <li>✓ Código Fuente</li> <li>✓ Bases de datos</li> <li>✓ Salida de la etapa de análisis</li> </ul>	<ul style="list-style-type: none"> <li>✓ Revisión e inspección del software</li> <li>✓ Verificación del diseño</li> </ul>	<ul style="list-style-type: none"> <li>✓ Lista de modificaciones revisadas</li> <li>✓ Análisis detallado revisado</li> <li>✓ Plan de implementación revisado</li> <li>✓ Línea base del diseño actualizada</li> <li>✓ Plan de pruebas actualizado</li> </ul>
<b>Implementación</b>	<ul style="list-style-type: none"> <li>✓ Código fuente</li> <li>✓ Documentación del sistema</li> <li>✓ Resultados de la etapa de diseño</li> </ul>	<ul style="list-style-type: none"> <li>✓ Revisión e inspección del software</li> <li>✓ Verificación del control de la administración de configuración</li> <li>✓ Verificación de la trazabilidad del diseño</li> </ul>	<ul style="list-style-type: none"> <li>✓ Software actualizado</li> <li>✓ Documentos del diseño actualizados</li> <li>✓ Documentos de pruebas actualizados</li> <li>✓ Documentos de usuario actualizados</li> <li>✓ Material de entrenamiento actualizado</li> <li>✓ Reporte de la preparación de pruebas actualizado</li> </ul>
<b>Pruebas del sistema</b>	<ul style="list-style-type: none"> <li>✓ Documentación actualizada del software</li> <li>✓ Reporte de la preparación de pruebas</li> <li>✓ Sistema actualizado</li> </ul>	<ul style="list-style-type: none"> <li>✓ Control de la administración de configuración de: código, listados, SM, documentación de pruebas</li> </ul>	<ul style="list-style-type: none"> <li>✓ Sistema probado</li> <li>✓ Reporte de pruebas</li> </ul>
<b>Pruebas de aceptación</b>	<ul style="list-style-type: none"> <li>✓ Reporte de la preparación de pruebas</li> <li>✓ Sistema totalmente integrado</li> <li>✓ Planes, casos y procedimientos de las pruebas de aceptación</li> </ul>	<ul style="list-style-type: none"> <li>✓ Pruebas de aceptación</li> <li>✓ Auditoría funcional</li> <li>✓ Establecimiento del sistema base</li> </ul>	<ul style="list-style-type: none"> <li>✓ Nueva línea de base del sistema</li> <li>✓ Reporte de pruebas de aceptación</li> <li>✓ Reporte de la auditoría de configuración funcional</li> </ul>
<b>Liberación</b>	<ul style="list-style-type: none"> <li>✓ Sistema probado y aceptado</li> </ul>	<ul style="list-style-type: none"> <li>✓ Auditoría de configuración física</li> <li>✓ Documento de la descripción de versión</li> </ul>	<ul style="list-style-type: none"> <li>✓ Reporte de auditoría de configuración física</li> <li>✓ Documento de la descripción de versión</li> </ul>

**Tabla 1.2 Entradas, controles y salidas de las distintas etapas del proceso de mantenimiento del software (IEEE 1219-1999)**

**Fuente:** J.A. Martínez Párraga, Planificación y Gestión de Sistemas de Información, Estándar IEEE 1219 de Mantenimiento de Software, 1999

### 1.2.1.2 Estándar ISO/IEC 12207

El estándar ISO/IEC 12207 establece un marco de referencia para los procesos del ciclo de vida del software, por tanto, no es un estándar especializado en mantenimiento de software. En este estándar se especifica que el proceso de mantenimiento se activa cuando el código o la documentación sufren modificaciones debido a errores o necesidades de mejora.

A continuación se describen las actividades y tareas que se deben realizar dentro del proceso de mantenimiento de software [3] :

#### **a) ACTIVIDAD 1: Implementación del proceso.**

En esta actividad se establecen los planes y procedimientos a ejecutar durante el proceso de mantenimiento. También se definen los procedimientos necesarios para gestionar las peticiones de modificación y para implementar el proceso de gestión de la configuración.

Las tareas de esta actividad son:

- Desarrollar planes y procedimientos de mantenimiento.
- Definir procedimientos para las peticiones de modificación.
- Implementar el proceso de gestión de la configuración.

#### **b) ACTIVIDAD 2: Análisis de problemas y modificaciones**

El análisis de los problemas y modificaciones es una actividad crítica, ya que se debe entender el problema, desarrollar una solución y obtener la aprobación para implementarla.

Las tareas de esta actividad son:

- Analizar la petición de modificación.
- Verificar el problema.
- Elaborar alternativas.
- Documentar el problema.
- Obtener aprobación.

**c) ACTIVIDAD 3: Implementación de las modificaciones**

En esta actividad se determinan que elementos del sistema deben modificarse. Además, el responsable del mantenimiento desarrolla y prueba la modificación del producto software.

Las tareas de esta actividad son:

- Determinar documentos a modificar.
- Abrir un proceso de desarrollo para implementar las modificaciones.

**d) ACTIVIDAD 4: Revisión y aceptación del mantenimiento**

El objetivo principal de esta actividad es asegurar que las modificaciones realizadas al sistema se hayan hecho de forma correcta y de acuerdo a los estándares aprobados.

Las tareas de esta actividad son:

- Revisar la integridad del sistema ya modificado.
- Obtener aprobaciones.

**e) ACTIVIDAD 5: Migración**

Durante la vida de un sistema, puede que haya que modificarlo para ejecutarlo en entornos diferentes. Para migrar un sistema a un nuevo entorno, el responsable del mantenimiento necesita determinar las acciones necesarias para lograr la migración, llevarlas a cabo y documentar los efectos.

Las tareas de esta actividad son:

- Asegurar ajuste a la norma.
- Desarrollar un plan de migración.
- Notificar la futura migración a los usuarios.
- Ejecutar paralelamente las operaciones en el antiguo y nuevo entorno.
- Notificar migración.
- Realizar revisión post-operación.
- Archivar datos del entorno antiguo.

**f) ACTIVIDAD 6: Retirada del software**

El producto software se retirará por petición del propietario. Se debería hacer un análisis para ayudar en la toma de decisión antes de retirar un producto software. El análisis a menudo está basado en aspectos económicos y se debería incluir en el Plan de Retiro.

Para retirar un producto software, el mantenedor debería determinar las acciones necesarias para conseguir el retiro y en base a ello, desarrollar y documentar los pasos necesarios para efectuar el retiro.

Las tareas de esta actividad son:

- Desarrollar plan de retiro.
- Notificar futura retirada.
- Implementar las operaciones paralelas y de entrenamiento.
- Notificar retirada.
- Archivar datos del entorno antiguo.

En la norma ISO/IEC 12207 se describe el esqueleto del proceso de mantenimiento de software, pero no se especifica el detalle de cómo se deben de ejecutar o implementar las actividades y tareas propuestas por este estándar. Esta es la gran desventaja que tiene esta norma, ya que el mantenedor finalmente no podrá implementar algunas actividades. Además, este estándar no define métricas para controlar el proceso, y no sugiere ningún método para estimar el esfuerzo del mantenimiento de software.

En la Tabla 1.3 se presenta un cuadro comparativo de las principales características de los estándares IEEE 1219 e ISO/IEC 12207.

CARACTERÍSTICA	IEEE 1219	ISO/IEC 12207
<b>Enfoque</b>	Define un proceso iterativo con actividades y tareas.	Define un marco global con actividades y tareas generales.
<b>Estilo</b>	Describe como ejecutar las tareas.	Es menos descriptivo.
<b>Control del proceso</b>	Es más riguroso.	Es menos riguroso.
<b>Estructura del procesos</b>	Detalla las tareas de cada una de las actividades.	Presenta un esqueleto de las actividades y tareas.
<b>Ventajas</b>	Tiene mejor definida la ejecución de las tareas.	A partir de este estándar se pueden generar metodologías, ya que es más flexible.
<b>Desventajas</b>	Menos flexible.	No hace énfasis en como ejecutar las tareas.
<b>Resumen</b>	Es el primer estándar que se creó para el mantenimiento de software, donde se definen actividades y tareas específicas para llevar a cabo el proceso de mantenimiento.	Es un estándar que propone un marco con actividades y tareas generales, pero no hace énfasis en cada tarea. A partir de este estándar se han iniciado nuevas metodologías para mantenimiento de software.

Tabla 1.3 Cuadro comparativo entre la IEEE 1219 e ISO/IEC 12207

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente de datos: J.A. Martínez Párraga, Estándar IEEE 1219 de Mantenimiento de Software, 1999

Tecnología de la información. Proceso del ciclo de vida del software, Norma Técnica Peruana, NTP-ISO/IEC 12207, AENOR, 2006

## 1.2.2 METODOLOGÍAS

### 1.2.2.1 Metodología Mantema

MANTEMA es una metodología para el mantenimiento de software que fue desarrollada entre el grupo Alarcos del Departamento de Informática de la Universidad de Castilla La Mancha y la empresa Atos ODS, ambos de España. El objetivo de esta metodología es hacer que el mantenimiento sea un proceso controlable y medible, mediante la identificación y definición clara de todos los elementos que intervienen en el proceso de mantenimiento. Pero esta metodología puede demandar mucho tiempo y esfuerzo para ser implementada por una pequeña empresa [8].

Esta metodología se desarrolló bajo el estándar de la ISO/IEC 12207; el proceso está compuesto por varias actividades y cada actividad está compuesta por varias tareas.

MANTEMA detalla que cada tarea debe tener lo siguiente:

- **Entradas:** Son el insumo de la tarea. Pueden ser programas, documentos, diagramas, etc.
- **Salidas:** Son los elementos generados por una tarea y pueden convertirse en la entrada de una tarea siguiente.
- **Técnicas:** Son las técnicas que se pueden utilizar para ejecutar la tarea.
- **Métricas:** Permiten tener el proceso y el producto bajo control.
- **Responsables:** Son los roles encargados de ejecutar las tareas.
- **Interfaces con otros procesos:** Son relaciones que se pueden dar con otros procesos, por ejemplo con el proceso de gestión de la configuración.

A continuación se presentan las actividades y tareas definidas por esta metodología:

### **1. Actividades y tareas iniciales comunes**

#### **Actividad I0. Estudio inicial**

- Tarea I0.1 Iniciar y recoger información.
- Tarea I0.2 Preparar propuesta de mantenimiento.
- Tarea I0.3 Contrato.

#### **Actividad I1. Planificación del proceso**

- Tarea I1.1 Planificar calendario y responsables.
- Tarea I1.2 Adquirir conocimiento de la aplicación.
- Tarea I1.3 Desarrollar planes.
- Tarea I1.4 Definir procedimientos de petición de modificación.
- Tarea I1.5 Implementar proceso de gestión de configuración.

- Tarea I1.6 Preparar entornos de pruebas.

### **Actividad I2. Análisis de la petición de modificación**

- Tarea I2.1 Recibir petición de modificación.
- Tarea I2.2 Decidir tipo de mantenimiento.

## **2. Actividades y tareas del mantenimiento no planificado (correctivo urgente)**

### **Actividad NP1. Análisis del error**

- Tarea NP1.1 Investigar y analizar causas.

### **Actividad NP2. Intervención correctiva urgente**

- Tarea NP2.1 Realizar acciones correctivas.
- Tarea NP2.2 Cumplir documentación.
- Tarea NP2.3 Ejecutar pruebas unitarias.

### **Actividad NP3. Cierre intervención**

- Tarea NP3.1 Pasar a producción.

## **3. Actividades y tareas del mantenimiento planificado**

Puesto que no todas las actividades del mantenimiento planificado son aplicables a todos los tipos de mantenimiento (correctivo no urgente, perfectivo, preventivo y adaptativo), indicamos con las siguientes siglas los tipos de mantenimiento a los que cada tarea es aplicable.

Siglas: CP-->mantenimientos correctivo no urgente y perfectivo.

A-->adaptativo.

P-->preventivo.

### **Actividad P1. Análisis de la petición**

- Tarea P1.1 Valorar petición (CP/P).
- Tarea P1.2 Documentar posibles soluciones (CP/P).

- Tarea P1.3 Elegir alternativas adecuadas (CP).

### **Actividad P2. Intervención y pruebas**

- Tarea P2.1 Planificar calendario (A).
- Tarea P2.2 Realizar copiar del producto software (A).
- Tarea P2.3 Ejecutar intervención (CP/P/A).
- Tarea P2.4 Ejecutar pruebas unitarias (CP/P/A).
- Tarea P2.5 Ejecutar pruebas de integración (CP/P/A).
- Tarea P2.6 Ejecutar paralelamente en software antiguo y nuevo (CP/P/A).
- Tarea P2.7 Verificar y valida corrección con el cliente (CP/P/A).
- Tarea P2.8 Redocumentar manual de usuario (CP/P/A).
- Tarea P2.8 Pasar a producción (CP/P/A).
- Tarea P2.8 Realizar revisión (CP/A).

### **Actividad P3. Cierre de intervención**

- Tarea P3.1 Archivar datos del producto software inicial (A).

## **4. Actividades y tareas finales**

### **Actividad F1. Registro de la intervención**

- Tarea F1.1 Registrar intervención.

### **Actividad F2. Actualización de la base de datos histórica**

- Tarea F2.1 Recoger información de la intervención.
- Tarea F2.2 Actualizar base de datos.

### **Actividad F3. Retirada**

- Tarea F3.1 Desarrollar plan de retirada.
- Tarea F3.2 Notificar futura retirada.
- Tarea F3.3 Ejecutar paralelo.
- Tarea F3.4 Notificar retirada.
- Tarea F3.5 Almacenar datos del entorno antiguo.

### Actividad F4. Fin de la externalización

- Tarea F4.1 Entrega del inventario y de la documentación.
- Tarea F4.2 Traspaso de experiencia y formación.
- Tarea F4.3 Cesión definitiva del servicio.

#### 1.2.2.2 Metodología MÉTRICA V3

Métrica V3 es una metodología que ofrece a las organizaciones un instructivo útil para las actividades que dan soporte al ciclo de vida del software. Esta metodología define un flujo corto para el proceso de mantenimiento con 4 actividades y 10 tareas. Fue ideada pensando en ayudar a las pequeñas organizaciones que desean disponer de una metodología para llevar a cabo el proceso de mantenimiento de software [9].

La estructura propuesta por Métrica Versión 3 para el proceso de Mantenimiento de software comprende las siguientes actividades que se pueden ver en la Figura 1.6:

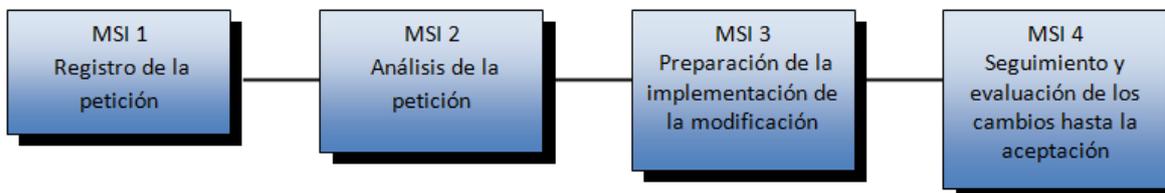


Figura 1.6 Proceso de mantenimiento (Métrica V3, 2000)

Fuente: Ministerio de Administración Pública de España, Metodología de Planificación, Desarrollo y Mantenimiento de Sistemas de Información, Metodología Métrica Versión 3, 2000

A continuación se indican las actividades y tareas definidas por esta metodología:

#### 1. Actividad MSI 1. Registro de la petición

- Tarea MSI 1.1 Registro de la petición.
- Tarea MSI 1.2 Asignación de la petición.

#### 2. Actividad MSI 2. Análisis de la petición

- Tarea MSI 2.1 Verificación y estudio de la petición.
- Tarea MSI 2.2 Estudio de la propuesta de solución.

### 3. **Actividad MSI 3. Preparación de la implementación de la modificación**

- Tarea MSI 3.1 Identificación de elementos afectados.
- Tarea MSI 3.2 Establecimiento del plan de acción.
- Tarea MSI 3.3 Especificación del plan de pruebas de regresión.

### 4. **Actividad MSI 4. Seguimiento y evaluación de los cambios hasta la aceptación**

- Tarea MSI 4.1 Seguimiento de los cambios.
- Tarea MSI 4.2 Realización de las pruebas de regresión.
- Tarea MSI 4.3 Aprobación y cierre de la petición.

## 1.3 DESCRIPCIÓN DE LOS TIPOS DE MANTENIMIENTO DE SOFTWARE

MANTEMA ha determinado que existen cinco tipos de mantenimiento [8]:

- Mantenimiento Correctivo Urgente
- Mantenimiento Correctivo No Urgente
- Mantenimiento Perfectivo
- Mantenimiento Adaptativo
- Mantenimiento Preventivo

En la Figura 1.7 se muestra la representación de cada tipo de mantenimiento:

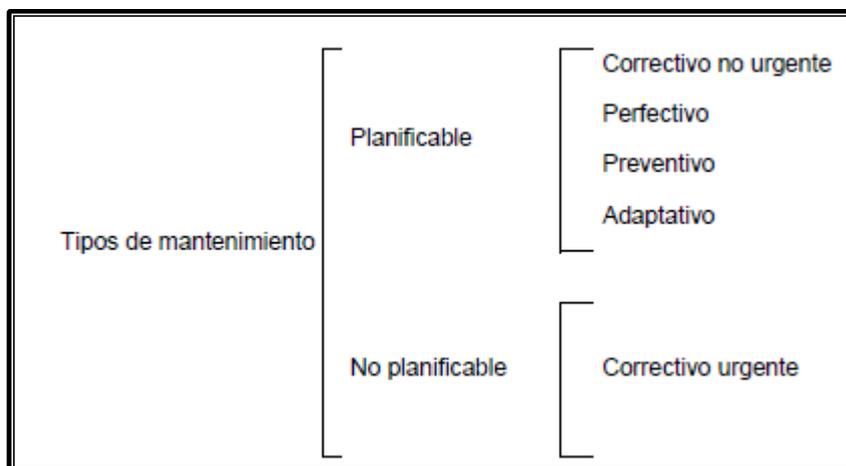


Figura 1.7 Tipos de mantenimiento según la Metodología – MANTEMA  
Fuente: M. Polo Usaola, MANTEMA: Una metodología para mantenimiento de Software, España, 2000

Según Métrica Versión 3, se establecen los siguientes tipos de mantenimiento:

- Mantenimiento Correctivo
- Mantenimiento Evolutivo
- Mantenimiento Adaptativo
- Mantenimiento Perfectivo

Cabe recalcar que los últimos dos tipos de mantenimiento se mantienen fuera de la Métrica Versión 3, por la razón que requieren de actividades diferentes de los del proceso de desarrollo [9].

En esta tesis nos basaremos en el estándar IEEE 1219, que expresa que “existen diferentes tipos de Mantenimiento de Software dependiendo de las demandas de los usuarios del producto Software a mantener” [2] y son los siguientes:

- Mantenimiento correctivo
- Mantenimiento de emergencia
- Mantenimiento adaptativo
- Mantenimiento perfectivo
- Mantenimiento preventivo

Estos tipos de mantenimiento se explican a continuación.

### **1.3.1 MANTENIMIENTO CORRECTIVO**

Se define como la modificación de un producto software después de su puesta en producción (entrega) para corregir los fallos que se han descubierto.

**Objetivo:** Localizar y eliminar los posibles defectos del programa.

### **1.3.2 MANTENIMIENTO DE EMERGENCIA**

Se trata de un mantenimiento correctivo, el cual mantiene operativo al sistema, pero el mismo es realizado sin planificarlo previamente.

**Objetivo:** Realizar el mantenimiento lo más pronto posible dando soluciones eficaces. Dar soluciones de una manera eficaz.

### 1.3.3 MANTENIMIENTO ADAPTATIVO

Este mantenimiento se encarga de mantener operativo un programa cuando se realizan cambios en el entorno de producción. Es importante mencionar que este tipo de mantenimiento es uno de los más usuales ya que se debe a los cambios que se producen en la tecnología informática, dejando sin operar a los productos software que han sido desarrollados por la competitividad que se da entre empresas, dando como resultado que el software sea más utilizado mientras ocurran más modificaciones.

**Objetivo:** Modificar un programa en los diferentes cambios en el entorno, es decir, que se realicen cambios en el hardware o software en el que está siendo ejecutado.

### 1.3.4 MANTENIMIENTO PREVENTIVO

El Mantenimiento Preventivo consiste en realizar la modificación del producto Software mejorando las propiedades del mismo, sin alterar las especificaciones dando facilidad al usuario a realizar las tareas de mantenimiento.

**Objetivo:** Facilitar el mantenimiento por los cambios que se han realizado al producto Software.

### 1.3.5 MANTENIMIENTO PERFECTIVO

Se basa en la modificación de un producto Software para mejorar el rendimiento del mismo, añadiendo nuevas funcionalidades que son requeridas por el usuario y que mejorarán las ya existentes. Según IEEE 1219, se concluye que esta modificación influye directamente en los costes del mantenimiento.

**Objetivo:** Realizar los cambios adecuados que resulten de las solicitudes de los usuarios [4].

En la Figura 1.8 se observa el esfuerzo dedicado a cada tipo de mantenimiento [10].

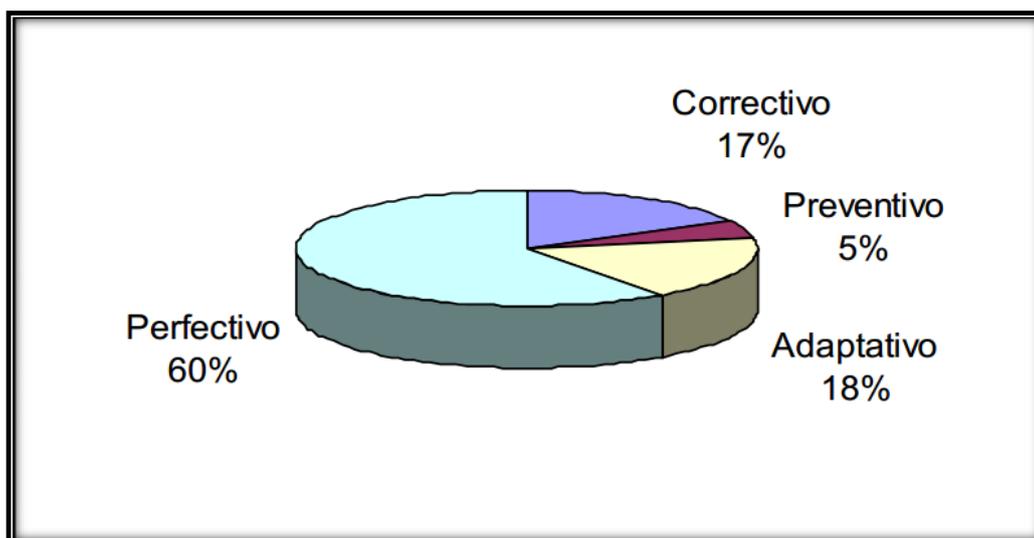


Figura 1.8 Esfuerzo dedicado a cada tipo de mantenimiento

Fuente: P.F Ruiz Macario, Mantenimiento del Software S1, 2001, <http://alarcos.esi.uclm.es/per/fruiz/curs/mso/trans/s1.pdf>

Estadísticamente se da a conocer que el 20 por ciento del trabajo de mantenimiento se basa en “componer errores”. El 80 por ciento restante se encarga de mejorar los sistemas existentes, añadir funcionalidades o rediseñar una aplicación para usarla en el futuro. Por lo tanto se concluye que todas estas actividades que se van a realizar en el mantenimiento absorben demasiado esfuerzo [1].

Según Ian Sommerville, no hay una distinción clara entre estos tipos de mantenimiento, es decir, cuando se adapta el sistema a un nuevo entorno, sin duda se pueden añadir funcionalidades, aprovechando de esta manera las ventajas que darán como resultado las nuevas características del entorno [5].

## **1.4 ELABORACIÓN DE GUÍAS TÉCNICAS**

Para elaborar un guía técnica existen muchas metodologías, las mismas que pueden adaptarse dependiendo del tema a tratarse, ya que una guía técnica depende del uso y las condiciones para las que fue diseñada, sin embargo sus características y funciones básicas deben permanecer invariables.

Existen varios modelos para presentar una guía, esto depende de su contenido ya que no existe uniformidad, la cual varía según los objetivos y propósitos de cada una así como de la organización en la que se va aplicar la guía, por lo tanto es recomendable que la empresa adopte normas generales que uniformicen tanto el contenido de la guía como su forma de presentación [11].

### **1.4.1 CARACTERÍSTICAS Y FUNCIONES DE UNA GUÍA**

Una guía técnica debe tener las siguientes características:

- Proporcionar información sobre el tema para la cual se va a elaborar la guía.
- Establecer objetivos y actividades.

Una guía técnica debe tener las siguientes funciones:

- Establecer recomendaciones para orientar el trabajo de los lectores.
- Establecer actividades integradas de aprendizaje en el lector, para así asegurar o garantizar la aplicación de la guía.

### **1.4.2 ESTRUCTURA DE UNA GUÍA TÉCNICA**

Los elementos básicos que deben integrar una guía, por ser los más relevantes para los objetivos que se persiguen con su elaboración son [11]:

#### **1. IDENTIFICACIÓN:**

La carátula de la guía debe incorporar la siguiente información:

- A. Logotipo de la Organización

- B. Nombre de la Organización
- C. Título del Manual
- D. Nombre y Apellido de los responsables de su elaboración
- E. Fecha de elaboración

En la Figura 1.9 se puede ver la ubicación de cada elemento del formato de identificación de una guía técnica.

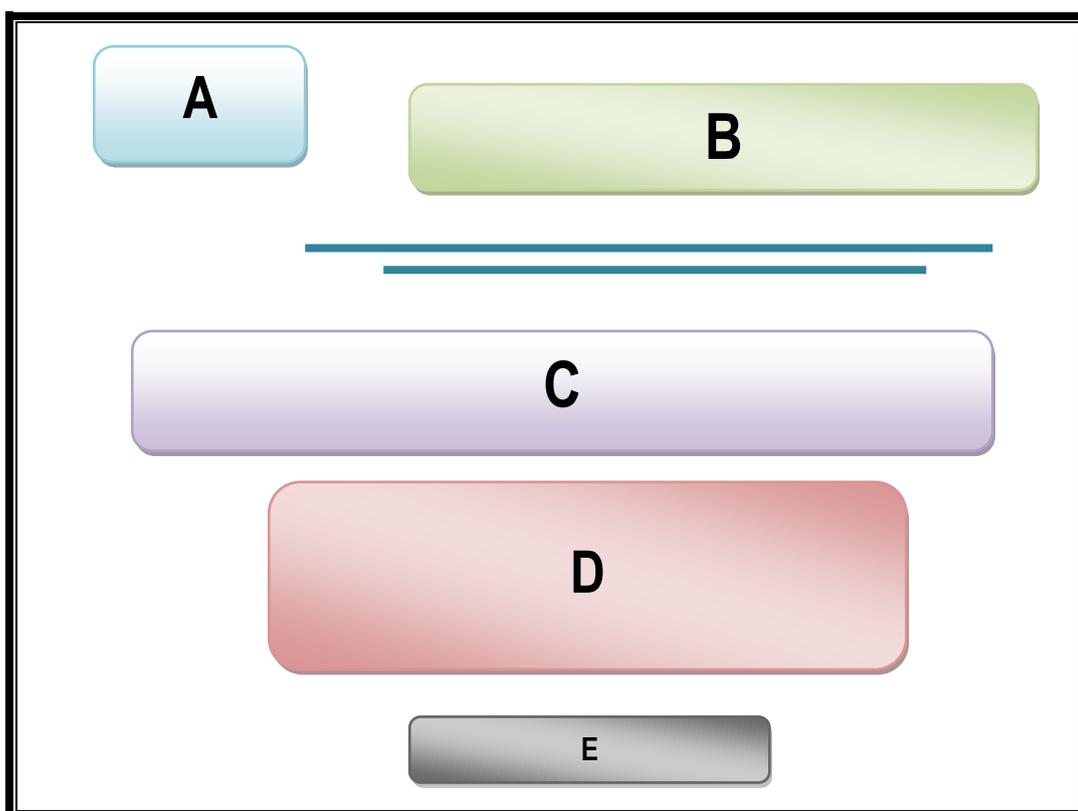


Figura 1.9 Ubicación de cada elemento del formato de identificación de una guía técnica.

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: C.Contreras, Guía técnica para elaboración de manuales de procedimientos

## 2. CONTENIDO:

Es la presentación, de manera sintética y ordenada, de los apartados principales que constituyen la guía, a efecto de unificar la presentación del documento. El contenido puede seguir el siguiente orden [11]:

- a) **Introducción:** En este apartado se señalan, de manera breve, los siguientes puntos:

- Razones y propósitos que motivaron a la elaboración del documento
- A quiénes está dirigido el documento, es decir, sus usuarios
- Forma en que debe utilizarse y actualizarse.

Al formular la Introducción, se deberá emplear un vocabulario sencillo, a efecto de facilitar su entendimiento, así como comprender en su totalidad los puntos antes mencionados.

La introducción proporciona una idea breve, pero a la vez exacta de los diversos aspectos que componen el trabajo. Se trata de realizar un planteamiento claro, conciso y ordenado del tema de investigación, de su importancia, de sus implicaciones, así como de la manera en que se ha creído conveniente abordar el estudio de sus diferentes elementos.

- b) Objetivos:** En este apartado se deberá anotar el propósito a alcanzar, y de acuerdo a la Organización, dónde y durante cuánto tiempo será útil la guía; el objetivo debe responder a las preguntas: ¿Qué se quiere lograr? y ¿Para qué se hace?

El objetivo deberá iniciar con un verbo en infinitivo y, en lo posible, evitar utilizar gerundios y adjetivos calificativos; deberá ser breve, claro y preciso.

- c) Alcance:** Aquí se delimita el ámbito de la aplicación de las actividades realizadas en la Guía.

- d) Desarrollo de contenidos:** En este apartado se presentan de forma detallada los puntos y actividades tratadas en la guía para alcanzar los objetivos planteados en el alcance.

- e) Anexos:** Los anexos son documentos de apoyo o adicionales de consulta que se deberán tomar en cuenta para llevar a cabo una actividad o trámite dentro del procedimiento.

## **CAPÍTULO 2**

# **ELABORACIÓN DE LA GUÍA DE PROCEDIMIENTOS PARA EL PROCESO DE MANTENIMIENTO DE SOFTWARE**

### **2.1 INTRODUCCIÓN**

La guía de procedimientos que se define en el presente capítulo tiene como finalidad, brindar una alternativa clara para realizar el proceso de mantenimiento de software.

Debido a las diferencias que pueden existir en cada organización, es difícil definir un proceso de mantenimiento que sea análogo a cualquier tipo de organización. Por lo tanto, se han tomado las fases propuestas por el estándar de la IEEE sobre mantenimiento de software IEEE 1219, para dar una idea de las fases que pueden constituir el proceso de mantenimiento del software. Además, se han incluido algunas actividades y tareas propuestas en las metodologías MANTEMA y METRICA V3.

Es necesario tener en cuenta que estas actividades y tareas del proceso de mantenimiento propuestas en esta guía sólo son sugerencias. Es decisión de cada organización definir qué actividades son relevantes, así como los elementos que participarán en éstas. Sin embargo, lo que aquí se muestra es una idea de lo que resulta recomendable considerar con el fin de tener un mejor control del proceso de mantenimiento.

Esta propuesta puede servir de guía para todas aquellas organizaciones que deseen tener un proceso de mantenimiento bien definido, de manera que permita la mejora continua del mismo.

### **2.1.1 VISIÓN GENERAL DE LA GUÍA**

En esta sección se indica el esquema general del contenido de la guía. Se detalla cada una de las fases del proceso y como los flujos de apoyo interactúan con ellas. Esta guía propone una serie de actividades agrupadas en siete fases, las cuales son:

1. Identificación, Clasificación y Priorización del problema.
2. Análisis
3. Diseño
4. Implementación
5. Pruebas del Sistema
6. Pruebas de Aceptación
7. Liberación

Adicionalmente se definen dos flujos de apoyo que se ejecutan a lo largo de las diferentes fases del proceso. Estos son:

- Aseguramiento de la Calidad
- Gestión de la Configuración.

La Figura 2.1 muestra el esquema general del contenido de la guía.

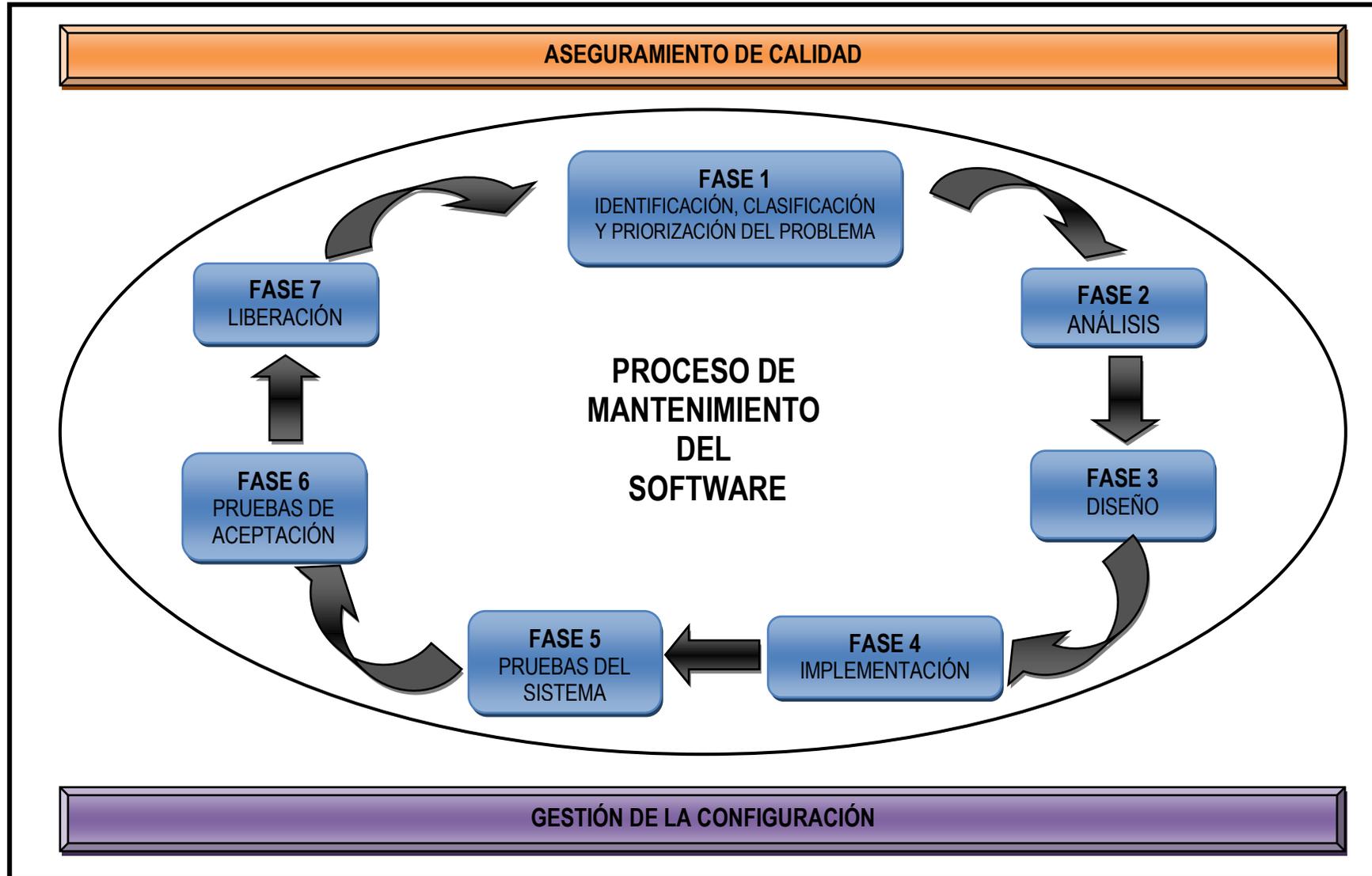


Figura 2.1: Esquema general del contenido de la guía.

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente de datos: J.A. Martínez Párraga, Planificación y Gestión de Sistemas de Información, Estándar IEEE 1219 de Mantenimiento de Software, 1999

### **2.1.2 ESTRUCTURA DE LA GUÍA**

La guía tiene 7 fases que resultan ser los pilares de su estructura, se desarrollan secuencialmente y por ser parte de un proceso de mejora continua, éstas pueden volverse a ejecutar en otra mejora.

Asimismo, los pilares de cada fase son las actividades, que pueden desarrollarse en forma paralela o secuencial. Para cada fase, existen varias actividades.

La estructura de cada actividad es similar para cada una existen roles asignados y elementos de entrada provenientes de fases anteriores; los roles y los elementos de entrada derivan en tareas específicas que generan uno o varios productos entregables, los cuales pueden constituir elementos de entrada de las etapas o fases siguientes.

La Figura 2.2 muestra la estructura del contenido de la guía.

### **2.1.3 RESUMEN DE LA GUÍA**

En la Figura 2.3, se presenta un resumen de las actividades de cada una de las fases del proceso de mantenimiento que contiene la guía, así como también se indican los entregables generados en cada una de las fases por tipo de mantenimiento.

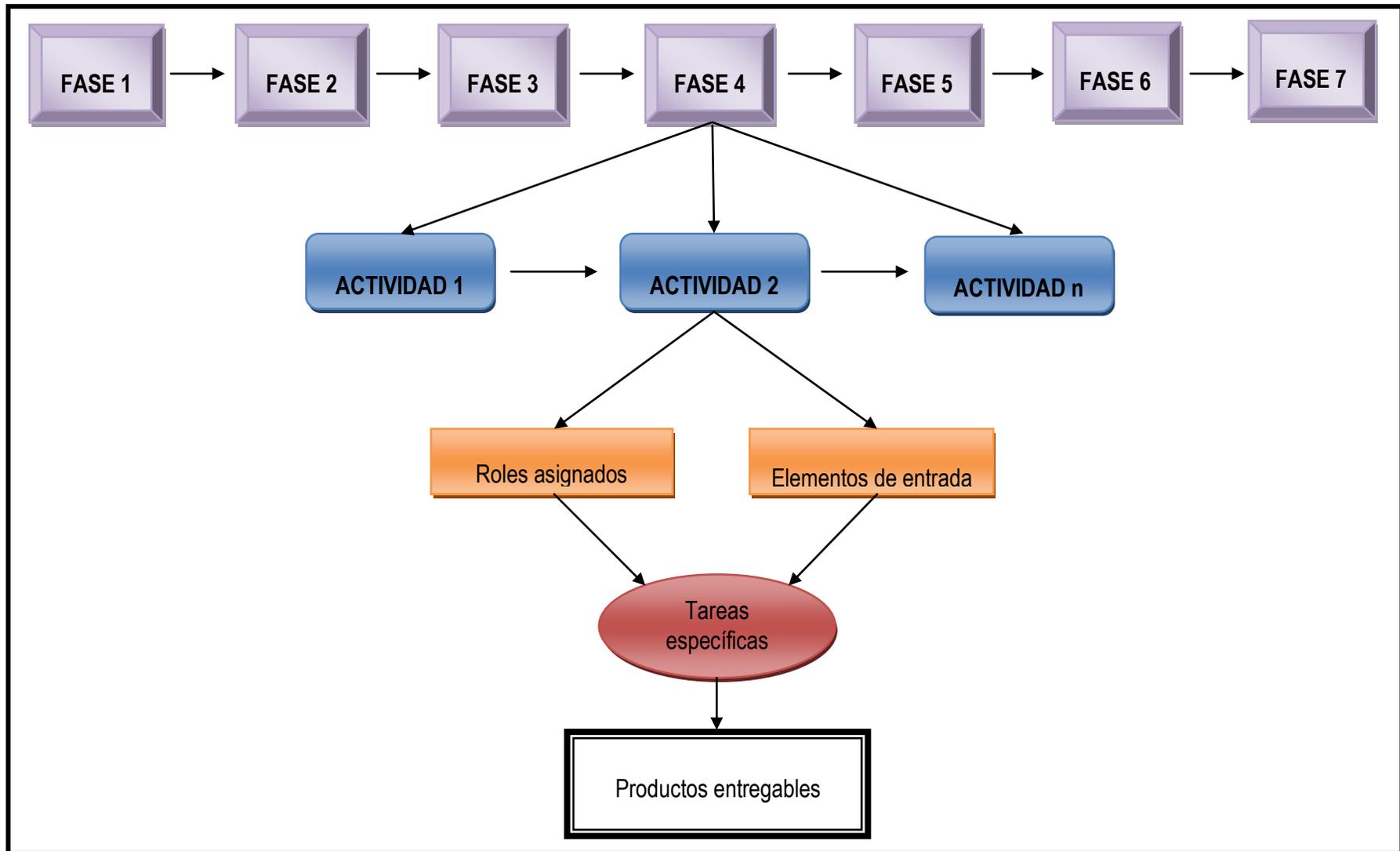


Figura 2.2: Estructura del contenido de la guía.  
Elaborado por: Balseca Marcela, Caiza Gabriela

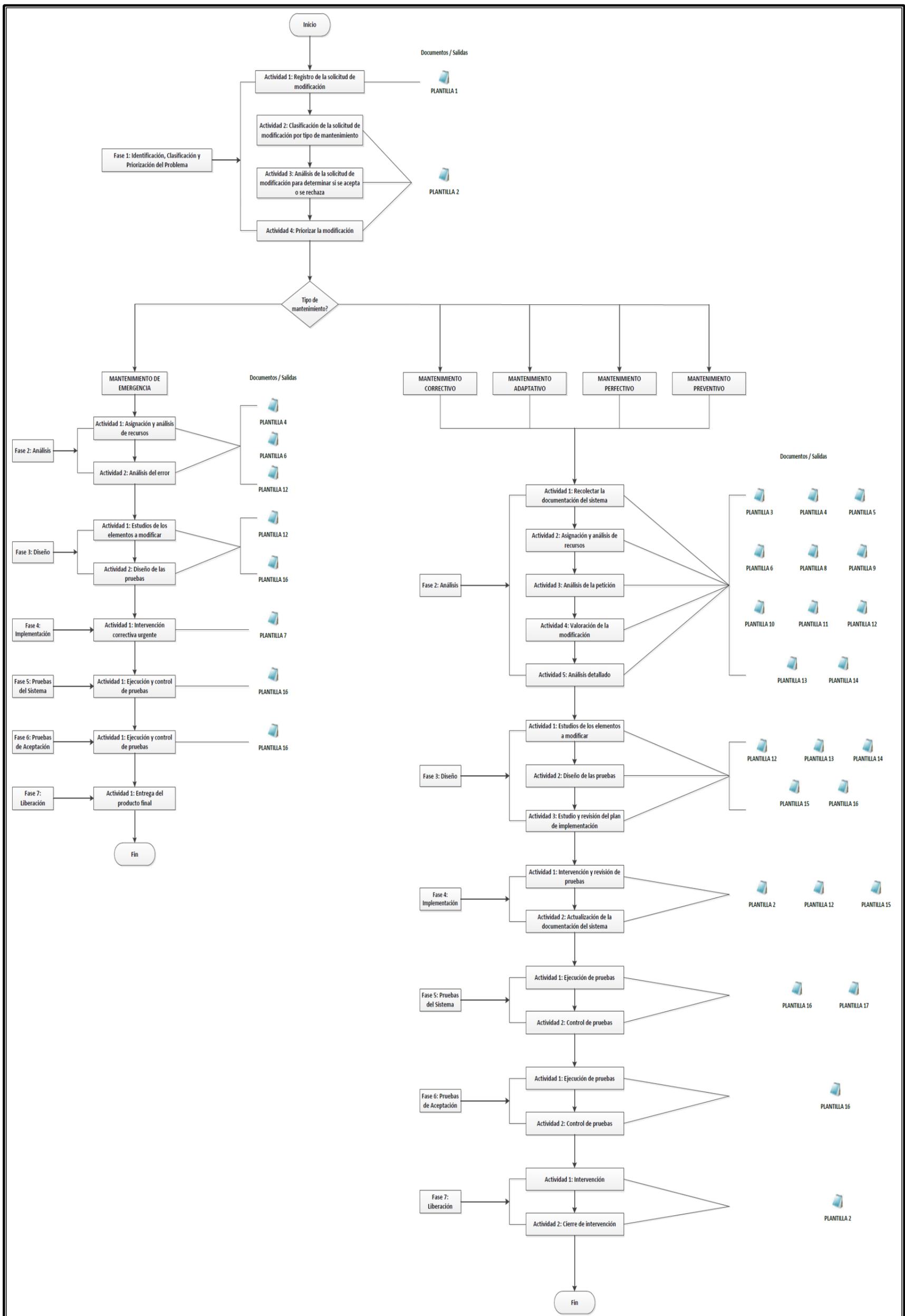


Figura 2.3: Fases, actividades y entregables que contiene la guía.  
Elaborado por: Balseca Marcela, Caiza Gabriela

## **2.2 OBJETIVOS Y ALCANCE**

### **2.2.1 OBJETIVOS**

- Proveer procedimientos claros que garanticen el éxito al realizar el mantenimiento de un sistema software.
- Definir las principales actividades y tareas a seguir en cada uno de los tipos de mantenimiento de software.
- Establecer los documentos que deben generarse como resultado de las tareas realizadas en cada una de las fases del proceso de mantenimiento de software.
- Ayudar al equipo de trabajo a organizar sus esfuerzos, optimizar el uso de recursos y alcanzar sus objetivos; mientras implementa el proceso de mantenimiento.
- Facilitar el proceso de mantenimiento de un sistema software liberado independientemente de cuál sea el requerimiento de cambio.

### **2.2.2 ALCANCE**

La presente guía es aplicable para las organizaciones que realizan el mantenimiento a sus propias aplicaciones y no contratan a empresas externas para que lo realicen.

Para aplicar la guía se requiere de la documentación del sistema. Si la documentación del sistema no está disponible o no es suficiente, y el código fuente es la única representación fiable del sistema, se recomienda aplicar el método de la ingeniería inversa. Hay que tener en cuenta que la documentación es una excelente práctica para minimizar problemas en el proceso de mantenimiento.

## 2.3 CONTENIDO DE LA GUÍA



**NOMBRE Y LOGOTIPO  
DE LA  
EMPRESA QUE APLICARÁ LA GUÍA**

---

---

**GUÍA DE PROCEDIMIENTOS  
PARA  
EL PROCESO DE MANTENIMIENTO DE SOFTWARE**

**ELABORADO POR:**

**MARCELA BALSECA  
GABRIELA CAIZA**

**FECHA:**

**06 / 01 / 2014**

## **I. DESCRIPCIÓN DEL SISTEMA**

En esta sección se describe el sistema que va a ser sujeto de mantenimiento.

## **II. PROCEDIMIENTOS**

### **1. ACTIVIDADES Y TAREAS INICIALES**

#### **1.1. Fase 1: Identificación, clasificación y priorización del problema**

En esta fase se deben identificar, clasificar y priorizar las solicitudes de modificación del software, siguiendo las siguientes actividades:

##### **ACTIVIDAD 1. Registro de la solicitud de modificación.**

Siga las siguientes tareas:

- **TAREA 1.1. Recibir la solicitud de modificación:** El Solicitante deberá entregar la solicitud de modificación llenando la PLANTILLA 1 mostrada en la Figura A.1 (Anexo A), misma que es recibida y registrada por el Gestor de Peticiones en un repositorio para su control.
- **TAREA 1.2. Asignar un número de identificación:** El Gestor de peticiones deberá asignar un identificador único a la solicitud de modificación registrada en la PLANTILLA 1.

##### **ACTIVIDAD 2. Clasificación de la solicitud de modificación por tipo de mantenimiento**

Siga las siguientes tareas:

- **TAREA 2.1. Seleccionar el tipo de mantenimiento:** El Gestor de Peticiones seleccionará el tipo de mantenimiento requerido para la solicitud registrada, teniendo en cuenta la información presentada en la solicitud de modificación y apoyándose en el diagrama de flujo mostrado en la Figura

2.3. En la PLANTILLA 2, mostrada en la Figura A.2 (Anexo A), se registrará el tipo de mantenimiento seleccionado y la razón por la cual escogió ese tipo de mantenimiento.

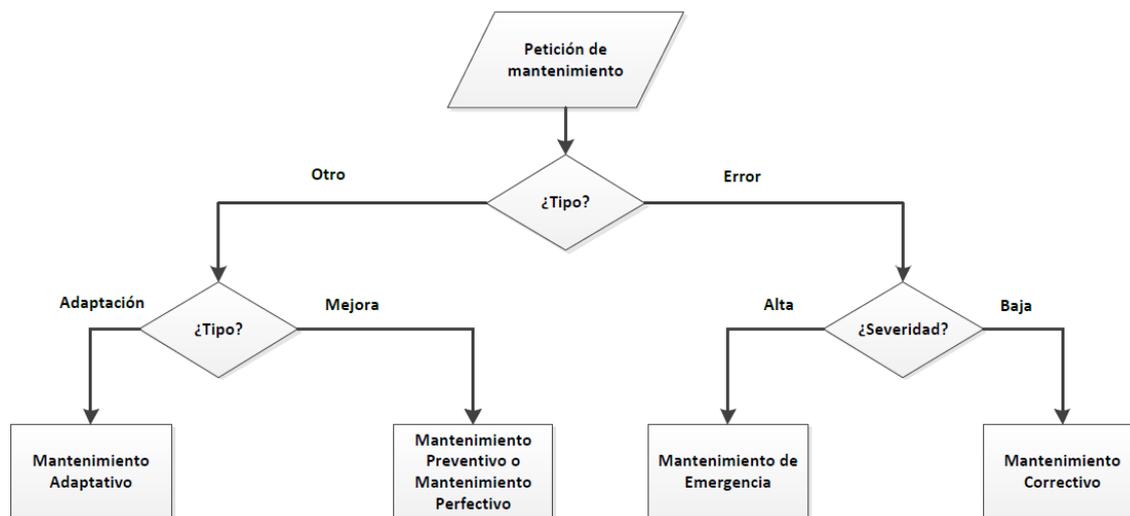


Figura 2.4 Clasificación de la solicitud de modificación por tipo de mantenimiento

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: R.S. Pressman, Ingeniería de Software, Séptima Ed. New York: Mac Graw Hill, 2010

### ACTIVIDAD 3. Análisis de la solicitud de modificación para determinar si se acepta o se rechaza.

Siga las siguientes tareas:

- **TAREA 3.1. Verificar y estudiar la solicitud de modificación:** Antes de iniciar el análisis de la solicitud de modificación, el Gestor de peticiones verificará que la información registrada en la PLANTILLA 2, mostrada en la Figura A.2 (Anexo A), es correcta.

Consideraciones para determinar su validez:

- ✓ Si se trata de un mantenimiento correctivo o de emergencia, deberá reproducir el problema.
- ✓ Si se trata de un mantenimiento perfecto, preventivo o adaptativo, comprobará que la petición es razonable o factible.

Una vez validada la petición se debe seguir el siguiente procedimiento:

- ✓ Si se trata de un mantenimiento correctivo, se evaluará su criticidad;

así, será posible determinar si la solución es a corto plazo, es decir, urgente, o a medio o largo plazo:

- Si el problema es crítico, su análisis y solución comienza inmediatamente. Sin embargo, este modo de actuación no eliminará la necesidad de una revisión posterior del problema para valorar los posibles efectos secundarios, establecer una solución definitiva y actualizar todos los productos implicados.
  - Si no es crítico, la petición continúa con el proceso normal.
- ✓ En el caso de un mantenimiento adaptativo, preventivo y perfectivo se delimitará su alcance, determinando si se trata de una modificación a los sistemas de información inicialmente afectados o de una incorporación para cubrir nuevas funcionalidades no contempladas hasta el momento en dichos sistemas de información.
- **TAREA 3.2. Determinar si se acepta o rechaza la petición de modificación:** A partir de la Solicitud de modificación registrada y clasificada, el Gestor de peticiones comprobará que el tipo de mantenimiento definido en la actividad anterior cubre el tipo de mantenimiento que requiere la Solicitud de modificación y, sobre la base de estos criterios, se aceptará o rechazará la petición.

Consideraciones para la toma de decisión:

- ✓ Lo más importante es saber que lo requerido se pueda hacer sin tener el más mínimo riesgo en la integridad del software.
- ✓ Es posible que se rechace porque la petición está mal redactada o porque el requerimiento o petición no sea necesario.
- ✓ Otras veces se rechaza porque la petición realizada costará demasiado.

El Gestor de peticiones registrará el resultado del análisis en la PLANTILLA 2, mostrada en la Figura A.2 (Anexo A).

Si la solicitud de modificación es aprobada, se asignará un Responsable de mantenimiento al cual se le notificarán los resultados obtenidos del análisis de la solicitud de modificación. El Responsable de mantenimiento asignado revisará la Solicitud de modificación; si está de acuerdo con el análisis realizado, aceptará preparar el proceso de mantenimiento y firmará la Solicitud de modificación presentada para indicar que fue aprobada y procederá a implementar los cambios solicitados.

Caso contrario, el Gestor de peticiones notificará al Solicitante el motivo de rechazo y finalizará el análisis de la petición.

#### **ACTIVIDAD 4.- Priorizar la modificación**

Siga las siguientes tareas:

- **TAREA 4.1 Asignar prioridad a la petición:** Si la Solicitud de modificación está aprobada, el Gestor de peticiones asignará la prioridad a la petición de modificación. Conociendo el tipo de mantenimiento y la necesidad del solicitante se asignará una prioridad para la planeación y la forma de actuación. Se tendrán las siguientes opciones para asignar la prioridad a la solicitud de modificación:
  - ✓ Alta: Se escoge esta opción cuando la modificación no es posible planearse y debe realizarse de inmediato.
  - ✓ Media: Se escoge esta opción cuando se debe realizar de inmediato pero se puede planear.
  - ✓ Baja: Se escoge esta opción cuando no se realiza de inmediato y se planea por completo.

En la PLANTILLA 2, mostrada en la Figura A.2 (Anexo A), se registrará la prioridad asignada a la solicitud de modificación presentada.

#### **ENTRADAS:**

- Solicitud de modificación inicial (PLANTILLA 1).

**SALIDAS:**

Solicitud de modificación clasificada y validada (aprobada o rechazada)

(PLANTILLA 1).

- Registro del estudio de la solicitud de modificación (PLANTILLA 2).

**RESUMEN:**

En la Tabla 2.1 se presenta el resumen de las actividades y tareas de la Fase 1.

<b>FASE 1: IDENTIFICACIÓN, CLASIFICACIÓN Y PRIORIZACIÓN DEL PROBLEMA (COMÚN PARA TODOS LOS TIPOS DE MANTENIMIENTO)</b>					
<b>ACTIVIDAD</b>	<b>TAREA</b>	<b>ENTRADAS</b>	<b>SALIDAS</b>	<b>RESPONSABLES</b>	<b>INTERFACES CON OTROS PROCESOS</b>
<b>1. Registro de la solicitud de modificación.</b>	1.1 Recibir la solicitud de modificación	-Solicitud de modificación (PLANTILLA 1)	-Solicitud de modificación recibida y registrada	-Solicitante -Gestor de peticiones	Aseguramiento de la Calidad
	1.2 Asignar un número de identificación	-Solicitud de modificación registrada	-Solicitud de modificación con número de identificación	-Gestor de peticiones	
<b>2. Clasificación de la solicitud de modificación por tipo de mantenimiento</b>	2.1 Seleccionar el tipo de mantenimiento	-Solicitud de modificación con número de identificación (PLANTILLA 1)	-Solicitud de modificación clasificada por el tipo de mantenimiento -Registro del estudio de la solicitud de modificación (PLANTILLA 2)	-Gestor de peticiones	
<b>3. Análisis de la solicitud de modificación para determinar si se acepta o se rechaza.</b>	3.1 Verificar y estudiar la solicitud de modificación	-Solicitud de modificación clasificada por tipo de mantenimiento	-Registro del estudio de la solicitud de modificación (PLANTILLA 2) -Solicitud de modificación aprobada o rechazada (PLANTILLA 1)	-Solicitante -Gestor de peticiones -Responsable de mantenimiento	
	3.2 Determinar si se acepta o rechaza la petición de modificación				
<b>4.-Priorizar la modificación</b>	4.1 Asignar prioridad a la petición	-Solicitud de modificación aceptada	-Solicitud de modificación priorizada -Registro del estudio de la solicitud de modificación (PLANTILLA 2)	-Gestor de peticiones	

**Tabla 2.1 Resumen de las actividades y tareas de la fase 1.**

**Elaborada por:** Balseca Marcela, Caiza Gabriela

**Fuente:** Estándar IEEE 1219 de Mantenimiento de Software (1999), Metodología MANTEMA para mantenimiento de software (2000), Metodología METRICA Versión 3 (2000)

## 2. ACTIVIDADES Y TAREAS PARA UN MANTENIMIENTO DE EMERGENCIA

Una vez finalizada la fase 1, para continuar con el proceso de mantenimiento se deberán revisar los resultados obtenidos del análisis de la Solicitud de modificación registrados en la PLANTILLA 1 y PLANTILLA 2.

Los requisitos para aplicar estas actividades y tareas son:

- Solicitud de modificación clasificada por el tipo de mantenimiento de emergencia.
- Solicitud de modificación aprobada.

### 2.1. Fase 2: Análisis

En esta fase se debe diagnosticar y analizar el cambio para dar respuesta a la solicitud de mantenimiento que ha sido aceptada en la fase anterior, siguiendo las siguientes actividades:

#### **ACTIVIDAD 1. Asignación y análisis de recursos**

Siga las siguientes tareas:

- **TAREA 1.1. Asignar recursos:** El Responsable de mantenimiento seleccionará dos o más personas para formar parte del equipo, dependiendo de la complejidad del mantenimiento. En la PLANTILLA 4, mostrada en la Figura A.4 (Anexo A), se registrará el recurso seleccionado para formar parte del Equipo de mantenimiento.

El Responsable de mantenimiento deberá tener ciertos criterios para conformar su equipo:

- ✓ Disponibilidad: deberá tener en cuenta que personas de la empresa están disponibles o pueden ser retiradas temporalmente de otras actividades laborales para integrar el grupo.

- ✓ Conocimiento: deberá tener en cuenta este factor ya que muchas veces el conocimiento puede beneficiar de varias maneras al mantenimiento, debido a que la mayoría de los nuevos requerimientos son de tipo correctivo y tienen un alto nivel de complejidad al ser solucionados.

## **ACTIVIDAD 2. Análisis del error.**

Siga las siguientes tareas:

- **TAREA 2.1. Investigar y analizar causas:** El Equipo de mantenimiento analizará la Solicitud de modificación registrada en la PLANTILLA 1, verificará el problema con la colaboración del Solicitante, lo reproducirá y estudiará diferentes alternativas para implementar la modificación; los resultados obtenidos los registrará en la PLANTILLA 6, mostrada en la Figura A.6 (Anexo A).
- **TAREA 2.2. Identificar los elementos a modificar:** El Equipo de mantenimiento identificará las partes del sistema que se van a modificar y la capa de la arquitectura a la que pertenecen. En la PLANTILLA 12, mostrada en la Figura A.12 (Anexo A), registrará la información recopilada y la descripción del cambio a implementar.

### **ENTRADAS:**

- Solicitud de modificación validada, clasificada y priorizada (PLANTILLA 1).
- Documentación del proyecto o sistema.
- Sistema existente.

### **SALIDAS:**

- Registro de asignación de recursos (PLANTILLA 4).
- Diagnóstico del error y posibles soluciones (PLANTILLA 6).
- Listado de elementos a modificar preliminar (PLANTILLA 12).

## 2.2. Fase 3: Diseño

En esta fase se deben identificar cada uno de los elementos afectados por el cambio, siguiendo las siguientes actividades:

### **ACTIVIDAD 1. Estudio de los elementos a modificar**

Siga las siguientes tareas:

- **TAREA 1.1. Identificar los elementos afectados:** En esta tarea, el Equipo de mantenimiento junto con el Responsable de mantenimiento identificarán de forma detallada cada uno de los elementos afectados por el cambio, dejando claramente definido qué parte del sistema se modificará, quién realizará la modificación y qué tiempo se requiere para realizar la modificación: toda esta información se registrará en la PLANTILLA 12, mostrada en la Figura A.12 (Anexo A).

### **ACTIVIDAD 2. Diseño de las pruebas**

Siga las siguientes tareas:

- **TAREA 2.1. Crear los casos de pruebas:** En esta tarea, el Equipo de mantenimiento deberá crear los casos de prueba necesarios para ejecutar las pruebas de funcionalidad. Para realizar los casos de prueba se utilizará la PLANTILLA 16, mostrada en la Figura A.16 (Anexo A).

### **ENTRADAS:**

- Documentación del proyecto o sistema.
- Código fuente.
- Base de datos.
- Salidas de la fase de análisis.

### **SALIDAS:**

- Lista de elementos a modificar actualizada y revisada (PLANTILLA 12).
- Casos de prueba (PLANTILLA 16).

### 2.3. Fase 4: Implementación

En esta fase se deben implementar los cambios al código fuente lo más pronto posible, siguiendo la siguiente actividad:

#### **ACTIVIDAD 1. Intervención correctiva urgente.**

Siga las siguientes tareas:

- **TAREA 1.1. Realizar acciones correctivas:** En esta tarea, el Equipo de mantenimiento procederá a realizar las correcciones necesarias e indispensables para que el problema detectado sea corregido de manera inmediata.
- **TAREA 1.2. Documentar los cambios:** El Equipo de mantenimiento deberá registrar los cambios realizados en la PLANTILLA 7, mostrada en la Figura A.7 (Anexo A).

#### **ENTRADAS:**

- Código fuente.
- Documentación del proyecto o sistema.
- Salidas de la fase de diseño.

#### **SALIDAS:**

- Software actualizado.
- Acciones correctivas urgentes realizadas (PLANTILLA 7).

### 2.4. Fase 5: Pruebas del sistema

En esta fase se deben realizar las pruebas del sistema sobre el sistema modificado, siguiendo las siguientes actividades:

#### **ACTIVIDAD 1. Ejecución y control de pruebas**

Siga las siguientes tareas:

- **TAREA 1.1. Ejecutar pruebas:** En esta tarea, se realizarán las pruebas de funcionalidad sobre el sistema software modificado. Los responsables encargados de las pruebas ejecutarán los casos de prueba preparados en la fase anterior.
- **TAREA 1.2. Registrar los resultados de las pruebas:** Los responsables que ejecutaron las pruebas, reportarán los resultados de las pruebas ejecutadas en la PLANTILLA 16, mostrada en la Figura A.16 (Anexo A).
- **TAREA 1.3. Preparar las pruebas de aceptación:** El Equipo de mantenimiento deberá preparar las pruebas de aceptación que serán ejecutadas en la siguiente fase. Para ello deberán crear los casos de prueba necesarios para ejecutar las pruebas de aceptación. Para realizar los casos de prueba se utilizará la PLANTILLA 16, mostrada en la Figura A.16 (Anexo A).

**ENTRADAS:**

- Sistema actualizado.
- Documentación del proyecto o sistema.
- Salidas de la fase de implementación.

**SALIDAS:**

- Sistema probado.
- Resultados de los Casos de prueba (PLANTILLA 16).
- Casos de prueba de aceptación (PLANTILLA 16).

**2.5. Fase 6: Pruebas de aceptación**

En esta fase se deben realizar las pruebas de aceptación sobre el sistema modificado, siguiendo las siguientes actividades:

**ACTIVIDAD 1. Ejecución y control de pruebas**

Siga las siguientes tareas:

- **TAREA 1.1. Ejecutar las pruebas de aceptación:** En esta tarea, se realizarán las pruebas de aceptación sobre el sistema software modificado. Los responsables asignados para realizar las pruebas de aceptación ejecutarán los casos de prueba preparados en la fase anterior.
- **TAREA 1.2. Registrar los resultados de las pruebas:** Los responsables que ejecutaron las pruebas, reportarán los resultados de las pruebas ejecutadas en la PLANTILLA 16, mostrada en la Figura A.16 (Anexo A).

**ENTRADAS:**

- Sistema totalmente integrado.
- Documentación del proyecto o sistema.
- Salidas de la fase de pruebas del sistema.

**SALIDAS:**

- Sistema integrado y probado.
- Resultados de los Casos de prueba de aceptación (PLANTILLA 16).

**2.6. Fase 7: Puesta en producción**

En esta fase se debe poner nuevamente en producción el sistema en un entorno real, siguiendo las siguientes actividades:

**ACTIVIDAD 1. Entrega del producto final**

Siga las siguientes tareas:

- **Tarea 1.1. Realizar copia del producto software:** En esta tarea, el Equipo de mantenimiento realizará una copia del producto software modificado y de todos los documentos del sistema.
- **Tarea 1.2. Pasar a producción:** El Equipo de mantenimiento pasará al entorno de producción el software corregido para ser utilizado por los usuarios finales.

**ENTRADAS:**

- Sistema probado y aceptado
- Documentación del proyecto o sistema.
- Salidas de la fase de pruebas de aceptación.

**SALIDAS:**

- Sistema en producción.

**RESUMEN:**

En la Tabla 2.2 se presenta el resumen de las actividades y tareas para un mantenimiento de Emergencia:

ACTIVIDADES Y TAREAS PARA UN MANTENIMIENTO DE EMERGENCIA						
FASE	ACTIVIDAD	TAREA	ENTRADAS	SALIDAS	RESPONSABLES	INTERFACES CON OTROS PROCESOS
FASE 2: ANÁLISIS	<b>1. Asignación y análisis de recursos</b>	1.1. Asignar recursos	- Solicitud de modificación validada, clasificada y priorizada (PLANTILLA 1) - Documentación del proyecto o sistema - Sistema existente	- Registro de asignación de recursos (PLANTILLA 4) - Diagnóstico del error y posibles soluciones (PLANTILLA 6) - Lista de elementos a modificar preliminar (PLANTILLA 12)	- Responsable de mantenimiento	Aseguramiento de la Calidad
	<b>2. Análisis del error</b>	2.1. Investigar y analizar causas			- Equipo de mantenimiento	
		2.2. Identificar los elementos a modificar				
FASE 3: DISEÑO	<b>1. Estudio de los elementos a modificar</b>	1.1. Identificar los elementos afectados	- Documentación del proyecto o sistema - Código fuente - Base de datos - Salidas de la fase de análisis	- Lista de elementos a modificar actualizada y revisada (PLANTILLA 12) - Casos de prueba (PLANTILLA 16)	- Responsable de mantenimiento - Equipo de mantenimiento	Aseguramiento de la Calidad
	<b>2. Diseño de las pruebas</b>	2.1. Crear los casos de prueba			- Equipo de mantenimiento	
FASE 4: IMPLEMENTACIÓN	<b>1. Intervención correctiva urgente</b>	1.1. Realizar acciones correctivas	- Código fuente - Documentación del proyecto o sistema - Salidas de la fase de diseño	- Software actualizado - Acciones correctivas urgentes realizadas (PLANTILLA 7)	- Equipo de mantenimiento	Aseguramiento de la Calidad
		1.2. Documentar los cambios				Gestión de la Configuración

<b>FASE 5: PRUEBAS DEL SISTEMA</b>	<b>1. Ejecución y control de pruebas</b>	1.1. Ejecutar pruebas unitarias	-Sistema actualizado -Documentación del proyecto o sistema -Salidas de la fase de implementación	-Sistema probado. -Resultados de los Casos de prueba (PLANTILLA 16) -Casos de prueba de aceptación (PLANTILLA 16)	-Equipo de mantenimiento (Responsables encargados de las pruebas)	Aseguramiento de Calidad
		1.2. Registrar los resultados de las pruebas				
		1.3. Preparar las pruebas de aceptación				
<b>FASE 6: PRUEBAS DE ACEPTACIÓN</b>	<b>1. Ejecución y control de pruebas</b>	1.1. Ejecutar las pruebas de aceptación	-Sistema totalmente integrado -Documentación del proyecto o sistema -Salidas de la fase de pruebas del sistema	-Sistema integrado y probado. -Resultados de los Casos de pruebas de aceptación (PLANTILLAS 16)	-Responsables asignados para las pruebas de aceptación	Aseguramiento de la Calidad
		1.2. Registrar los resultados de las pruebas				
<b>FASE 7: LIBERACIÓN</b>	<b>1. Entrega del producto final</b>	1.1. Realizar copia del producto software	-Sistema probado y aceptado -Documentación del proyecto o sistema -Salidas de la fase de pruebas de aceptación	-Sistema en producción	-Equipo de mantenimiento	Aseguramiento de la Calidad
		1.2. Pasar a producción				Gestión de la Configuración

**Tabla 2.2 Resumen de las actividades y tareas para un Mantenimiento de Emergencia.**

**Elaborada por:** Balseca Marcela, Caiza Gabriela

**Fuente:** Estándar IEEE 1219 de Mantenimiento de Software (1999), Metodología MANTEMA para mantenimiento de software (2000), Metodología METRICA Versión 3 (2000)

### **3. ACTIVIDADES Y TAREAS PARA UN MANTENIMIENTO CORRECTIVO, PERFECTIVO, PREVENTIVO Y ADAPTATIVO**

Una vez finalizada la fase 1, para continuar con el proceso de mantenimiento se deberán revisar los resultados obtenidos del análisis de la Solicitud de modificación, registrados en la PLANTILLA 1 y PLANTILLA 2.

Los requisitos para aplicar estas actividades y tareas son:

- Solicitud de modificación clasificada por el tipo de mantenimiento correctivo, perfectivo, preventivo o adaptativo.
- Solicitud de modificación aprobada.

#### **3.1. Fase 2: Análisis**

En esta fase se debe diagnosticar y analizar el cambio para dar respuesta a la solicitud de mantenimiento que ha sido aceptada en la fase anterior, siguiendo las siguientes actividades:

##### **ACTIVIDAD 1. Recolectar la documentación del sistema**

Esta actividad consta de las siguientes tareas:

- **TAREA 1.1. Recopilar toda la documentación del sistema a modificar:** El Responsable de mantenimiento deberá obtener la última versión de la documentación del proyecto o sistema a modificarse para facilitar el mantenimiento, ahorrar tiempo y dinero. Toda la documentación disponible del sistema se deberá tener lista y a la mano a la hora de empezar el proceso de mantenimiento.
- **TAREA 1.2. Realizar un inventario de la documentación recopilada:** El Responsable de mantenimiento deberá realizar un inventario de toda la documentación del sistema recopilada, para ello utilizará la PLANTILLA 3, mostrada en la Figura A.3 (Anexo A).

## **ACTIVIDAD 2. Asignación y análisis de recursos**

Siga las siguientes tareas:

- **TAREA 2.1. Asignar recursos:** El Responsable de mantenimiento seleccionará dos o más personas para formar parte del equipo, dependiendo de la complejidad del mantenimiento. En la PLANTILLA 4, mostrada en la Figura A.4 (Anexo A), se registrará el recurso seleccionado para formar parte del Equipo de mantenimiento.

El Responsable de mantenimiento deberá tener ciertos criterios para conformar su equipo:

- ✓ Disponibilidad: deberá tener en cuenta que personas de la empresa están disponibles o pueden ser retiradas temporalmente de otras actividades laborales para integrar el grupo.
- ✓ Conocimiento: deberá tener en cuenta este factor ya que muchas veces el conocimiento puede beneficiar de varias maneras al mantenimiento, debido a que la mayoría de los nuevos requerimientos son de tipo correctivo y tienen un alto nivel de complejidad al ser solucionados.

## **ACTIVIDAD 3. Análisis de la petición.**

Siga las siguientes tareas:

- **TAREA 3.1. Definir impacto de la modificación:** El Equipo de mantenimiento determinará el impacto que va a producir el cambio en el sistema, la base de datos y la documentación.

Además se deberá analizar ¿En qué forma va a cambiar el software después de introducirle el cambio? El impacto se realizará por petición o requerimiento. En la PLANTILLA 5, mostrada en la Figura A.5 (Anexo A), se registrará el elemento a modificar y la descripción del cambio que se va realizar al elemento registrado.

- **TAREA 3.2. Plantear posibles soluciones:** A partir del producto software en explotación y de la Solicitud de modificación presentada, el Equipo de mantenimiento realizará un diagnóstico del error y planteará posibles soluciones para dar una respuesta satisfactoria a los problemas detectados. En la PLANTILLA 6, mostrada en la Figura A.6 (Anexo A), se registrará el diagnóstico del error y las posibles soluciones planteadas.

**Nota:** Esta tarea solo aplicará para el mantenimiento correctivo.

- **TAREA 3.3. Elegir alternativa adecuada:** El Equipo de mantenimiento junto con el Responsable de mantenimiento seleccionarán la alternativa más adecuada para implementar la solución, el resultado lo registrarán en la PLANTILLA 6, mostrada en la Figura A.6 (Anexo A).

**Nota:** Esta tarea solo aplicará para el mantenimiento correctivo.

- **TAREA 3.4. Documentar posibles soluciones:** A partir del producto software en explotación y de la Solicitud de modificación presentada, el Equipo de mantenimiento realizará un listado de los elementos software y propiedades mejorables, para ello se utilizará la PLANTILLA 8, mostrada en la Figura A.8 (Anexo A).

**Nota:** Esta tarea solo aplicará para el mantenimiento preventivo.

- **TAREA 3.5. Estudiar la solicitud de modificación:** A partir del producto software en explotación y de la solicitud de modificación, el Equipo de mantenimiento realizará un diagnóstico de la adaptación para identificar qué elementos se van a modificar, para ello utilizará la PLANTILLA 9, mostrada en la Figura A.9 (Anexo A).

**Nota:** Esta tarea solo aplicará para el mantenimiento adaptativo.

#### **ACTIVIDAD 4. Valoración de la modificación**

Siga las siguientes tareas:

- **TAREA 4.1. Valorar la petición:** En esta tarea, el Equipo de mantenimiento junto con el Responsable de mantenimiento valorarán los costos y esfuerzos necesarios para atender la petición. La estimación de costo y esfuerzo se determinará realizando una matriz de rastreabilidad que lleve los requerimientos nuevos vs los requerimientos actuales.

Al tener esta matriz completa, sabiendo el conocimiento y experiencia de cada desarrollador y del líder del proyecto, se determinarán los tiempos estimados para cada cambio; con estos tiempos y lo que cuesta cada hora de trabajo de todos los involucrados, se calculará el costo estimado del mantenimiento, para lo cual:

- ✓ El Responsable de mantenimiento deberá llenar la matriz de rastreabilidad con cada nuevo requerimiento y los requerimientos actuales del sistema a modificar, llenando la PLANTILLA 10, mostrada en la Figura A.10 (Anexo A).
- ✓ Una vez que esté completa la matriz, el Equipo de mantenimiento estimará el tiempo y esfuerzo necesario para cada requerimiento nuevo, para ello utilizará la PLANTILLA 11, mostrada en la Figura A.11 (Anexo A).
- ✓ Por último, en la estimación de costo y esfuerzo se deberá calcular el valor unitario y el valor total del mantenimiento, estos valores se registrarán en la PLANTILLA 11, mostrada en la Figura A.11 (Anexo A).

#### **ACTIVIDAD 5. Análisis detallado**

Siga las siguientes tareas:

- **TAREA 5.1. Identificar los elementos a modificar:** El Equipo de mantenimiento identificará las partes del sistema que se van a modificar y la capa de la arquitectura a la que pertenecen. En la PLANTILLA 12, mostrada en la Figura A.12 (Anexo A), se registrará la información recopilada y la descripción del cambio a implementar.
- **TAREA 5.2. Realizar el plan de pruebas:** El Equipo de mantenimiento deberá desarrollar el plan de pruebas. Se tendrán que definir los requisitos de pruebas para al menos tres tipos de pruebas: de funcionalidad, de rendimiento y de aceptación.

En el plan de pruebas se deberá especificar qué tipo de pruebas se van a ejecutar sobre el sistema modificado. Se indican algunos tipos de pruebas de software:

- ✓ Pruebas de funcionalidad
- ✓ Pruebas de las interfaces
- ✓ Pruebas de regresión
- ✓ Pruebas de aceptación
- ✓ Pruebas de interoperabilidad
- ✓ Pruebas de estrés
- ✓ Pruebas de carga
- ✓ Pruebas de estabilidad

Se deberá seleccionar el tipo de pruebas de acuerdo a como el cambio solicitado afecte al sistema; ya que las pruebas deberán servir para validar y verificar que el cambio solicitado se haya implementado correctamente. Además, las pruebas de rendimiento nos servirán para validar que el cambio realizado no haya afectado la funcionalidad del sistema en general.

Para realizar el plan de pruebas se utilizará la PLANTILLA 13, mostrada en la figura A.13 (Anexo A).

- **TAREA 5.3. Desarrollar un plan de implementación preliminar:** El Responsable de mantenimiento desarrollará un plan, en el cual se detallará cada una de las actividades que se llevarán a cabo para implementar el cambio. Se deberá establecer un plan de trabajo en el que se determinen los plazos estimados para su implementación con las fechas de comienzo y fin y la composición del equipo de trabajo necesario, teniendo en cuenta el alcance de la modificación y el nivel de esfuerzo requerido.

Para realizar el plan de implementación se utilizará la PLANTILLA 14, mostrada en la figura A.14 (Anexo A).

#### **ENTRADAS:**

- Solicitud de modificación validada, clasificada y priorizada (PLANTILLA 1).
- Documentación del proyecto o sistema.
- Sistema existente.

#### **SALIDAS:**

- Inventario de la documentación del sistema (PLANTILLA 3).
- Registro de asignación de recursos (PLANTILLA 4).
- Definición de impacto del cambio (PLANTILLA 5).
- Diagnóstico del error o cambio y posibles soluciones (PLANTILLA 6).
- Listado de elementos software y propiedades mejorables (PLANTILLA 8).
- Diagnóstico de la adaptación a ser implementada (PLANTILLA 9).
- Matriz de rastreabilidad (PLANTILLA 10).
- Estimación de costo y esfuerzo (PLANTILLA 11).
- Lista de elementos a modificar preliminar (PLANTILLA 12).
- Plan de pruebas preliminar (PLANTILLA 13).
- Plan de implementación preliminar (PLANTILLA 14).

### 3.2. Fase 3: Diseño

En esta fase se debe determinar qué parte del sistema software se ve afectado, dejando claramente definido y documentado qué componentes hay que modificar, siguiendo las siguientes actividades:

#### **ACTIVIDAD 1. Estudio de los elementos a modificar**

Siga las siguientes tareas:

- **TAREA 1.1. Identificar los elementos afectados:** En esta tarea, el Equipo de mantenimiento junto con el Responsable de mantenimiento identificarán de forma detallada cada uno de los elementos del sistema afectados por el cambio, dejando claramente definido qué parte del sistema se modificará, quién realizará la modificación y qué tiempo se requiere para realizar la modificación, toda esta información se registrará en la PLANTILLA 12, mostrada en la Figura A.12 (Anexo A).
- **TAREA 1.2. Identificar la documentación del sistema a modificar:** El Responsable de mantenimiento realizará un listado de la documentación que se debe actualizar por motivo del cambio para llevar un control de toda la documentación del sistema.  
En la PLANTILLA 15, mostrada en la Figura A.15 (Anexo A), se registrará el listado de documentos a modificar.

#### **ACTIVIDAD 2. Diseño de las pruebas.**

Siga las siguientes tareas:

- **TAREA 2.1. Crear los casos de prueba:** En esta tarea, el Equipo de mantenimiento deberá crear los casos de prueba necesarios de acuerdo a los niveles de pruebas establecidos para el sistema en la fase de análisis. Se tendrán que definir los casos de prueba para al menos tres niveles de pruebas: de funcionalidad, de rendimiento y de aceptación. Para realizar los casos de prueba se utilizará la PLANTILLA 16, mostrada en la Figura A.16 (Anexo A).

### **ACTIVIDAD 3. Estudio y revisión del plan de implementación**

Siga las siguientes tareas:

- **TAREA 3.1. Revisar los documentos actualizados:** El Responsable de mantenimiento revisará el Listado de elementos a modificar actualizado registrado en la PLANTILLA 12 y el Listado de documentos a modificar registrado en la PLANTILLA 15 para identificar los cambios realizados en estos documentos y en base a ello, se actualizará el plan de implementación.
- **TAREA 3.2 Actualizar el plan de implementación:** En esta tarea, el Responsable de mantenimiento actualizará el plan de implementación realizado en la PLANTILLA 14, en base a los resultados obtenidos en la tarea anterior. Es muy importante que este plan de trabajo se presente actualizado, ya que es la base para el inicio de la fase de implementación.

#### **ENTRADAS:**

- Documentación del proyecto o sistema.
- Código fuente.
- Base de datos.
- Salidas de la fase de análisis.

#### **SALIDAS:**

- Lista de elementos a modificar actualizada y revisada (PLANTILLA 12).
- Lista de documentos a modificar (PLANTILLA 15).
- Plan de pruebas revisado (PLANTILLA 13).
- Casos de prueba (PLANTILLA 16).
- Plan de implementación actualizado y revisado. (PLANTILLA 14).

### 3.3. Fase 4: Implementación

El objetivo de esta fase es implementar los cambios en el código fuente, ejecutar las pruebas y mantener actualizada toda la documentación del sistema.

#### **ACTIVIDAD 1. Intervención y revisión de pruebas.**

Esta actividad consta de las siguientes tareas:

- **TAREA 1.1. Ejecutar intervención:** El Equipo de mantenimiento debe ejecutar las acciones necesarias para implementar los cambios en el código fuente basándose en el Plan de implementación registrado en la PLANTILLA 14. Para implementar los cambios en el sistema se deberá revisar y actualizar el Listado de elementos a modificar registrado en la PLANTILLA 12. Además, deberá actualizar el Registro de la solicitud de modificación ingresado en la PLANTILLA 2.
- **TAREA 1.2. Revisar las pruebas preparadas:** Una vez implementadas todas las modificaciones, el Equipo de mantenimiento procederá a revisar los casos de prueba preparados para validar si están correctos y poder ejecutarlos en la siguiente fase.

#### **ACTIVIDAD 3. Actualización de la documentación del sistema**

Siga las siguientes tareas:

- **TAREA 3.1. Modificar la documentación de los elementos afectados:** En este punto el Equipo de mantenimiento deberá actualizar la documentación que ha sido afectada por el cambio, para ello se deberá revisar y actualizar el Listado de documentos a modificar registrado en la PLANTILLA 15.

#### **ENTRADAS:**

- Código fuente.
- Documentación del proyecto o sistema.
- Salidas de la fase de diseño.

**SALIDAS:**

- Software actualizado.
- Documentos de diseño actualizados.
- Documentos de pruebas actualizados.
- Documentos de usuario actualizados.
- Listado de elementos a modificar actualizado (PLANTILLA 12).
- Listado de documentos a modificar actualizado (PLANTILLA 15).
- Registro del estudio de la solicitud de modificación actualizado (PLANTILLA 2).

**3.4. Fase 5: Pruebas del sistema**

En esta fase se deben realizar las pruebas del sistema sobre el software modificado, siguiendo las siguientes actividades:

**ACTIVIDAD 1. Ejecución de pruebas**

Esta actividad consta de las siguientes tareas:

- **TAREA 1.1. Revisar pruebas preparadas:** En esta tarea, el Equipo de mantenimiento revisará el plan de pruebas y los casos de prueba preparados en la fase de diseño para saber qué tipo de pruebas se van a ejecutar sobre el sistema.
- **TAREA 1.2. Ejecutar pruebas del sistema:** En esta tarea, se realizarán las pruebas sobre el producto software modificado. Los responsables encargados de las pruebas ejecutarán los casos de prueba preparados para las pruebas del sistema.

**ACTIVIDAD 2. Control de pruebas**

Siga las siguientes tareas:

- **Tarea 2.1. Crear reportes de las pruebas realizadas:** Los responsables encargados de las pruebas deberán generar reportes de cada una de las

pruebas ejecutas, para ello utilizarán la PLANTILLA 17, mostrada en la Figura A.17 (Anexo A).

- **TAREA 2.2. Crear un repositorio para archivar los reportes de pruebas:** El Responsable de mantenimiento deberá crear un repositorio en el cual se archivarán todos los reportes de pruebas generados durante el proceso de mantenimiento.
- **TAREA 2.3. Registrar los resultados de las pruebas:** Los responsables que ejecutaron las pruebas, reportarán los resultados de las pruebas ejecutadas en la PLANTILLA 16, mostrada en la Figura A.16 (Anexo A).

#### **ENTRADAS:**

- Software actualizado.
- Documentación del proyecto o sistema actualizado.
- Salidas de la fase de implementación.

#### **SALIDAS:**

- Software probado.
- Reportes detallado de pruebas (PLANTILLAS 17).
- Resultados de los Casos de prueba (PLANTILLA 16).

### **3.5. Fase 6: Pruebas de aceptación**

En esta fase se deben realizar las pruebas de aceptación sobre el sistema modificado, siguiendo las siguientes actividades:

#### **ACTIVIDAD 1. Ejecución de pruebas**

Siga las siguientes tareas:

- **TAREA 1.1. Revisar pruebas preparadas:** En esta tarea, el Equipo de mantenimiento revisará el plan de pruebas y los casos de prueba preparados en la fase de diseño para saber que pruebas se van a ejecutar para la aceptación del sistema.

- **TAREA 1.2. Ejecutar pruebas de aceptación:** En esta tarea, se realizarán las pruebas sobre el producto software modificado. Los responsables asignados para realizar las pruebas de aceptación ejecutarán los casos de prueba preparados.

## **ACTIVIDAD 2. Control de pruebas**

Siga las siguientes tareas:

- **TAREA 2.1. Registrar los resultados de las pruebas:** Los responsables que ejecutaron las pruebas, reportarán los resultados de las pruebas ejecutadas en la PLANTILLA 16, mostrada en la Figura A.16 (Anexo A).
- **TAREA 2.2. Revisar las pruebas ejecutadas:** En esta tarea, se evaluará el resultado de las pruebas ejecutadas para que de esta manera se garantice la puesta en producción de la versión final. El solicitante deberá estar presente en dicha revisión junto con el Equipo de mantenimiento.

### **ENTRADAS:**

- Software totalmente integrado.
- Documentación del proyecto o sistema actualizado.
- Salidas de la fase de pruebas del sistema.

### **SALIDAS:**

- Software probado y aceptado.
- Resultados de los Casos de pruebas de aceptación (PLANTILLAS 16).

## **3.6. Fase 7: Puesta en producción**

En esta fase se debe poner en producción el sistema en un entorno real, siguiendo las siguientes actividades:

## ACTIVIDAD 1. Intervención

Siga las siguientes tareas:

- **TAREA 1.1. Notificar a la comunidad de usuarios:** En esta tarea, el Responsable de mantenimiento notificará a los usuarios la puesta en producción del sistema software modificado.
- **TAREA 1.2. Realizar copia del producto software:** En esta tarea el equipo de mantenimiento realizará una copia del producto software modificado.

## ACTIVIDAD 2. Cierre de intervención

Siga las siguientes tareas:

- **TAREA 2.1. Pasar a producción:** El Equipo de mantenimiento, pasará al entorno de producción el software modificado para ser utilizado por los usuarios finales.

Además, el Responsable de mantenimiento deberá actualizar el Registro de la solicitud de modificación ingresado en la PLANTILLA 2.

- **TAREA 2.2. Archivar documentación del producto software modificado:** En esta tarea, el Equipo de mantenimiento archivará toda la documentación del sistema actualizada y generada durante el proceso de mantenimiento del software.

## ENTRADAS:

- Software aceptado.
- Documentación del proyecto o sistema.
- Salidas de la fase de pruebas de aceptación.

**SALIDAS:**

- Software en producción.
- Documentación del proyecto o sistema totalmente actualizado.
- Registro del estudio de la solicitud de modificación actualizado (PLANTILLA 2).

**RESUMEN:**

En la Tabla 2.3 se presenta el resumen de las actividades y tareas para un mantenimiento correctivo, perfectivo, preventivo y adaptativo.

ACTIVIDADES Y TAREAS POR FASES PARA UN MANTENIMIENTO CORRECTIVO, PREVENTIVO, PERFECTIVO Y ADAPTATIVO						
FASE	ACTIVIDAD	TAREA	ENTRADAS	SALIDAS	RESPONSABLES	INTERFACES CON OTROS PROCESOS
FASE 2: ANÁLISIS	<b>1. Recolectar la documentación del sistema</b>	1.1. Recopilar toda la documentación del sistema a modificar	-Solicitud de modificación validada, clasificada y priorizada (PLANTILLA 1)  -Documentación del proyecto o sistema  -Sistema existente	-Inventario de la documentación del sistema (PLANTILLA 3) -Registro de asignación de recursos (PLANTILLA 4) -Definición de impacto del cambio (PLANTILLA 5) -Diagnóstico del error o cambio y posibles soluciones (PLANTILLA 6) -Listado de elementos software y propiedades mejorables (PLANTILLA 8) -Diagnóstico de la adaptación a ser implementada (PLANTILLA 9) -Matriz de rastreabilidad. (PLANTILLA 10) -Estimación de costo y esfuerzo (PLANTILLA 11) -Lista de elementos a modificar preliminar (PLANTILLA 12) -Plan de pruebas preliminar (PLANTILLA 13) -Plan de implementación	-Responsable de mantenimiento	Aseguramiento de la Calidad
		1.2. Realizar un inventario de la documentación recopilada				
	<b>2. Asignación y análisis de recursos</b>	2.1. Asignar recursos			-Responsable de mantenimiento	
	<b>3. Análisis de la petición</b>	3.1. Definir impacto de la documentación			-Equipo de mantenimiento	Aseguramiento de la Calidad
		3.2. Plantear posibles soluciones (Mantenimiento correctivo)				
		3.3. Elegir alternativa adecuada (Mantenimiento correctivo)				
		3.4. Documentar posibles soluciones (Mantenimiento preventivo)				
		3.5. Estudiar la solicitud de				

		modificación (Mantenimiento adaptativo)		preliminar (PLANTILLA 14)	-Responsable de mantenimiento	
	<b>4. Valoración de la modificación</b>	4.1. Valorar la petición			-Equipo de mantenimiento -Responsable de mantenimiento	
	<b>5. Análisis detallado</b>	5.1. Identificar los elementos a modificar			-Responsable de mantenimiento	Aseguramiento de la Calidad
		5.2. Realizar el plan de pruebas				
		5.3. Desarrollar un plan de implementación preliminar				
<b>FASE 3: DISEÑO</b>	<b>1. Estudio de los elementos a modificar</b>	1.1. Identificar elementos afectados	-Documentación del proyecto o sistema -Código fuente -Base de datos -Salidas de la fase de análisis	-Lista de elementos a modificar actualizada y revisada (PLANTILLA 12) -Lista de documentos a modificar (PLANTILLA 15) -Plan de pruebas revisado (PLANTILLA 13) -Casos de prueba (PLANTILLA 16) -Plan de implementación actualizado y revisado (PLANTILLA 14)	-Responsable de mantenimiento -Equipo de mantenimiento	Aseguramiento de la Calidad
		1.2. Identificar la documentación del sistema a modificar			-Responsable de mantenimiento	
	<b>2. Diseño de las pruebas</b>	2.1. Crear los casos de prueba			-Equipo de mantenimiento	
	<b>3. Estudio y revisión del plan de implementación</b>	3.1 Revisar los documentos actualizados			-Responsable de mantenimiento	
3.2. Actualizar el plan de implementación						

<b>FASE 4: IMPLEMENTACIÓN</b>	<b>1. Intervención y revisión de pruebas.</b>	1.1. Ejecutar intervención	-Código fuente -Documentación del proyecto o sistema -Salidas de la fase de diseño	-Software actualizado -Documentos de diseño actualizados -Documentos de usuario actualizados -Lista de documentos a modificar actualizada (PLANTILLA 12) -Listado de documentos a modificar actualizado (PLANTILLA 15) -Registro del estudio de la solicitud de modificación actualizado (PLANTILLA 2)	-Equipo de mantenimiento	Aseguramiento de la Calidad  Gestión de la Configuración
		1.2. Revisar las pruebas preparadas				
	<b>2. Actualización de la documentación del sistema</b>	2.1. Modificar la documentación de los elementos afectados				
<b>FASE 5: PRUEBAS DE SISTEMA</b>	<b>1. Ejecución de pruebas</b>	1.1. Revisar pruebas preparadas	-Software actualizado. -Documentación del proyecto o sistema actualizado -Salidas de la fase de implementación	-Software probado -Reportes detallados de pruebas (PLANTILLA 17) -Resultado de los Casos de prueba (PLANTILLA 16)	-Equipo de mantenimiento (Responsables encargados de las pruebas)	Aseguramiento de la Calidad
		1.2. Ejecutar pruebas del sistema				
	<b>2. Control de pruebas</b>	2.1. Crear reporte de las pruebas del sistema				
		2.2. Crear un repositorio para archivar los reportes de pruebas				
		2.3. Registrar los resultados de las pruebas				
			-Equipo de mantenimiento			
			-Equipo de mantenimiento (Responsables encargados de las pruebas)			

<b>FASE 6: PRUEBAS DE ACEPTACIÓN</b>	<b>1.Ejecución de pruebas</b>	1.1.Revisar pruebas preparadas	-Software totalmente integrado -Documentación del proyecto o sistema actualizado -Salidas de la fase de pruebas del sistema	-Software probado y aceptado -Resultados de los Casos de pruebas de aceptación (PLANTILLA 16)	-Responsables asignados para las pruebas de aceptación	Aseguramiento de la Calidad
		1.2.Ejecutar pruebas de aceptación				
	<b>2.Control de pruebas</b>	2.1.Registrar los resultados de las pruebas				
		2.2.Revisión de las pruebas ejecutadas				
<b>FASE 7: PUERTA EN PRODUCCIÓN</b>	<b>1.Intervención</b>	1.1.Notificar a la comunidad de usuarios	-Software aceptado -Documentación del proyecto o sistema -Salidas de la fase de pruebas de aceptación	-Software en producción -Documentación del proyecto o sistema totalmente actualizado -Registro del estudio de la solicitud de modificación actualizado (PLANTILLA 2)	-Equipo de mantenimiento	Aseguramiento de la Calidad Gestión de la Configuración
		1.3.Realizar copia del producto software				
	<b>2.Cierre de intervención</b>	2.1.Pasar a producción				
		2.2.Archivar datos del producto software				

Tabla 2.3 Resumen de las actividades y tareas para un Mantenimiento Correctivo, Perfectivo, Preventivo y Adaptativo.

Elaborada por: Balseca Marcela, Caiza Gabriela

Fuente: Estándar IEEE 1219 de Mantenimiento de Software (1999), Metodología MANTEMA para mantenimiento de software (2000), Metodología METRICA Versión 3 (2000)

## **4. FLUJOS DE APOYO AL PROCESO**

Se definen dos interfaces con las cuales el proceso de mantenimiento está involucrado. Estas interfaces son los procesos de Aseguramiento de la Calidad y Gestión de la Configuración, con los cuales el proceso de mantenimiento tiene relación y debe implementarse para tener un proceso controlado.

### **4.1. Aseguramiento de la Calidad**

El Responsable de mantenimiento intervendrá durante el proceso de mantenimiento, efectuando revisiones de seguimiento periódicas, más o menos frecuentes según los casos, para constatar que el mantenimiento establecido para responder a una solicitud de modificación se realice de forma correcta.

El proceso de Gestión de calidad es aquel que se encarga de garantizar y velar la calidad de los entregables, artefactos y productos del sistema producidos durante el proceso de mantenimiento.

Esta tarea de la interfaz de aseguramiento de calidad se aplica a todas las actividades del proceso de Mantenimiento.

Para realizar el aseguramiento de calidad, se utilizará la PLANTILLA 18, mostrada en la Figura A.18 (Anexo A), la cual contendrá una lista de todos los documentos que se van a generar durante el proceso de mantenimiento de software para ir controlando el desarrollo de estos documentos durante cada una de las fases del proceso y de esta manera poder llevar un proceso controlado y de calidad.

### **4.2. Gestión de la Configuración**

El objetivo de la interfaz de gestión de configuración con el proceso de Mantenimiento, es conservar la integridad del sistema de información cuando se producen cambios en el mismo, mediante la identificación y control de versiones.

Para realizar la gestión de la configuración se utilizará la PLANTILLA 19, mostrada en la Figura A.17 (Anexo A), en la cual se llevará un registro de las versiones que tiene el sistema a ser modificado.

Este registro constituye el nexo de unión entre la petición o peticiones de mantenimiento y los cambios que se van a realizar sobre el sistema. Recoge datos referentes a las versiones de los sistemas de información de los que se parte y cuáles van a ser las nuevas versiones generadas, así como las versiones de los productos concretos afectados por el cambio y cuál será la nueva versión de dichos productos.

Para establecer la nueva versión de la aplicación, se aplicará lo siguiente:

- a)** El primer dígito se incrementará en 1, cuando se realicen cambios significativos en la funcionalidad y en la estructura de la base de datos, un cambio de ambiente o un rediseño. La primera versión de una aplicación debe empezar siempre con 1; Ejemplo V1.0.

Cuando se incrementa el primer dígito de la versión, los siguientes dígitos deben encerrarse; ejemplo si se cambia de la versión V1.3 a la versión 2, el código de versión será V2.0.

- b)** El segundo dígito se incrementará en 1, cuando el producto de software no es objeto de cambios significativos en su estructura y sólo se le agrega o modifica la funcionalidad, el identificador de versión es de la forma X.Y, el cual se obtiene sumando "0.1" al número de la versión anterior, truncándose cualquier decimal; ejemplo si la versión anterior es la 2.1 la siguiente versión debe ser 2.2.

Además, se utilizará la PLANTILLA 20, mostrada en la Figura A.20 (Anexo A), la cual contendrá un listado de todos los documentos del sistema o proyecto, para tener un registro y control de los documentos que se modificarán durante el proceso de mantenimiento de software con el fin de controlar la versión y actualización de estos documentos durante cada una de las fases del proceso.

## **CAPÍTULO 3**

# **APLICACIÓN DE LA GUÍA DE PROCEDIMIENTOS DE MANTENIMIENTO DE SOFTWARE A UN CASO DE ESTUDIO**

### **3.1 DESCRIPCIÓN DEL CASO DE ESTUDIO**

La guía se aplicará para dar mantenimiento a un sistema que se encarga del registro de información de medidas y generación de métricas de proyectos, conocido como LogiHistoricos el cual es utilizado por la empresa LogiCiel Cía. Ltda.

La aplicación permite el registro de información de medidas como: tamaño, esfuerzo, número de errores, costo y defectos de un proyecto durante su desarrollo; con esta información se realiza el cálculo de las métricas correspondientes.

Las medidas se registran por tareas y responsables, esta información se toma del cronograma del proyecto que se maneja desde Project Server 2010. Una vez que cada fase se finaliza, se procede con el cálculo de las métricas.

LogiHistoricos permite almacenar información histórica de los proyectos desarrollados en la empresa, lo que ayuda con la estimación de nuevos proyectos de similares características.

Debido a que se van a implementar nuevos cambios en el sistema LogiHistoricos, se vio la necesidad de aplicar la guía de procedimientos de mantenimiento de software para poder tener un control de cada uno de los cambios a implementarse, así como también tener actualizada la documentación del sistema. Contar con procedimientos bien definidos para realizar el mantenimiento del sistema, ayudará a mantener la calidad del software.

## 3.2 APLICACIÓN DE LA GUÍA



# **LOGICIEL SOLUCIONES INFORMÁTICAS**

---

---

## **GUÍA DE PROCEDIMIENTOS PARA EL PROCESO DE MANTENIMIENTO DE SOFTWARE**

**ELABORADO POR:  
MARCELA BALSECA  
GABRIELA CAIZA**

**FECHA:  
02 / 06 / 2014**



## **I. DESCRIPCIÓN DEL SISTEMA**

El sistema al cual se va a dar mantenimiento se denomina LogiHistoricos, que permite almacenar información histórica de los proyectos desarrollados en la empresa, lo que ayuda con la estimación de nuevos proyectos de similares características.

Además, el sistema permite el registro de información de medidas como: tamaño, esfuerzo, número de errores, costo y defectos de un proyecto durante su desarrollo; con esta información se realizara el cálculo de las métricas correspondientes.

Las medidas se registrarán por tareas y responsables, esta información se tomara del cronograma del proyecto que será manejado desde Project Server 2010. Una vez que cada fase sea finalizada, se procederá con el cálculo de las métricas.

## II. PROCEDIMIENTOS

### FASE 1: Identificación, clasificación y priorización del problema

1. El Solicitante de mantenimiento se encarga de llenar la Solicitud de Modificación inicial mostrada en la Figura 3.1.

#### SOLICITUD DE MODIFICACIÓN

<i>Fecha</i>	02/06/2014	<i>Id. Solicitud</i>	
<i>Tipo de solicitud</i>	Nuevos requerimientos	<i>Severidad de la Solicitud</i>	
<i>Nombre del proyecto</i>	LogiHistoricos		
<i>Líder del proyecto</i>	Isabel Vasco		
<i>Versión del proyecto</i>	1.0		
<i>Descripción de la solicitud</i>	<p>Generar reportes para la gerencia y exportarlos a Excel de las siguientes métricas :</p> <ul style="list-style-type: none"> <li>-Métricas por proyecto</li> <li>-Métricas por responsable</li> <li>-Métricas estándar</li> </ul> <p>Tener en cuenta lo siguiente:  Las métricas por proyecto y las métricas estándar ya están calculadas.  Se tienen que calcular las métricas por responsable.  Incluir en los reportes el valor de la métrica, el indicador y las observaciones.</p>		
<i>Solicitado por</i>	Líder del proyecto		
<b>REQUERIMIENTOS</b>			
<i>N°</i>	<i>Descripción</i>	<i>Tipo de Mantenimiento</i>	<i>Prioridad</i>
<b>RESULTADO DE LA SOLICITUD</b>			
<i>Estado de la solicitud</i>			
<i>Autorización</i>	Revisado por  Gestor de peticiones	Aceptado por  Responsable de Mantenimiento del Software	

Figura 3.1 Solicitud de modificación inicial del caso de estudio (PLANTILLA 1).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.

2. El Gestor de peticiones se encarga de llenar el Registro del estudio de la solicitud de modificación mostrado en la Figura 3.2.

LogiHistoricos											
DATOS GENERALES DEL PROYECTO O SISTEMA											
Nombre del Proyecto	LogiHistoricos										
Líder del Proyecto	Isabel Vasco										
Gestor de Peticiones	Marcela Balseca										
REGISTRO DEL ESTUDIO DE LA SOLICITUD DE MODIFICACIÓN											
ID. SOLICITUD	DESCRIPCIÓN O DETALLE	RESPONSABLE DEL MANTENIMIENTO	FECHA RECEPCIÓN	FECHA APROBACIÓN	FECHA CIERRE	TIPO DE MANTENIMIENTO	MOTIVO DE SELECCIÓN	PRIORIDAD	ESTADO DEL CAMBIO	MOTIVO DEL RECHAZO	ESTADO DE IMPLEMENTACIÓN
SM01	Generar un reporte de las métricas por proyecto y exportarlo a Excel.	Gabriela Caiza	02/06/2014	03/06/2014		Perfectivo	Seleccionamos este tipo de mantenimiento porque se va añadir nuevos requerimientos al sistema existente.	Medio	Aprobado		No iniciado
	Calcular y generar un reporte de las métricas por responsable y exportarlo a Excel.										
	Generar un reporte de las métricas estándar y exportarlo a Excel.										

Figura 3.2 Registro del estudio de la solicitud de modificación inicial (PLANTILLA 2).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.

3. EL Gestor de peticiones junto con el Responsable de mantenimiento actualizan la Solicitud de modificación clasificada y validada mostrada en la Figura 3.3.

### SOLICITUD DE MODIFICACIÓN

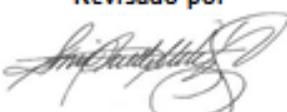
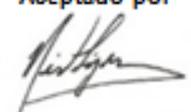
<b>Fecha</b>	02/06/2014	<b>Id. Solicitud</b>	SM01
<b>Tipo de solicitud</b>	Nuevos requerimientos	<b>Severidad de la Solicitud</b>	Media
<b>Nombre del proyecto</b>	LogiHistoricos		
<b>Líder del proyecto</b>	Isabel Vasco		
<b>Versión del proyecto</b>	1.0		
<b>Descripción de la solicitud</b>	<p>Generar reportes para la gerencia y exportarlos a Excel de las siguientes métricas :</p> <ul style="list-style-type: none"> <li>-Métricas por proyecto</li> <li>-Métricas por responsable</li> <li>-Métricas estándar</li> </ul> <p>Tener en cuenta lo siguiente:  Las métricas por proyecto y las métricas estándar ya están calculadas.  Se tienen que calcular las métricas por responsable.  Incluir en los reportes el valor de la métrica, el indicador y las observaciones.</p>		
<b>Solicitado por</b>	Lider del proyecto		
<b>REQUERIMIENTOS</b>			
<b>N°</b>	<b>Descripción</b>	<b>Tipo de Mantenimiento</b>	<b>Prioridad</b>
R1	Generar un reporte de las métricas por proyecto y exportarlo a Excel.	Perfectivo	Media
R2	Calcular y generar un reporte de las métricas por responsable y exportarlo a Excel.		
R3	Generar un reporte de las métricas estándar y exportarlo a Excel.		
<b>RESULTADO DE LA SOLICITUD</b>			
<b>Estado de la solicitud</b>	APROBADA 03/06/2014		
<b>Autorización</b>	<p style="text-align: center;"><b>Revisado por</b></p>  <p style="text-align: center;">Marcela Balseca Gestor de peticiones 02/06/2014</p>	<p style="text-align: center;"><b>Aceptado por</b></p>  <p style="text-align: center;">Gabriela Caiza Responsable de Mantenimiento del Software 03/06/2014</p>	

Figura 3.3 Solicitud de modificación clasificada y validada (PLANTILLA 1).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.

## FASE 2: Análisis

4. El Responsable de mantenimiento se encarga de realizar el Inventario de la documentación del sistema mostrado en la Figura 3.4.

					
DATOS GENERALES DEL PROYECTO O SISTEMA					
Nombre del Proyecto	LogiHistoricos				
Líder del Proyecto	Isabel Vasco				
Responsable del Mantenimiento	Gabriela Caiza				
INVENTARIO DE LA DOCUMENTACIÓN DEL SISTEMA O PROYECTO					
FASES DE DESARROLLO DEL SISTEMA	No.	ID - DOCUMENTO	NOMBRE DEL DOCUMENTO	VERSIÓN	OBSERVACIONES
Gestión de Requerimientos	1	LHS_ESP_FUN	Especificación de requerimientos	2,0	
	2	LHS_FOR_GRQ	Gestión de requerimientos	2,0	
Planeación del Proyecto	3	LHS_FOR_ESC	Listado de elementos de la configuración	2,0	
	4	LHS_PLN_PRY	Plan de gestión de la configuración	2,0	
	5	LHS_DOC_RCO	Riesgos y contingencias del proyecto	2,0	
	6	LHS_PLN_ADM	Plan de administración del proyecto	3,0	
Análisis y Diseño	7	LHS_DGA_ERC	Modelo conceptual	12,0	
	8	LHS_DGA_ERF	Modelo físico	12,0	
	9	LHS_SCT_BDD	Script de generación de la basa de datos	6,0	
	10	LHS_DGA_PRY	Diagramas de casos de uso	1,0	
	11	LHS_ESP_CUS	Especificación de casos de uso	1,0	
Implementación	12		Medidas-Métricas	1,0	
	13		Diagrama de n-capas lógicas	1,0	
	14	LHS_MAN_USR	Manual de usuario	1,0	
	15	LHS_ESP_TEC	Documento de especificaciones técnicas	2,0	
	16		LogiHistoricosRecursos	1,0	
	17		Plantilla estimación tamaño	1,0	
	18	LHS_FOR_AUD	Formulario de tablas a auditar	2,0	
Administración del Proyecto	19	LHS_ACT_SEI	Acta de seguimiento interno	13,0	

Figura 3.4 Inventario de la documentación del sistema (PLANTILLA 3).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.

5. El Responsable de mantenimiento se encarga de llenar el Registro de asignación de recursos mostrado en la Figura 3.5.



**DATOS GENERALES DEL PROYECTO O SISTEMA**

<b>Nombre del Proyecto</b>	LogiHistoricos
<b>Líder del Proyecto</b>	Isabel Vasco
<b>Responsable del Mantenimiento</b>	Gabriela Caiza

**REGISTRO DE ASIGNACIÓN DE RECURSOS**

ID. SOLICITUD	DESCRIPCIÓN O DETALLE	EQUIPO DE MANTENIMIENTO	ROL	OBSERVACIONES
SM01	*Generar un reporte de las métricas por proyecto y exportarlo a Excel *Calcular y generar un reporte de las métricas por responsable y exportarlo a Excel. *Generar un reporte de las métricas estándar y exportarlo a Excel.	María Vines	Analista - programador	
		Gabriela Caiza	Responsable de SQA Analista - SQA	
		Gabriela Caiza Marcela Balseca	Tester	
		Marcela Balseca	Documentador	

**Figura 3.5 Registro de asignación de recursos (PLANTILLA 4).**

**Elaborado por:** Balseca Marcela, Caiza Gabriela

**Fuente:** Información obtenida de la Empresa Logiciel Cía. Ltda.

6. El Equipo de mantenimiento se encarga de llenar la plantilla sobre la Definición del impacto del cambio mostrada en la Figura 3.6.



DATOS GENERALES DEL PROYECTO O SISTEMA	
Nombre del Proyecto	LogiHistoricos
Líder del Proyecto	Isabel Vasco
Equipo de Mantenimiento	María Vincés, Gabriela Caiza, Marcela Balseca

DEFINICIÓN DEL IMPACTO DEL CAMBIO					
ID	REQUERIMIENTO	ELEMENTO A MODIFICAR	ESPECIFICAR ELEMENTO A MODIFICAR	DESCRIPCIÓN DEL CAMBIO	OBSERVACIONES
RN1	Generar un reporte de las métricas por proyecto y exportarlo a Excel.	Sistema	Pantalla	Implementar una nueva pantalla que permita obtener un reporte de las métricas de un determinado proyecto. Además exportar el reporte a Excel. El nombre de la nueva pantalla será Métricas por proyecto.	
		Documentación	LHS_ESP_FUN	Añadir el nuevo requerimiento funcional.	
			LHS_FOR_GRQ	Añadir el nuevo requerimiento de gestión.	
			LHS_ESP_CUS	Actualizar los casos de uso con el nuevo requerimiento.	
			LHS_DGA_PRY	Actualizar los diagramas del sistema.	
			LHS_MAN_USR	Incluir la funcionalidad de la pantalla Métricas por proyecto.	
RN2	Calcular y generar un reporte de las métricas por responsable y exportarlo a Excel.	Sistema	Pantalla	Implementar una nueva pantalla que permita obtener un reporte de las métricas por responsable de un determinado proyecto . Además exportar el reporte a Excel. El nombre de la nueva pantalla será Métricas por responsable.	
		Documentación	LHS_ESP_FUN	Añadir el nuevo requerimiento funcional.	
			LHS_FOR_GRQ	Añadir el nuevo requerimiento de gestión.	
			LHS_ESP_CUS	Actualizar los casos de uso con el nuevo requerimiento.	
			LHS_DGA_PRY	Actualizar los diagramas del sistema.	
			LHS_MAN_USR	Incluir la funcionalidad de la pantalla Métricas por responsable.	
RN3	Generar un reporte de las métricas estándar y exportarlo a Excel.	Sistema	Pantalla	Implementar una nueva pantalla que permita obtener un reporte de las métricas estándar de una determinada metodología. Además exportar el reporte a Excel. El nombre de la nueva pantalla será Métricas estándar.	
		Documentación	LHS_ESP_FUN	Añadir el nuevo requerimiento funcional.	
			LHS_FOR_GRQ	Añadir el nuevo requerimiento de gestión.	
			LHS_ESP_CUS	Actualizar los casos de uso con el nuevo requerimiento.	
			LHS_DGA_PRY	Actualizar los diagramas del sistema.	
			LHS_MAN_USR	Incluir la funcionalidad de la pantalla Métricas estándar.	

**Figura 3.6 Definición del impacto del cambio (PLANTILLA 5).**

**Elaborado por:** Balseca Marcela, Caiza Gabriela

**Fuente:** Información obtenida de la Empresa Logiciel Cía. Ltda.

7. Lista de requerimientos funcionales del sistema LogiHistoricos mostrada en la Figura 3.7.

CÓD.		REQUERIMIENTO ACTUALES DE LOGIHISTORICOS
<b>RF1</b>	<b>Registro del proyecto</b>	
RF1.1	El Sistema para hacer el registro de un proyecto tomará la información publicada en el Project Server : <ul style="list-style-type: none"> <li>• Líder del proyecto</li> <li>• Descripción</li> <li>• Fechas inicio y fin del proyecto</li> <li>• Responsables asignados al proyecto</li> <li>• Fases, actividades, tareas, responsables por tarea</li> </ul>	
RF1.2	Será obligatorio el mapeo de las sub-fases de la fase de "Desarrollo" del cronograma del proyecto con las fases de la metodología seleccionada. Será opcional el mapeo de las actividades del cronograma del proyecto con las actividades de la metodología seleccionada.	
RF1.3	Los integrantes del proyecto deben estar previamente registrados como empleados, caso contrario no se hará el registro del proyecto.	
RF1.4	Las estimaciones de las medidas de esfuerzo, costo y duración se tomarán de los datos publicados en el Project Server: <ul style="list-style-type: none"> <li>• Previo al registro del proyecto publicado en el Project Server en el LogiHistoricos, el Líder del proyecto debe trazar la "línea base" inicial.</li> <li>• Estas estimaciones se toman de los campos "previsto". En caso de que no se haya trazado una línea base del proyecto se toma el dato del campo disponible del Project Server.</li> </ul>	
RF1.5	La estimación del tamaño se ingresará manualmente: <ul style="list-style-type: none"> <li>• Si el criterio de tamaño del proyecto es PCO, el dato del tamaño se tomará de la plantilla Excel para estimaciones.</li> <li>• El Líder del proyecto es el responsable de llenar la plantilla Excel para estimaciones</li> <li>• La medida del tamaño que debe ingresar el usuario es la medida ajustada, es decir <math>\text{Subtotal PCO} * (1 + \text{Factor de ajuste})</math></li> </ul>	
RF1.6	Cuando se registra un proyecto su estado deber ser "Abierto".	
RF1.7	El factor de ajuste está en relación a los riesgos del proyecto y debe ser menor o igual a 0,30 y mayor a cero.	
RF1.8	El Sistema al registrar el proyecto también deberá registrar las actividades, tareas y responsables por tarea del proyecto: <ul style="list-style-type: none"> <li>• Las actividades del proyecto deben estar atadas con la metodología de desarrollo.</li> </ul>	
RF1.9	Los datos de la "descripción", "fecha inicio" y "fecha fin" se tomarán de los datos publicados en el Project Server.	
RF1.10	Se podrá cambiar el tipo de una actividad siempre que no se haya registrado aún una medida en la actividad.	
RF1.11	Durante el registro inicial de las medidas estimadas por nivel se grabará el guid de la tarea que consta en Project Server <ul style="list-style-type: none"> <li>• Para la actualización de medidas por nivel se utilizará este guid para consultar en el Project Server las medidas de la tarea</li> </ul>	

<b>RF2</b>	<b>Recopilación de medidas del Project Server</b>
RF2.1	Se recopilarán las medidas de las tareas que fueron registradas inicialmente: <ul style="list-style-type: none"> <li>○ Si se requiere adicionar una nueva tarea tendrá que hacerse a nivel de <u>subtarea</u> (no a nivel de fase, actividad o tarea).</li> <li>○ Cada vez que se recopilan las medidas se actualizarán los campos de fecha inicio/fin de las actividades y tareas (comienzo y fin real)</li> <li>○ El estado de la tarea debe pasar a “Actualizado” cuando se actualizan los campos de fecha inicio/fin</li> </ul>
RF2.2	La duración o plazo de un proyecto se tomará del Project Server del campo “tiempo”
RF2.3	El esfuerzo de un proyecto se tomará del Project Server del campo “trabajo”
RF2.4	Si el proceso que recopila las medidas detecta que un proyecto ya no existe en la base de datos de “proyectos publicados”, deberá recopilar las últimas medidas del proyecto de la base de datos de “archivo de proyectos”, y cambiar el estado del proyecto a “Cerrado”. Procedimentalmente, antes de hacer el archivo del proyecto debe verificarse que está al 100% completado.
RF2.5	El registro periódico de las medidas recopiladas del Project Server no es obligatorio, pero sí se exigirá al cierre del proyecto.
RF2.6	Se debe permitir la homologación de los campos y medidas definidos en la aplicación con los campos de Project Server, para obtener los datos estimados, reales y proyectados.
<b>RF3</b>	<b>Recopilación medidas</b>
RF3.1	Mediante una macro Excel se invocará un servicio Web de la aplicación, al cual pasará las medidas (de tamaño) del proyecto de la hoja Excel para registrar en la base de datos de la aplicación
<b>RF4</b>	<b>Registro manual de medidas</b>
RF4.1	El registro de medidas “reales” se hará por tarea y responsable, y lo podrán hacer el Líder del proyecto o el Líder de SQM: <ul style="list-style-type: none"> <li>● El Líder de SQM sólo podrá hacer el registro de medidas que se recojan de los aseguramientos y pruebas.</li> </ul>
RF4.2	El registro de la medida “tamaño” sólo se podrá hacer en tareas de actividades de tipo “codificación”: <ul style="list-style-type: none"> <li>○ Las medidas de tamaño por responsable se calcularán con una plantilla Excel, que también podrá utilizarse para la estimación del tamaño, tanto por proyecto como por responsable.</li> </ul>
RF4.3	El registro de la medida “errores” sólo se podrá hacer en tareas de actividades de tipo “prueba”
RF4.4	El registro de la medida “inconformidades” sólo se podrá hacer en tareas de actividades de tipo “aseguramiento”.
RF4.5	Sólo se podrán actualizar los registros con tipo “proyectado” y “real”.
RF4.6	Las medidas de desviación en el Project Server deben estar establecidas con la siguiente fórmula: (valor proyectado – valor previsto) / valor estimado
RF4.7	Las medidas de registro manual deberán ingresarse periódicamente.

<b>RF5</b>	<b>Cálculo de medidas y métricas</b>
RF5.1	El Sistema hará la agregación (suma) de las medidas en este orden: tareas, actividades, fases y proyecto: <ul style="list-style-type: none"> <li>○ Mediante un flag se indicará si se hizo una última actualización de medidas</li> <li>○ La última medida se almacenará en conjunto con el registro de la nueva medida.</li> </ul>
RF5.2	El cálculo de las métricas se hará a nivel de fase y de proyecto: <ul style="list-style-type: none"> <li>○ Mediante un flag se indicará si se actualizaron las métricas de un proyecto que pasó a estado "cerrado"</li> <li>○ Una vez actualizadas las métricas el sistema cambiará el flag de actualización de medidas</li> <li>○ La última métrica se almacenará en conjunto con el registro de la nueva métrica calculada (para seguimiento del proyecto)</li> </ul>
RF5.3	El Sistema periódicamente debe hacer la actualización de medidas (de las que se toman del Project Server) y métricas.
RF5.4	Sólo se podrá asignar una medida a una métrica de desviación. Una métrica de desviación establece el porcentaje de desviación de una medida con respecto a lo inicialmente planificado para dicha medida.
<b>RF6</b>	<b>Criterios de tamaño</b>
RF6.1	La técnica para cálculo del tamaño de un proyecto debe estar acorde a los criterios de tamaño definidos en el Sistema.
RF6.2	Se debe registrar el factor de ajuste utilizado para calcular el tamaño del proyecto.
<b>RF7</b>	<b>Cálculo de medidas y métricas estándar</b>
RF7.1	El cálculo de medidas y métricas estándar se establece por: <ul style="list-style-type: none"> <li>○ Criterio de tamaño</li> <li>○ Metodología de desarrollo (fase -&gt; proyecto)</li> <li>○ Plataforma de desarrollo</li> </ul>
RF7.2	Para calcular las medidas y métricas estándar se agregarán (suma) los valores de las medidas y métricas de los proyectos con las mismas características de criterio de tamaño, metodología y plataforma de desarrollo, y se prorratearán por el número de proyectos, tanto a nivel de fase como de proyecto.
<b>RF8</b>	<b>Reportes de medidas</b>
RF8.1	El Sistema debe generar los siguiente reportes (vistas de base de datos): <ul style="list-style-type: none"> <li>○ Resumen de medidas por proyecto</li> <li>○ Resumen de medidas por fase del proyecto</li> <li>○ Resumen de medidas por responsable-tarea de un proyecto</li> </ul>
RF8.2	<u>Indicadores</u> : los reportes de las métricas deben emitir alertas de acuerdo a determinados rangos por métrica (indicador): <ul style="list-style-type: none"> <li>○ Los indicadores comparan las métricas del proyecto con las métricas estándar.</li> <li>○ Las métricas estándar son un promedio de las métricas de proyectos similares (proyectos similares son aquellos que tienen la misma metodología, plataforma de desarrollo y criterio de tamaño)</li> </ul>

<b>RF9</b>	<b>Autenticación y permisos</b>
RF9.1	Se utilizará una autenticación mixta (forms y Windows) para integrarse con LogiSeg e interactuar con Project Server.
RF9.2	Se aplicarán los siguientes criterios para parametrizar los permisos en LogiSeg: <ul style="list-style-type: none"> <li>• Tipo de recursos: páginas, menús, controles</li> <li>• A nivel de perfil-aplicación</li> <li>• Usar la tabla perfiles-empleado en lugar de usuarios-perfiles-aplicación</li> </ul>
<b>RF10</b>	<b>Cierre/eliminación de un proyecto</b>
RF10.1	Cuando algún proyecto quede en stand by y luego se archive, la aplicación debe hacer una eliminación lógica del proyecto y de sus históricos.

Figura 3.7 Lista de requerimientos funcionales del sistema LogiHistoricos.

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.

8. Resumen de los requerimientos del sistema LogiHistoricos a utilizarse en la matriz de trazabilidad mostrada en la Figura 3.8.

REQUERIMIENTOS ACTUALES		REQUERIMIENTOS NUEVOS	
RA°	Requerimiento	RN°	Requerimiento
RF1	Registro del proyecto	RN1	Generar un reporte de las métricas por proyecto
RF2	Recopilación de medidas del Project Server	RN2	Calcular y generar un reporte de las métricas por responsable
RF3	Recopilación de medidas	RN3	Generar un reportes de las métricas estándar
RF4	Registro manual de medidas		
RF5	Cálculo de medidas y métricas		
RF6	Criterios de tamaño		
RF7	Cálculo de medidas y métricas estándar		
RF8	Reportes de medidas		
RF9	Autenticación y permisos		
RF10	Cierre/eliminación de un proyecto		

Figura 3.8 Resumen de los requerimientos del sistema LogiHistoricos

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.

9. El Responsable de mantenimiento se encarga de llenar la Matriz de rastreabilidad mostrada en la Figura 3.9.

LogiHistoricos											
DATOS GENERALES DEL PROYECTO O SISTEMA											
Nombre del Proyecto	LogiHistoricos										
Líder del Proyecto	Isabel Vasco										
Responsable del Mantenimiento	Gabriela Caiza										
MATRIZ DE RASTREABILIDAD											
RC \ RA	RF 1	RF 2	RF 3	RF 4	RF 5	RF 6	RF 7	RF 8	RF 9	RF 10	OBSERVACIONES
RN1					X			X			En el RF 5 se extraen las métricas por proyecto y se calculan las métricas por responsable ; como se utilizarán estas metricas para los reportes por este motivo influyen en los nuevos requerimiento RC1 y RC2.
RN 2					X			X			En el RF 8 se implementan los indicadores para las métricas y como en todos los reportes se deben incluir los indicadores por este motivo influyen en todos los nuevos requerimientos.
RN 3							X	X			En el RF 7 se calculan las métricas estándar y por este motivo influye en el nuevo requerimiento RC3.

Figura 3.9 Matriz de rastreabilidad (PLANTILLA 10).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.

10. El Equipo de mantenimiento se encarga de llenar la plantilla de Estimación de costo y esfuerzo mostrada en la Figura 3.10.



**DATOS GENERALES DEL PROYECTO O SISTEMA**

<b>Nombre del Proyecto</b>	LogiHistoricos
<b>Líder del Proyecto</b>	Isabel Vasco
<b>Equipo de Mantenimiento</b>	María Vinces, Gabriela Caiza, Marcela Balseca

**ESTIMACIÓN DE COSTO Y ESFUERZO**

	HORAS/PERSONA	COSTO/HORA	COSTO TOTAL	OBSERVACIONES
RC 1	4	\$ 40,00	\$ 160,00	
RC 2	7	\$ 40,00	\$ 280,00	
RC 3	5	\$ 40,00	\$ 200,00	
		<b>COSTO TOTAL ESTIMADO</b>	<b>\$ 640,00</b>	

Figura 3.10 Estimación de costo y esfuerzo (PLANTILLA 11).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.

11.El Equipo de mantenimiento se encarga de llenar el Listado de elementos a modificar mostrado en la Figura 3.11.



**DATOS GENERALES DEL PROYECTO O SISTEMA**

<b>Nombre del Proyecto</b>	LogiHistoricos
<b>Líder del Proyecto</b>	Isabel Vasco
<b>Equipo de Mantenimiento</b>	María Vincés, Gabriela Caiza, Marcela Balseca

**LISTADO DE ELEMENTOS A MODIFICAR**

No.	ELEMENTO A MODIFICAR	CAPA DE LA ARQUITECTURA	DESCRIPCIÓN DEL CAMBIO A IMPLEMENTAR	RESPONSABLE DE LA MODIFICACIÓN	DURACIÓN	ESTADO DE LA MODIFICACIÓN	OBSERVACIONES
1	Sistema	VISTA	Crear los elementos de la interfaz de usuario para las pantallas. Implementar tres pantallas una para cada reporte; los nombres de cada una de las pantallas serán: *Métricas por proyecto *Métricas por responsable *Métricas estándar				
2		CONTROLADOR	Crear un control para cada reporte. Implementar las acciones que devuelven los datos para la vista.				
3		GESTOR	Realizar el cálculo de las métricas por responsable.				

**Figura 3.11 Listado de elementos a modificar (PLANTILLA 12).**

**Elaborado por:** Balseca Marcela, Caiza Gabriela

**Fuente:** Información obtenida de la Empresa Logiciel Cía. Ltda.

12.El Equipo de mantenimiento se encarga de realizar el Plan de prueba mostrado en la Figura 3.12.

## PLAN DE PRUEBAS LOGIHISTORICOS

### INTRODUCCIÓN

El presente documento corresponde al plan de pruebas para el sistema “LogiHistoricos”, mismo que permitirá el registro de información para las medidas de: tamaño, esfuerzo, número de errores y costo de un proyecto al término de una fase de una metodología de desarrollo. Las medidas se registrarán por tareas y responsables, esta información se tomará del cronograma del proyecto que será manejado desde Project Server 2010, con estos datos se realizará el cálculo de las métricas. El sistema generará reportes Excel de las métricas calculadas.

### ALCANCE

El objetivo del plan es definir las pruebas, y detalles de las mismas, a llevarse a cabo para verificar que se cumpla con la nueva funcionalidad definida para el sistema “LogiHistoricos”.

### MOTIVADORES DE LA PRUEBA

Los principales elementos que crearon la necesidad de llevar a cabo el proceso de pruebas son:

- Verificar la correcta implementación de los nuevos requerimientos definidos.
- Comprobar que los botones cumplan las funciones para las que fueron diseñados.
- Comprobar que se usen los estándares gráficos de la empresa.

### ENFOQUE DE LAS PRUEBAS

Los tipos de pruebas que se realizarán son:

- Pruebas de Funcionalidad

PRUEBAS DE FUNCIONALIDAD	
<b>Objetivo:</b>	El objetivo principal de esta prueba es comprobar que el sistema cumpla con la funcionalidad esperada.
<b>Descripción:</b>	Se probará la nueva funcionalidad implementada.
<b>Técnica:</b>	Para realizar estas pruebas se empleará un tester que conozca el funcionamiento que debería presentar el sistema y esté encargado de llevar a cabo pruebas manuales.
<b>Entorno de la prueba:</b>	Para las pruebas se hará uso de datos erróneos para ver cómo responde el sistema, además de datos válidos para comprobar resultados esperados.
<b>Aprobación de pruebas:</b>	Las pruebas serán consideradas como exitosas si se obtiene de las mismas los resultados esperados, no se admitirán resultados erróneos.

- Pruebas de Interface de usuario

### PRUEBAS DE INTERFACE DE USUARIO

<b>Objetivo:</b>	El objetivo principal de esta prueba es comprobar el uso de estándares y las validaciones de los campos.
<b>Descripción:</b>	Se probará que cada campo se encuentre correctamente validado y que los controles en la interfaz se encuentren bien distribuidos y cumplan con los estándares de la empresa.
<b>Técnica:</b>	Para realizar estas pruebas se empleará un tester que conozca sobre los estándares gráficos y esté encargado de llevar a cabo pruebas manuales para la verificación de validaciones.
<b>Entorno de la prueba:</b>	Para las pruebas se hará uso de datos erróneos para ver cómo responde el sistema, además se comparará los componentes implementados con los diseñados.
<b>Aprobación de pruebas:</b>	Las pruebas serán consideradas como exitosas si todos los campos están validados y las interfaces están implementadas de acuerdo a los estándares.

#### ➤ Pruebas de Estrés

### PRUEBAS DE ESTRÉS

<b>Objetivo:</b>	El objetivo principal de esta prueba es determinar la solidez de la aplicación en los momentos de carga extrema y ayudar a los administradores para determinar si la aplicación rendirá lo suficiente en caso de que la carga real supere a la carga esperada.
<b>Descripción:</b>	Se probará la nueva funcionalidad implementada.
<b>Técnica:</b>	Para realizar estas pruebas se emplearán a varias personas de la empresa que estén fuera del Equipo de mantenimiento.
<b>Entorno de la prueba:</b>	Para las pruebas se hará uso de datos erróneos para ver cómo responde el sistema, además de datos válidos para comprobar resultados esperados.
<b>Aprobación de pruebas:</b>	Las pruebas serán consideradas como exitosas si se obtiene de las mismas los resultados esperados, no se admitirán resultados erróneos.

#### ➤ Pruebas de Aceptación

### PRUEBAS DE ACEPTACIÓN

<b>Objetivo:</b>	El objetivo principal de esta prueba es comprobar que el sistema cumpla con la funcionalidad esperada.
<b>Descripción:</b>	Se probará la nueva funcionalidad implementada.
<b>Técnica:</b>	Para realizar estas pruebas se empleará a una persona de la empresa que este fuera del Equipo de mantenimiento.
<b>Entorno de la prueba:</b>	Para las pruebas se hará uso de datos erróneos para ver cómo responde el sistema, además de datos válidos para comprobar resultados esperados.
<b>Aprobación de pruebas:</b>	Las pruebas serán consideradas como exitosas si se obtiene de las mismas los resultados esperados, no se admitirán resultados erróneos.

## **CRITERIOS PARA PRUEBAS**

### **CRITERIOS DE ACEPTACIÓN DE REQUERIMIENTOS**

Se cumpla con el 100% de los requerimientos establecidos para esta versión.

### **CRITERIOS PARA INICIO DE PRUEBAS**

Para que se pueda dar inicio al proceso de pruebas se debe:

- Completar la iteración para realizar las pruebas correspondientes a la misma.
- Contar con los ambientes listos.
- Contar con el personal capacitado para realizar la prueba.

### **CRITERIOS PARA FIN DE PRUEBAS**

Para dar por concluido el proceso de pruebas se debe contar con:

- La comprobación de la implementación total de los requerimientos.
- La comprobación de validaciones en todos los campos
- El cumplimiento de estándares gráficos.

Se deben realizar las iteraciones necesarias hasta que la aplicación cumpla con todo lo establecido.

### **CRITERIOS PARA SUSPENSIÓN**

El proceso de pruebas se suspenderá cuando:

- La funcionalidad a ser probada aun no esté lista.
- Se presente nueva funcionalidad que no fue considerada inicialmente y sea necesaria o indispensable.
- Se presenten muchos errores en el componente que está siendo probado.

Todos estos criterios implican que se debe planificar una nueva fecha para llevar a cabo las pruebas.

### **CRITERIOS PARA RETOMAR LAS PRUEBAS**

El proceso de pruebas se retomará cuando:

- La funcionalidad a ser probada ya esté lista.
- La nueva funcionalidad que no fue considerada inicialmente ya esté implementada.
- Los errores reportados ya han sido corregidos.

Figura 3.12 Plan de pruebas del sistema LogiHistoricos (PLANTILLA 13).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.

13.El Responsable de mantenimiento se encarga de realizar el Plan de implementación mostrado en la Figura 3.13.

										
DATOS GENERALES DEL PROYECTO O SISTEMA										
Nombre del Proyecto	LogiHistoricos									
Líder del Proyecto	Isabel Vasco									
Responsable de Mantenimiento	Gabriela Caiza									
PLAN DE IMPLEMENTACIÓN										
ID. SOLICITUD	METAS PARA CADA PERIODO					RESPONSABLES				
	Periodo 1	Periodo 2	Periodo 3	Periodo 4	Fin del Proyecto	Equipo de Mantenimiento				
SM01	09/06/2014	10/06/2014	11/06/2014	12/06/2014	13/06/2014	Responsable de Mantenimiento	Analista programador	SQA	Tester	Documentador
Revisar el documento de elementos a modificar						X	X			
Implementar la pantalla y la funcionalidad para generar el reporte de las Métricas por proyecto							X			
Implementar la pantalla y la funcionalidad para generar el reporte de las Métricas por responsable							X			
Implementar la pantalla y la funcionalidad para generar el reporte de las Métricas estándar							X			
Revisar el listado de documentos a modificar								X		X
Actualizar la documentación del sistema						X				X
Ejecutar las pruebas de funcionalidad						X			X	
Ejecutar las pruebas de aceptación						X			X	
Realizar el aseguramiento de calidad de los artefactos internos actualizados								X		

Figura 3.13 Plan de implementación inicial (PLANTILLA 14).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.

### FASE 3: Diseño

14. El Responsable de mantenimiento se encarga de actualizar el Listado de elementos a modificar mostrado en la Figura 3.14.

LogiHistoricos							
DATOS GENERALES DEL PROYECTO O SISTEMA							
Nombre del Proyecto	LogiHistoricos						
Líder del Proyecto	Isabel Vasco						
Equipo de Mantenimiento	María Vincés, Gabriela Caiza, Marcela Balseca						
LISTADO DE ELEMENTOS A MODIFICAR							
No.	ELEMENTO A MODIFICAR	CAPA DE LA ARQUITECTURA	DESCRIPCIÓN DEL CAMBIO A IMPLEMENTAR	RESPONSABLE DE LA MODIFICACIÓN	DURACIÓN	ESTADO DE LA MODIFICACIÓN	OBSERVACIONES
1	Sistema	VISTA	Crear los elementos de la interfaz de usuario para las pantallas. Implementar tres pantallas una para cada reporte; los nombres de cada una de las pantallas serán: *Métricas por proyecto *Métricas por responsable *Métricas estándar	Analista-programador	8h	No iniciado	
2		CONTROLADOR	Crear un control para cada reporte. Implementar las acciones que devuelven los datos para la vista.	Analista-programador	4h	No iniciado	
3		GESTOR	Realizar el cálculo de las métricas por responsable.	Analista-programador	4h	No iniciado	

Figura 3.14 Listado de elementos a modificar actualizado y revisado (PLANTILLA 12).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.

15.El Responsable de mantenimiento se encarga de realizar el Listado de documentos a modificar mostrado en la Figura 3.15.

								
DATOS GENERALES DEL PROYECTO O SISTEMA								
Nombre del Proyecto	LogiHistoricos							
Líder del Proyecto	Isabel Vasco							
Equipo de Mantenimiento	María Vincés, Gabriela Caiza, Marcela Balseca							
LISTADO DE DOCUMENTOS A MODIFICAR								
FASE	No.	ID - DOCUMENTO	DESCRIPCIÓN	VERSIÓN	RESPONSABLE DE LA MODIFICACIÓN	DURACIÓN	ESTADO DE LA ACTUALIZACIÓN	OBSERVACIONES
Gestión de Requerimientos	1	LHS_ESP_FUN	Especificación de requerimientos	2.0	Documentador	1h	No iniciado	
	2	LHS_FOR_GRQ	Gestión de requerimientos	2,0	Documentador	1h	No iniciado	
Análisis y Diseño	3	LHS_ESP_CUS	Especificación de casos de uso	1,0	Documentador	3h	No iniciado	
	4	LHS_DGA_PRY	Diagramas de casos de uso	1,0	Documentador	3h	No iniciado	
Implementación	5	LHS_MAN_USR	Manual de usuario	1,0	Documentador	8h	No iniciado	

Figura 3.15 Listado de documentos a modificar (PLANTILLA 15).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.

16. El Equipo de mantenimiento se encarga de realizar los Casos de prueba: los casos de pruebas de funcionalidad se muestran en las Figuras 3.16, 3.17 y 3.18, los casos de prueba de rendimiento se muestran en las Figuras 3.19, 3.20, 3.21 y los casos de prueba de aceptación se muestran en las Figuras 3.22, 3.23 y 3.24.



DATOS DEL CASO DE PRUEBA	
Nombre del Proyecto	LogiHistoricos
Id. Requerimiento	RN1
Id. Caso de prueba	CP01
Tipo de prueba	Funcionalidad
Módulo a probar	Registro de Medidas
Autor del caso de prueba	Analista programador/ Tester
Nombre del tester	Gabriela Caiza
Objetivo	Probar la funcionalidad de la pantalla de métricas por proyecto.
Descripción	Validar la funcionalidad de la pantalla métricas por proyecto.
Fecha de creación	05/06/2014
Fecha de ejecución:	

EJECUCIÓN Y RESULTADOS DEL CASO DE PRUEBA:

ID.	ELEMENTO A PROBAR	PRECONDICIÓN	DATOS DE ENTRADA	RESULTADO ESPERADO	RESULTADO OBTENIDO	DOCUMENTOS DE RESPALDO	ESTADO DEL CASO DE PRUEBA	FECHA DE APROBACIÓN
1	Métricas por proyecto	Disponibilidad de la aplicación y de la pantalla mencionada.	Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Proyecto" 4.-Seleccionar del filtro Proyecto:PRY_Proceso de atención de incidencias	Se visualice en pantalla las métricas existentes del proyecto PRY_Proceso de atención de incidencias. En la pantalla se pueda ver el valor de las métrica y los indicadores.			Generado	
2			Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Proyecto" 4.-Seleccionar del filtro Proyecto:PRY_Proceso de atención de incidencias 5.-Presionar el botón Exportar	Se genera el reporte en Excel con las métricas existentes del proyecto PRY_Proceso de atención de incidencias. En el reporte se presenta el valor de las métricas, los indicadores y las observaciones.				
3			Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Proyecto" 4.-Seleccionar del filtro Proyecto:PRY_Proceso de atención de incidencias 5.-Presionar el botón Salir	Se cierra la pantalla de Métricas por Proyecto.				

Figura 3.16 Caso de Prueba 01 (PLANTILLA 16-CP01).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.



DATOS DEL CASO DE PRUEBA	
Nombre del Proyecto	LogiHistoricos
Id. Requerimiento	RN2
Id. Caso de prueba	CP02
Tipo de prueba	Funcionalidad
Módulo a probar	Registro de Medidas
Autor del caso de prueba	Analista programador/ Tester
Nombre del tester	Gabriela Caiza
Objetivo	Probar la funcionalidad de la pantalla de métricas por responsable.
Descripción	Validar la funcionalidad de la pantalla métricas por responsable.
Fecha de creación	05/06/2014
Fecha de ejecución:	

EJECUCIÓN Y RESULTADOS DEL CASO DE PRUEBA:								
ID.	ELEMENTO A PROBAR	PRECONDICIÓN	DATOS DE ENTRADA	RESULTADO ESPERADO	RESULTADO OBTENIDO	DOCUMENTOS DE RESPALDO	ESTADO DEL CASO DE PRUEBA	FECHA DE APROBACIÓN
1	Métricas por Responsable	Disponibilidad de la aplicación y de la pantalla mencionada.	Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Responsable" 4.-Seleccionar del filtro Proyecto: PRY_Proceso de atención de incidencias 5.-Seleccionar del filtro Responsable: Isabel Vasco Aguas.	Se visualice en pantalla las métricas existentes del proyecto PRY_Proceso de atención de incidencias del responsable Isabel Vasco Aguas . En la pantalla se pueda ver el valor de las métrica y los indicadores.			Generado	
2			Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Responsable" 4.-Seleccionar del filtro Proyecto: PRY_Proceso de atención de incidencias 5.-Seleccionar del filtro Responsable: Isabel Vasco Aguas. 6.-Presionar el botón Exportar	Se generará el reporte en excel con las métricas existentes del proyecto PRY_Proceso de atención de incidencias del responsable Isabel Vasco Aguas. En el reporte el valor de las métricas y los indicadores.				
3			Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Responsable" 4.-Seleccionar del filtro Proyecto: PRY_Proceso de atención de incidencias 5.-Seleccionar del filtro Responsable: Isabel Vasco Aguas. 6.-Presionar el botón Salir	Se cierra la pantalla de Métricas por Responsable.				

**Figura 3.17 Caso de Prueba 02 (PLANTILLA 16-CP02).**

**Elaborado por:** Balseca Marcela, Caiza Gabriela

**Fuente:** Información obtenida de la Empresa Logiciel Cía. Ltda.



DATOS DEL CASO DE PRUEBA	
Nombre del Proyecto	LogiHistoricos
Id. Requerimiento	RN3
Id. Caso de prueba	CP03
Tipo de prueba	Funcionalidad
Módulo a probar	Registro de Medidas
Autor del caso de prueba	Analista programador/ Tester
Nombre del tester	Gabriela Caiza
Objetivo	Probar la funcionalidad de la pantalla de métricas estándar.
Descripción	Validar la funcionalidad de la pantalla métricas estándar.
Fecha de creación	05/06/2014
Fecha de ejecución:	

#### EJECUCIÓN Y RESULTADOS DEL CASO DE PRUEBA:

ID.	ELEMENTO A PROBAR	PRECONDICIÓN	DATOS DE ENTRADA	RESULTADO ESPERADO	RESULTADO OBTENIDO	DOCUMENTOS DE RESPALDO	ESTADO DEL CASO DE PRUEBA	FECHA DE APROBACIÓN
1	Métricas Estándar	Disponibilidad de la aplicación y de la pantalla mencionada.	Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas Estándar" 4.-Seleccionar del filtro Metodología:Proyectos Parametrización 5.-Seleccionar del filtro Plataforma: .Net 6.-Seleccionar del filtro Criterio Tamaño: PCO	Se visualice en pantalla las métricas estándar existentes para la metodología Proyectos Parametrización con plataforma .Net y criterio de tamaño PCO. En la pantalla se pueda ver el valor de las métrica y los indicadores.			Generado	
2			Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas Estándar" 3.- Seleccionar del filtro Metodología:Proyectos Parametrización 4.-Seleccionar del filtro Plataforma: .Net 5.-Seleccionar del filtro Criterio Tamaño: PCO 6.-Presionar el botón Exportar	Se genera el reporte en Excel con las métricas existentes del proyecto para la metodología Proyectos Parametrización con plataforma .Net y criterio de tamaño PCO. En el reporte se presenta el valor de las métricas y los indicadores.				
3			Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas Estándar" 4.-Seleccionar del filtro Metodología:Proyectos Parametrización 5.-Seleccionar del filtro Plataforma: .Net 6.-Seleccionar del filtro Criterio Tamaño: PCO 7.-Presionar el botón Salir	Se cierra la pantalla de Métricas Estándar.				

Figura 3.18 Caso de Prueba 03 (PLANTILLA 16-CP03).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.



DATOS DEL CASO DE PRUEBA	
Nombre del Proyecto	LogiHistoricos
Id. Requerimiento	RN1
Id. Caso de prueba	CPR01
Tipo de prueba	Estrés
Módulo a probar	Registro de Medidas
Autor del caso de prueba	Analista programador/ Tester
Nombre del tester	Gabriela Caiza
Objetivo	Probar la solidez de la aplicación en los momentos de carga.
Descripción	Validar el rendimiento del sistema LogiHistoricos al generar varias reportes sobre las métricas por proyecto a la misma vez con diferentes usuarios.
Fecha de creación	12/09/2014
Fecha de ejecución:	

#### EJECUCIÓN Y RESULTADOS DEL CASO DE PRUEBA:

ID.	ELEMENTO A PROBAR	PRECONDICIÓN	DATOS DE ENTRADA	RESULTADO ESPERADO	RESULTADO OBTENIDO	DOCUMENTOS DE RESPALDO	ESTADO DEL CASO DE PRUEBA	FECHA DE APROBACIÓN
1	Métricas por proyecto	Disponibilidad de la aplicación y de la pantalla mencionada.	Pasos a realizar usuario 1: 1.-Autenticarse en el sistema el usuario 1: ebalseca 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Proyecto" 4.-Seleccionar del filtro Proyecto:PRY_Proceso de atención de incidencias 5.-Presionar el botón Exportar	Se genera el reporte en Excel con las métricas existentes del proyecto PRY_Proceso de atención de incidencias.			Generado	
2			Pasos a realizar usuario 2: 1.-Autenticarse en el sistema el usuario 2: esalas. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Proyecto" 4.-Seleccionar del filtro Proyecto:PRY_Repositorio de medidas y metricas 5.-Presionar el botón Exportar	Se genera el reporte en Excel con las métricas existentes del proyecto PRY_Proceso de atención de incidencias.				
3			Pasos a realizar usuario 3: 1.-Autenticarse en el sistema el usuario 3: gcaiza. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Proyecto" 4.-Seleccionar del filtro Proyecto:PRY_Proceso de atención de incidencias 5.-Presionar el botón Exportar	Se genera el reporte en Excel con las métricas existentes del proyecto PRY_Proceso de atención de incidencias.				
4			Pasos a realizar usuario 4: 1.-Autenticarse en el sistema el usuario 4: cperez. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Proyecto" 4.-Seleccionar del filtro Proyecto:PRY_Proceso de atención de incidencias 5.-Presionar el botón Exportar	Se genera el reporte en Excel con las métricas existentes del proyecto PRY_Proceso de atención de incidencias.				

Figura 3.19 Caso de Prueba de Estrés 01 (PLANTILLA 16-CPR01).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.



DATOS DEL CASO DE PRUEBA	
Nombre del Proyecto	LogiHistoricos
Id. Requerimiento	RN2
Id. Caso de prueba	CPR02
Tipo de prueba	Estrés
Módulo a probar	Registro de Medidas
Autor del caso de prueba	Analista programador/ Tester
Nombre del tester	Gabriela Caiza
Objetivo	Probar la solidez de la aplicación en los momentos de carga.
Descripción	Validar el rendimiento del sistema LogiHistoricos al generar varios reportes sobre las métricas estándar a la vez.
Fecha de creación	12/09/2014
Fecha de ejecución:	

#### EJECUCIÓN Y RESULTADOS DEL CASO DE PRUEBA:

ID.	ELEMENTO A PROBAR	PRECONDICIÓN	DATOS DE ENTRADA	RESULTADO ESPERADO	RESULTADO OBTENIDO	DOCUMENTOS DE RESPALDO	ESTADO DEL CASO DE PRUEBA	FECHA DE APROBACIÓN
1	Métricas por Responsable	Disponibilidad de la aplicación y de la pantalla mencionada.	Pasos a realizar usuario 1: 1.-Autenticarse en el sistema el usuario 1: ebalseca. 2.-Seleccionar del Menú Principal la opción "Reportes". 3.-Seleccionar del menú de Reportes la opción "Métricas por Responsable" 4.-Seleccionar del filtro Proyecto: PRY_Proceso de atención de incidencias 5.-Seleccionar del filtro Responsable: Isabel Vasco Aguas. 6.-Presionar el botón Exportar	Se generará el reporte en excel con las métricas existentes del proyecto PRY_Proceso de atención de incidencias del responsable Isabel Vasco Aguas.			Generado	
2			Pasos a realizar usuario 2: 1.-Autenticarse en el sistema el usuario 2: esalas. 2.-Seleccionar del Menú Principal la opción "Reportes". 3.-Seleccionar del menú de Reportes la opción "Métricas por Responsable" 4.-Seleccionar del filtro Proyecto: PRY_Proceso de atención de incidencias 5.-Seleccionar del filtro Responsable: Isabel Vasco Aguas. 6.-Presionar el botón Exportar	Se generará el reporte en excel con las métricas existentes del proyecto PRY_Proceso de atención de incidencias del responsable Isabel Vasco Aguas.				
3			Pasos a realizar usuario 3: 1.-Autenticarse en el sistema el usuario 3: gcaiza. 2.-Seleccionar del Menú Principal la opción "Reportes". 3.-Seleccionar del menú de Reportes la opción "Métricas por Responsable" 4.-Seleccionar del filtro Proyecto: PRY_Proceso de atención de incidencias 5.-Seleccionar del filtro Responsable: Isabel Vasco Aguas. 6.-Presionar el botón Exportar	Se generará el reporte en excel con las métricas existentes del proyecto PRY_Proceso de atención de incidencias del responsable Isabel Vasco Aguas.				
4			Pasos a realizar usuario 4: 1.-Autenticarse en el sistema el usuario 4: pperez. 2.-Seleccionar del Menú Principal la opción "Reportes". 3.-Seleccionar del menú de Reportes la opción "Métricas por Responsable" 4.-Seleccionar del filtro Proyecto: PRY_Proceso de atención de incidencias 5.-Seleccionar del filtro Responsable: Isabel Vasco Aguas. 6.-Presionar el botón Exportar	Se generará el reporte en excel con las métricas existentes del proyecto PRY_Proceso de atención de incidencias del responsable Isabel Vasco Aguas.				

Figura 3.20 Caso de Prueba de Estrés 02 (PLANTILLA 16-CPR02).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.



DATOS DEL CASO DE PRUEBA	
Nombre del Proyecto	LogiHistoricos
Id. Requerimiento	RN3
Id. Caso de prueba	CPR03
Tipo de prueba	Estrés
Módulo a probar	Registro de Medidas
Autor del caso de prueba	Analista programador/ Tester
Nombre del tester	Gabriela Caiza
Objetivo	Probar la solidez de la aplicación en los momentos de carga.
Descripción	Validar el rendimiento del sistema LogiHistoricos al generar varias reportes sobre las métricas estándar a la vez.
Fecha de creación	12/09/2014
Fecha de ejecución:	

#### EJECUCIÓN Y RESULTADOS DEL CASO DE PRUEBA:

ID.	ELEMENTO A PROBAR	PRECONDICIÓN	DATOS DE ENTRADA	RESULTADO ESPERADO	RESULTADO OBTENIDO	DOCUMENTOS DE RESPALDO	ESTADO DEL CASO DE PRUEBA	FECHA DE APROBACIÓN
1	Métricas Estándar	Disponibilidad de la aplicación y de la pantalla mencionada.	Pasos a realizar usuario 1: 1.-Autenticarse en el sistema usuario 1: ebalseca. 2.-Seleccionar del Menú Principal la opción Reportes. 3.- Seleccionar del menú de Reportes la opción "Métricas Estándar" 3.- Seleccionar del filtro Metodología:Proyectos Parametrización 4.- Seleccionar del filtro Plataforma: .Net 5.-Seleccionar del filtro Criterio Tamaño: PCO 6.-Presionar el botón Exportar	Se genera el reporte en Excel con las métricas existentes del proyecto para la metodología Proyectos Parametrización con plataforma .Net y criterio de tamaño PCO.			Generado	
2			Pasos a realizar usuario 2: 1.-Autenticarse en el sistema usuario 2: esalas. 2.-Seleccionar del Menú Principal la opción Reportes. 3.- Seleccionar del menú de Reportes la opción "Métricas Estándar" 3.- Seleccionar del filtro Metodología:Proyectos Parametrización 4.- Seleccionar del filtro Plataforma: .Net 5.-Seleccionar del filtro Criterio Tamaño: PCO 6.-Presionar el botón Exportar	Se genera el reporte en Excel con las métricas existentes del proyecto para la metodología Proyectos Parametrización con plataforma .Net y criterio de tamaño PCO.				
3			Pasos a realizar usuario 3: 1.-Autenticarse en el sistema usuario 3: gcaiza. 2.-Seleccionar del Menú Principal la opción Reportes. 3.- Seleccionar del menú de Reportes la opción "Métricas Estándar" 3.- Seleccionar del filtro Metodología:Proyectos Parametrización 4.- Seleccionar del filtro Plataforma: .Net 5.-Seleccionar del filtro Criterio Tamaño: PCO 6.-Presionar el botón Exportar	Se genera el reporte en Excel con las métricas existentes del proyecto para la metodología Proyectos Parametrización con plataforma .Net y criterio de tamaño PCO.				
4			Pasos a realizar usuario 4: 1.-Autenticarse en el sistema usuario 4: pperez. 2.-Seleccionar del Menú Principal la opción Reportes. 3.- Seleccionar del menú de Reportes la opción "Métricas Estándar" 3.- Seleccionar del filtro Metodología:Proyectos Parametrización 4.- Seleccionar del filtro Plataforma: .Net 5.-Seleccionar del filtro Criterio Tamaño: PCO 6.-Presionar el botón Exportar	Se genera el reporte en Excel con las métricas existentes del proyecto para la metodología Proyectos Parametrización con plataforma .Net y criterio de tamaño PCO.				

Figura 3.21 Caso de Prueba de Estrés 03 (PLANTILLA 16-CPR03).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.



DATOS DEL CASO DE PRUEBA			
Nombre del Proyecto	LogiHistoricos		
Id. Requerimiento	RN1		
Id. Caso de prueba	CPA01		
Tipo de prueba	Aceptación		
Módulo a probar	Registro de Medidas		
Autor del caso de prueba	Analista programador/ Tester		
Nombre del tester	Marcela Balseca		
Objetivo	Probar la funcionalidad de la pantalla de métricas por proyecto.		
Descripción	Validar la funcionalidad de la pantalla métricas por proyecto.		
Fecha de creación	06/06/2014	Fecha de ejecución:	

#### EJECUCIÓN Y RESULTADOS DEL CASO DE PRUEBA:

ID.	ELEMENTO A PROBAR	PRECONDICIÓN	DATOS DE ENTRADA	RESULTADO ESPERADO	RESULTADO OBTENIDO	ESTADO DEL CASO DE PRUEBA	FECHA DE APROBACIÓN
1	Métricas por proyecto	Disponibilidad de la aplicación y de la pantalla mencionada.	Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Proyecto" 4.-Seleccionar del filtro un proyecto determinado.	Se visualice en la pantalla las métricas existentes del proyecto seleccionada y se pueda visualizar el valores de las métricas y los indicadores.		Generado	
2			Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Proyecto" 4.-Seleccionar del filtro un proyecto determinado 5.-Presionar el botón Exportar	Se genera el reporte en Excel con las métricas existentes del proyecto seleccionado y se pueda visualizar el valor de las métricas, los indicadores y las observaciones.			
3			Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Proyecto" 4.-Seleccionar del filtro un proyecto determinado 5.-Presionar el botón Salir	Se cierre la pantalla de Métricas por Proyecto.			

Figura 3.22 Caso de Prueba de Aceptación 01 (PLANTILLA 16-CPA01).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.



DATOS DEL CASO DE PRUEBA	
Nombre del Proyecto	LogiHistoricos
Id. Requerimiento	RN2
Id. Caso de prueba	CPA02
Tipo de prueba	Aceptación
Módulo a probar	Registro de Medidas
Autor del caso de prueba	Analista programador/ Tester
Nombre del tester	Marcela Balseca
Objetivo	Probar la funcionalidad de la pantalla de métricas por proyecto.
Descripción	Validar la funcionalidad de la pantalla métricas por proyecto.
Fecha de creación	06/06/2014
Fecha de ejecución:	

EJECUCIÓN Y RESULTADOS DEL CASO DE PRUEBA:

ID.	ELEMENTO A PROBAR	PRECONDICIÓN	DATOS DE ENTRADA	RESULTADO ESPERADO	RESULTADO OBTENIDO	ESTADO DEL CASO DE PRUEBA	FECHA DE APROBACIÓN
1	Métricas por proyecto	Disponibilidad de la aplicación y de la pantalla mencionada.	Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Proyecto" 4.-Seleccionar del filtro un proyecto determinado.	Se visualice en la pantalla las métricas existentes del proyecto seleccionada y se pueda visualizar el valores de las métricas y los indicadores.		Generado	
2			Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Proyecto" 4.-Seleccionar del filtro un proyecto determinado 5.-Presionar el botón Exportar	Se genera el reporte en Excel con las métricas existentes del proyecto seleccionado y se pueda visualizar el valor de las métricas, los indicadores y las observaciones.			
3			Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Proyecto" 4.-Seleccionar del filtro un proyecto determinado 5.-Presionar el botón Salir	Se cierre la pantalla de Métricas por Proyecto.			

Figura 3.23 Caso de Prueba de Aceptación 02 (PLANTILLA 16-CPA02).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.



DATOS DEL CASO DE PRUEBA			
Nombre del Proyecto	LogiHistoricos		
Id. Requerimiento	RN3		
Id. Caso de prueba	CPA03		
Tipo de prueba	Aceptación		
Módulo a probar	Registro de Medidas		
Autor del caso de prueba	Analista programador/ Tester		
Nombre del tester	Marcela Balseca		
Objetivo	Probar la funcionalidad de la pantalla de métricas estándar.		
Descripción	Validar la funcionalidad de la pantalla métricas estándar.		
Fecha de creación	06/06/2014	Fecha de ejecución:	

EJECUCIÓN Y RESULTADOS DEL CASO DE PRUEBA:

ID.	ELEMENTO A PROBAR	PRECONDICIÓN	DATOS DE ENTRADA	RESULTADO ESPERADO	RESULTADO OBTENIDO	ESTADO DEL CASO DE PRUEBA	FECHA DE APROBACIÓN
1	Métricas Estándar	Disponibilidad de la aplicación y de la pantalla mencionada.	Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas Estándar" 4.-Seleccionar del panel de filtros: una Metodología, una Plataforma y un Criterio Tamaño.	Se visualice en pantalla las métricas estándar existentes de acuerdo a los filtros seleccionados y se pueda visualizar el valor de las métrica y los indicadores.		Generado	
2			Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas Estándar" 4.-Seleccionar del panel de filtros: una Metodología, una Plataforma y un Criterio Tamaño. 5.-Presionar el botón Exportar	Se genera el reporte en Excel de acuerdo a los filtros seleccionados y se pueda visualizar el valor de las métricas y los indicadores.			
3			Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas Estándar" 4.-Seleccionar del panel de filtros: una Metodología, una Plataforma y un Criterio Tamaño. 5.-Presionar el botón Salir	Se cierra la pantalla de Métricas Estándar.			

Figura 3.24 Caso de Prueba de Aceptación 03 (PLANTILLA 16-CPA03).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.

17.El Responsable de mantenimiento se encarga de actualizar el Plan de implementación mostrado en la Figura 3.25.

LogiHistoricos										
DATOS GENERALES DEL PROYECTO O SISTEMA										
Nombre del Proyecto		LogiHistoricos								
Líder del Proyecto		Isabel Vasco								
Responsable de Mantenimiento		Gabriela Caiza								
PLAN DE IMPLEMENTACIÓN										
METAS PARA CADA PERIODO						RESPONSABLES				
ID. SOLICITUD	Periodo 1	Periodo 2	Periodo 3	Periodo 4	Fin del Proyecto	Equipo de Mantenimiento				
SM01	09/06/2014	10/06/2014	11/06/2014	12/06/2014	13/06/2014	Responsable de Mantenimiento	Analista programador	SQA	Tester	Documentador
Revisar el documento de elementos a modificar						X	X			
Implementar la pantalla y la funcionalidad para generar el reporte de las Métricas por proyecto						X	X			
Implementar la pantalla y la funcionalidad para generar el reporte de las Métricas por responsable						X	X			
Implementar la pantalla y la funcionalidad para generar el reporte de las Métricas estándar						X	X			
Revisar el listado de documentos a modificar						X		X		X
Actualizar la documentación del sistema						X				X
Ejecutar las pruebas de funcionalidad						X			X	
Ejecutar las pruebas de aceptación						X			X	
Realizar el aseguramiento de calidad de los artefactos internos actualizados						X		X		

Figura 3.25 Plan de implementación actualizado y revisado (PLANTILLA 14).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.

## FASE 4: Implementación

El sistema LogiHistoricos antes de implementar el cambio:

a) Pantalla principal del sistema: mostrada en la Figura 3.26.

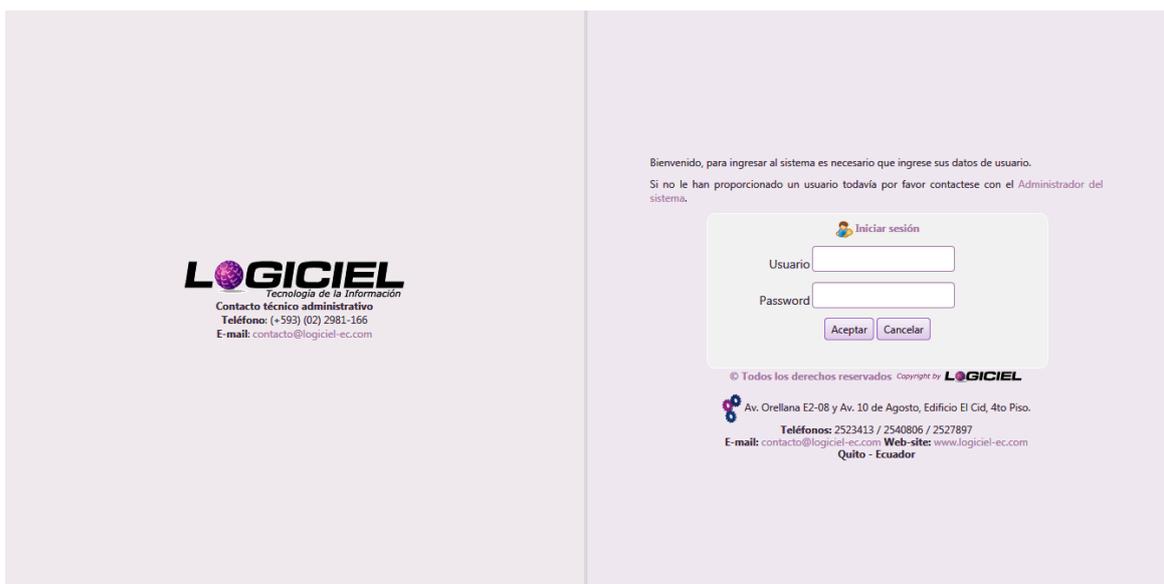


Figura 3.26 Pantalla principal de LogiHistoricos.

Fuente: Sistema de la empresa Logiciel Cía. Ltda.

b) Menú principal del sistema: mostrado en la Figura 3.27.

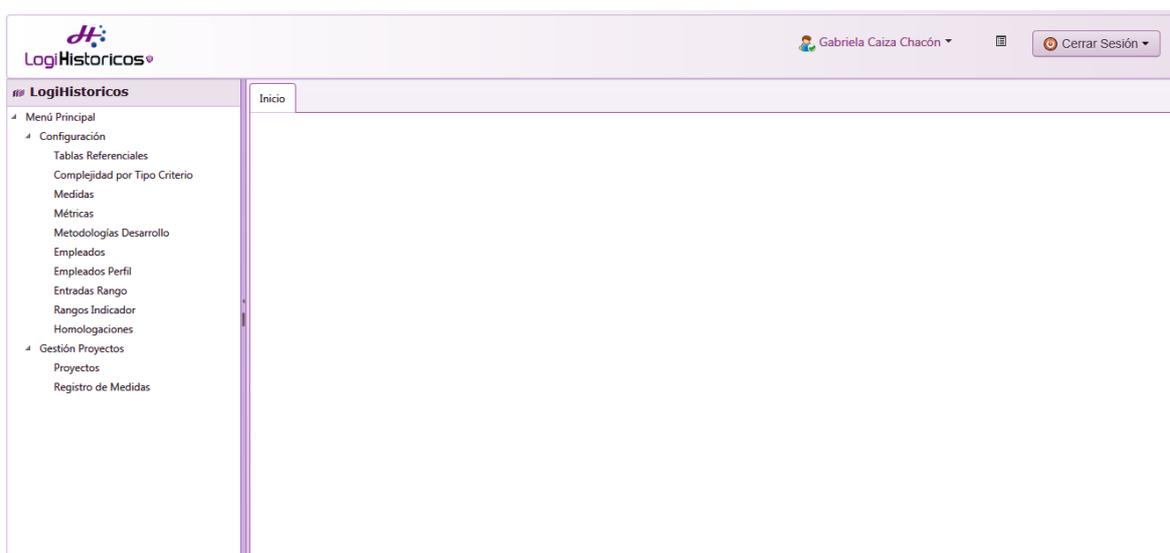


Figura 3.27 Menú principal de LogiHistoricos antes del cambio.

Fuente: Sistema de la empresa Logiciel Cía. Ltda.

c) Estructura del sistema LogiHistoricos: mostrada en la Figura 3.28.

Arquitectura: Modelo Vista Controlador

Lenguaje de programación: Kendo

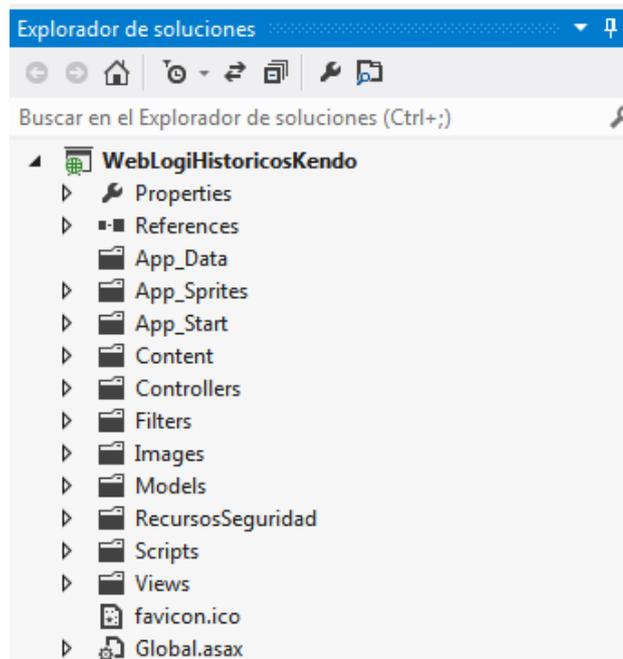


Figura 3.28 Solución del sistema LogiHistoricos

Fuente: Sistema de la empresa Logiciel Cía. Ltda.

d) Gestores que contiene el sistema LogiHistoricos: mostrados en la Figura 3.29.

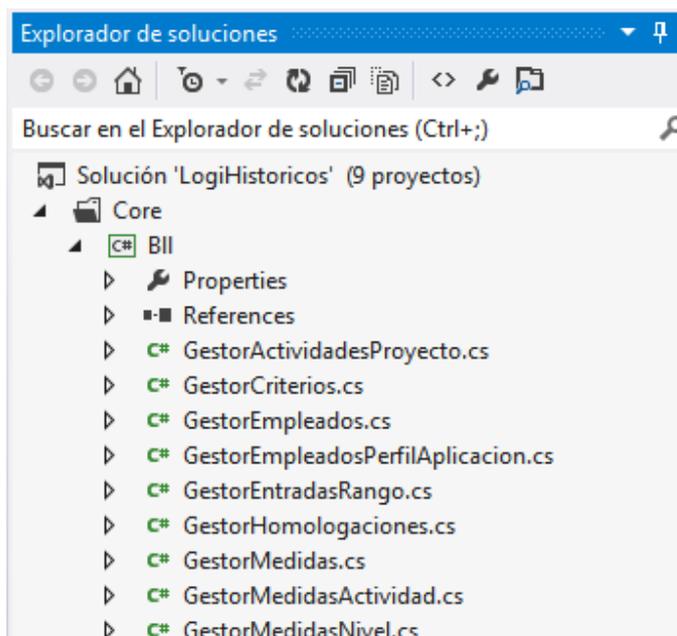


Figura 3.29 Gestores que contiene el sistema LogiHistoricos

Fuente: Sistema de la empresa Logiciel Cía. Ltda.

- e) Controles que contiene el sistema LogiHistoricos: mostrados en la Figura 3.30.

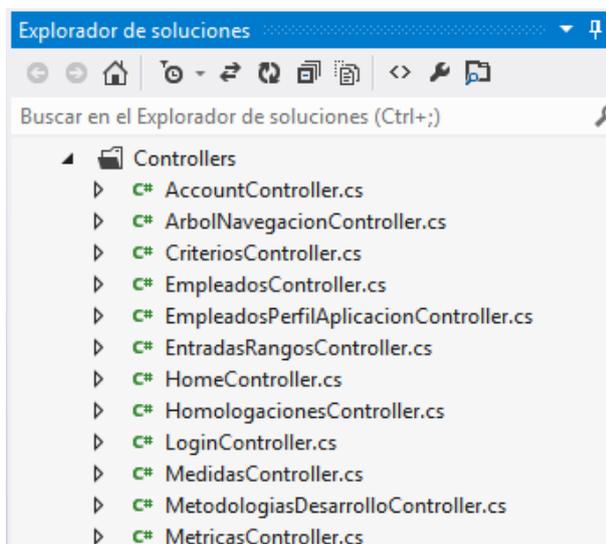


Figura 3.30 Controles que contiene el sistema LogiHistoricos

Fuente: Sistema de la empresa Logiciel Cía. Ltda.

- f) Vistas que contiene el sistema LogiHistoricos: mostradas en la Figura 3.31.

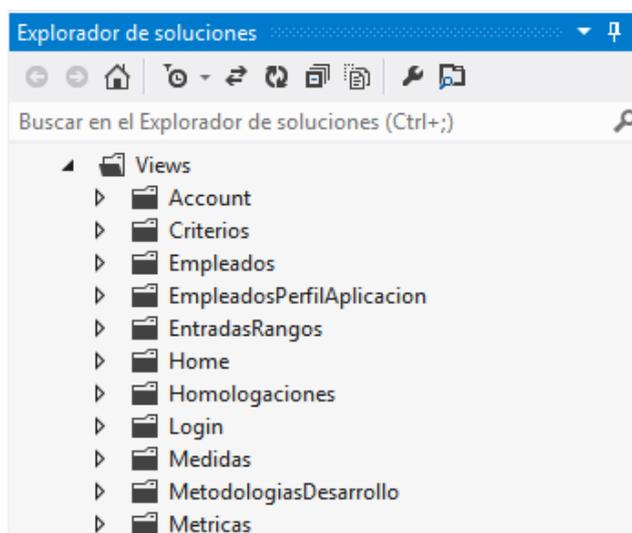


Figura 3.31 Vistas que contiene el sistema LogiHistoricos

Fuente: Sistema de la empresa Logiciel Cía. Ltda.

Implementación de los cambios en el código fuente de acuerdo a lo planificado en el plan de implementación:

a) Implementar la pantalla y la funcionalidad para generar el reporte de las métricas por proyecto.

✓ Control creado: mostrado en la Figura 3.32.

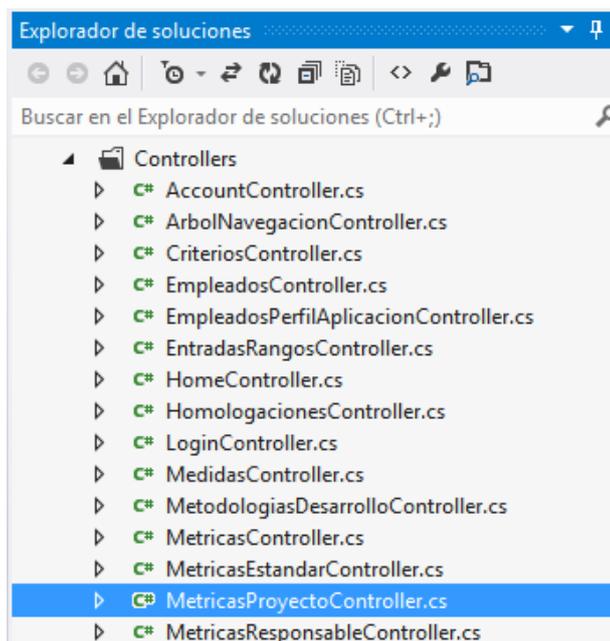


Figura 3.32 Control de la pantalla Métricas por proyecto  
Fuente: Sistema de la empresa Logiciel Cía. Ltda.

✓ En la Figura 3.33 se muestra un fragmento del código creado para obtener las métricas por proyecto.

```
namespace LogiHistoricos.WebLogiHistoricosKendo.Controllers
{
    public class MetricasProyectoController : Controller
    {
        public ActionResult ObtenerResumenProyectos([DataSourceRequest]
        DataSourceRequest request)
        {
            try
            {
                var resumenProyectos =
                GestorProyectos.ObtenerResumenProyectos().Where(p
                =>!p.CodigoEstado.Equals(ConfigurationManager.AppSettings["CodigoE
                stadoProyectoEliminado"])).ToList();
                return Json(resumenProyectos.ToDataSourceResult(request),
```

```

        JsonRequestBehavior.AllowGet);
    }
    catch (Exception exception)
    {
        ExceptionPolicy.HandleException(exception, "UI Policy");
        Response.StatusCode = 1;
        return Content(Mensajero.DevolverMensaje("3021"));
    }
}
public ActionResult ObtenerMetricasProyecto([DataSourceRequest]
DataSourceRequest request, string codigoProyecto)
{
    try
    {
        var metricasProyecto =
        GestorMetricasNivel.ObtenerMetricasNivel(codigoProyecto);
        return Json(metricasProyecto.ToDataSourceResult(request),
        JsonRequestBehavior.AllowGet);
    }
    catch (Exception exception)
    {
        ExceptionPolicy.HandleException(exception, "UI Policy");
        Response.StatusCode = 1;
        return Content(Mensajero.DevolverMensaje("3023"));
    }
}
}

```

Figura 3.33 Código generado para obtener las métricas por proyecto

Fuente: Sistema de la empresa Logiciel Cía. Ltda.

- ✓ Vista creada: mostrada en la Figura 3.34.

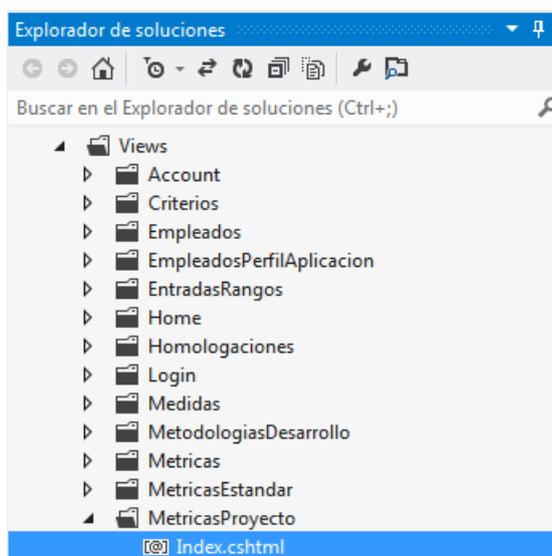


Figura 3.34 Vista de la pantalla Métricas por Proyecto

Fuente: Sistema de la empresa Logiciel Cía. Ltda.

- ✓ En la Figura 3.35 se muestra un fragmento del código realizado para crear los elementos de la interfaz de usuario de la pantalla métricas por proyecto.

```

@using LogiHistoricos.Entidades.LogiHistoricosDesigner
@using LogiHistoricos.Recursos
@{
    Layout = null;
    var mensaje = Mensajero.DevolverMensaje("3024");
}
<script type="text/javascript">
    var filtroMetricasProyecto = function () {
        var codigoProyecto =
            $('#ddlProyectosMetricasP').data('kendoDropDownList').value();

        return {
            codigoProyecto: codigoProyecto
        };
    };
</script>
<div class="k-content">
    @using (Html.BeginForm("Index", "MetricasProyecto"))
    {
        @(Html.Kendo().Grid<ResumenProyectosInfo>()
            .Name("grdResumenProyectos")
            .Columns(columns =>
                {
                    columns.Bound(p => p.NombreProyecto).Title("Proyecto");
                    columns.Bound(p => p.LiderProyecto).Title("Líder de Proyecto");
                    columns.Bound(p => p.Responsables).Title("Responsables");
                    columns.Bound(p => p.Estado).Title("Estado");
                    columns.Bound(p => p.FechaInicio).ClientTemplate("#= kendo.toString.FechaInicio, 'dd/MM/yyyy' #");
                    columns.Bound(p => p.FechaFin).ClientTemplate("#= kendo.toString.FechaFin, 'dd/MM/yyyy' #");
                })
            .Pageable(page => page.Enabled(true).Refresh(true))
            .Sortable()
            .Scrollable()
            .DataSource(dataSource =>dataSource
                .Ajax()
                .PageSize(20)
                .Events(events => events.Error("error_handler"))
                .Model(model => model.Id(p => p.CodigoProyecto))
                .Read(read => read

                    .Action("ObtenerResumenProyectos", "MetricasProyecto")))
        )
    }

```

Figura 3.35 Código para crear los elementos de la pantalla métricas por proyecto

Fuente: Sistema de la empresa Logiciel Cía. Ltda.

b) Implementar la pantalla y la funcionalidad para generar el reporte de las métricas por responsable.

✓ Gestor creado: mostrado en la Figura 3.36.

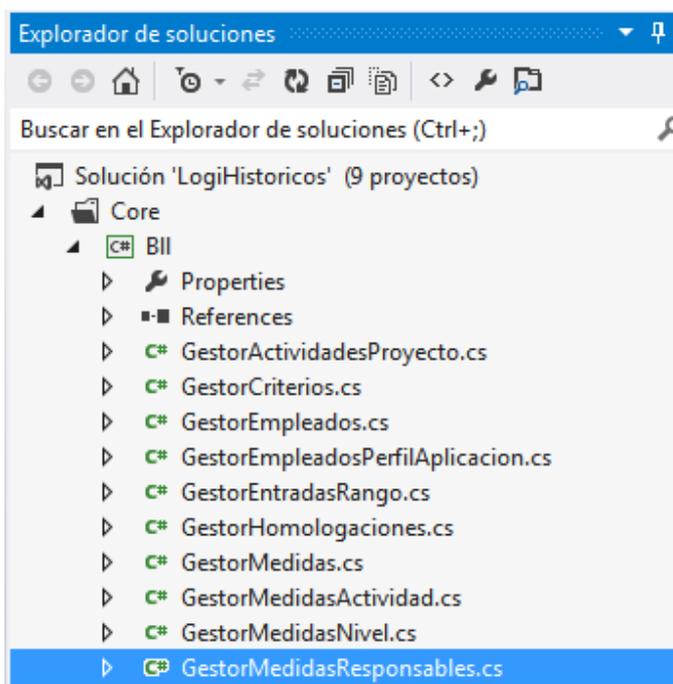


Figura 3.36 Gestor de la pantalla Métricas por Responsable

Fuente: Sistema de la empresa Logiciel Cía. Ltda.

✓ En la Figura 3.37 se muestra un fragmento del código creado para calcular las métricas por responsable.

```
namespace LogiHistoricos.Bll
{
    public class GestorMetricasResponsable
    {
        public static IList<MetricasResponsableInfo>
        ObtenerMetricasResponsable(string codigoProyecto, short
        codigoResponsable)
        {
            var contexto = new LogiHistoricosTablasEntities();
            var repositorioResponsables = new RESPONSABLESRepository {
            UnitOfWork = new EFUnitOfWork() };
            var responsablesProyecto =
            repositorioResponsables.ObtenerResponsablesProyecto(codigoProyecto)
            .Where(res => res.CodigoEmpleado == codigoResponsable).ToList();
            IList<MetricasResponsableInfo> listaMetricasResponsable = new
            List<MetricasResponsableInfo>();
            IList<METRICAS_REF> metricas = contexto.METRICAS_REF
            .Where(met => met.ACTIVO.Equals(true)).ToList();
        }
    }
}
```

```

foreach (var responsable in responsablesProyecto)
{
    try
    {
        foreach (var metrica in metricas)
        {
            var codigosRango = contexto.RANGOS_INDICADOR.Where(rng =>
            rng.MET_CODIGO == metrica.CODIGO)
            .Select(s => s.RNG_CODIGO).Distinct();
            var entradasRango = new List<ENTRADAS_RANGO>();
            foreach (var rango in codigosRango)
            {
                var secuencias = contexto.RANGOS_INDICADOR.Where(rng
                =>rng.MET_CODIGO == metrica.CODIGO
                && rng.RNG_CODIGO.Equals(rango))
                .Select(s => s.ERN_SECUENCIA).ToList();

                foreach (var secuencia in secuencias)
                {
                    var entrada =
                    contexto.ENTRADAS_RANGO.FirstOrDefault
                    (ern => ern.SECUENCIA == secuencia &
                    ern.RNG_CODIGO .Equals(rango));
                }
            }
        }
    }
}

```

Figura 3.37 Código creado para calcular las métricas por responsable

Fuente: Sistema de la empresa Logiciel Cía. Ltda.

- ✓ Control creado: mostrado en la Figura 3.38.

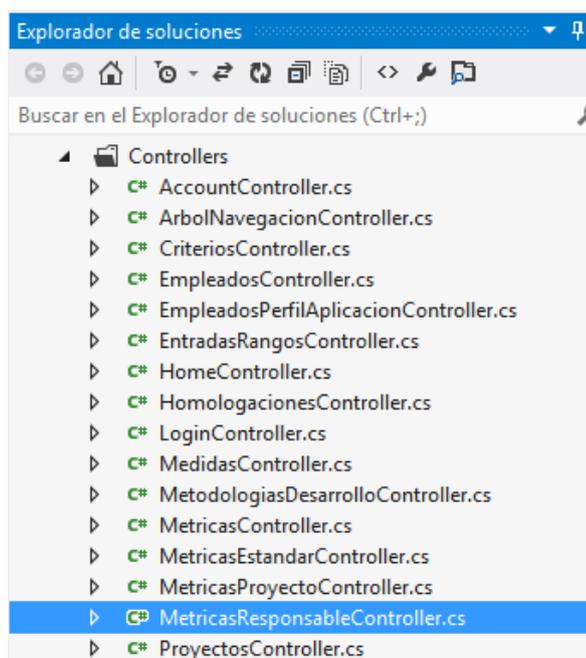


Figura 3.38 Control de la pantalla Métricas por Responsable

Fuente: Sistema de la empresa Logiciel Cía. Ltda.

- ✓ En la Figura 3.39 se muestra un fragmento del código creado para obtener las métricas por responsable.

```

namespace LogiHistoricos.WebLogiHistoricosKendo.Controllers
{
    public class MetricasResponsableController : Controller
    {
        public JsonResult ObtenerProyectos()
        {
            try
            {
                var proyectos = GestorProyectos.ObtenerProyectos().Where(p =>
                    p.CodigoEstado== ConfigurationManager.AppSettings
                    ["CodigoEstadoProyectoCerrado"]).ToList();
                return Json(proyectos, JsonRequestBehavior.AllowGet);
            }
            catch (Exception exception)
            {
                ExceptionPolicy.HandleException(exception, "UI Policy");
                ModelState.AddModelError(exception.Message + "|dw|CriteriosT",
                    Mensajero.DevolverMensaje("1087"));
                return Json(ModelState.ToDataSourceResult());
            }
        }
        /// Función para obtener los responsables de los proyectos
        public JsonResult ObtenerResponsablesProyecto(string codigoProyecto)
        {
            try
            {
                var responsablesProyecto =
                    (string.IsNullOrEmpty(codigoProyecto)) ? new
                    List<ResponsablesInfo>() : GestorResponsables.
                    ObtenerResponsablesProyecto(codigoProyecto);
                return Json(responsablesProyecto, JsonRequestBehavior.AllowGet);
            }
            catch (Exception exception)
            {
                ExceptionPolicy.HandleException(exception, "UI Policy");
                ModelState.AddModelError(exception.Message + "|dw|CriteriosT",
                    Mensajero.DevolverMensaje("1087"));
                return Json(ModelState.ToDataSourceResult());
            }
        }
        // Método para obtener todas la métricas de los responsables
        public ActionResult ObtenerMetricasResponsables([DataSourceRequest]
            DataSourceRequest request, string codigoProyecto, string
            codigoResponsable)
        {
            try
            {
                var metricasResponsables =
                    (string.IsNullOrEmpty(codigoResponsable)) ? new
                    List<MetricasResponsableInfo>() :
                    GestorMetricasResponsable.ObtenerMetricasResponsable(codigoProyec
                    to, Convert.ToInt16(codigoResponsable));
            }
        }
    }
}

```

Figura 3.39 Código generado para obtener las métricas por responsable

Fuente: Sistema de la empresa Logiciel Cía. Ltda.

- ✓ Vista creada: mostrada en la Figura 3.40.

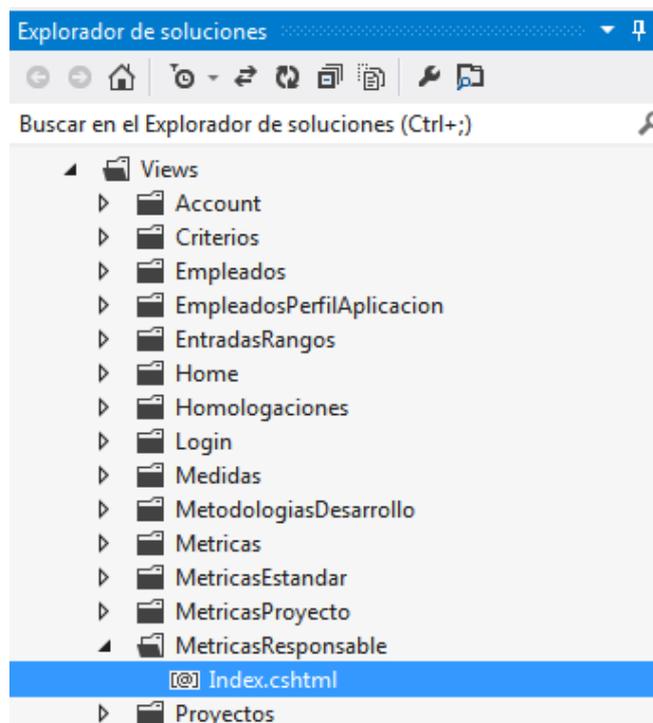


Figura 3.40 Vista de la pantalla Métricas por Responsable

Fuente: Sistema de la empresa Logiciel Cía. Ltda.

- ✓ En la Figura 3.41 se muestra un fragmento del código realizado para crear los elementos de la interfaz de usuario de la pantalla métricas por responsable.

```
@using LogiHistoricos.Entidades.LogiHistoricosDesigner
@using LogiHistoricos.Recursos
@{
    Layout = null;
    var mensajeExportar = Mensajero.DevolverMensaje("3024");
}
<script type="text/javascript">
    var filtroCodigoProyectoData = function () {
        var comboProyectos = $('#dwlProyecto').data('kendoDropDownList');
        return {
            codigoProyecto: comboProyectos.value()
        };
    };
    var metricasResponsableData = function () {
        var comboProyecto = $('#dwlProyecto').data('kendoDropDownList');
        var comboResponsable = $('#dwlResponsable').data('kendoDropDownList');
        return {
            codigoProyecto: comboProyecto.value(),
            codigoResponsable: comboResponsable.value()
        };
    };
</script>
```

```

};
@( Html.Kendo().DropDownList()
    .Name("dwlResponsable")
    .HtmlAttributes(new { style = "width:200px" })
    .OptionLabel("Seleccione un responsable...")
    .DataValueField("CodigoEmpleado")
    .DataTextField("NombreEmpleado")
    .AutoBind(false)
    .DataSource(source => source.Events(ev => ev.Error("error_handler")))
    .Read(read => read.Action("ObtenerResponsablesProyecto",
        "MetricasResponsable")
    .Data("filtroCodigoProyectoData"))
    .ServerFiltering(true))
    .CascadeFrom("dwlProyecto"))
    </td>
</tr>
</table>
</div>

```

Figura 3.41 Código para crear los elementos de la pantalla métricas por responsable

Fuente: Sistema de la empresa Logiciel Cía. Ltda.

c) Implementar la pantalla y la funcionalidad para generar el reporte de las métricas estándar.

✓ Control creado: mostrado en la Figura 3.42.

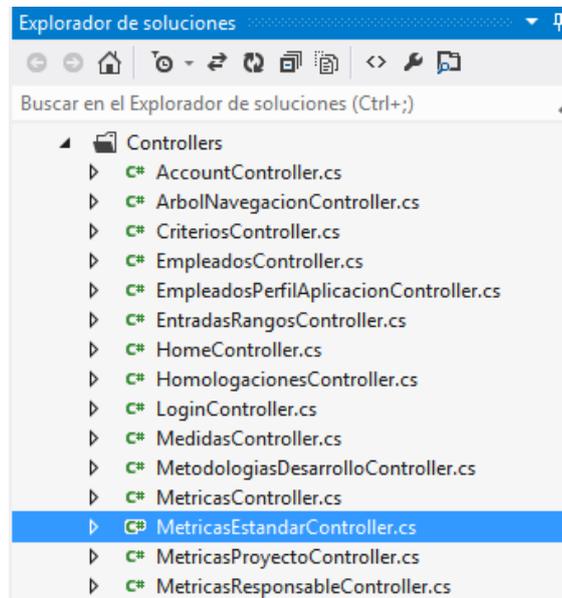


Figura 3.42 Control de la pantalla Métricas Estándar

Fuente: Sistema de la empresa Logiciel Cía. Ltda.

- ✓ En la Figura 3.43 se muestra un fragmento del código creado para obtener las métricas estándar.

```

namespace LogiHistoricos.WebLogiHistoricosKendo.Controllers
{
    public class MetricasEstandarController : Controller
    {
        // Método para obtener todas las métricas estándar
        public ActionResult ObtenerMetricasEstandar([DataSourceRequest]
        DataSourceRequest request, string codigoMetodologia, string
        codigoPlataforma, string codigoCriterio)
        {
            try
            {
                var metricasEstandar =
                GestorMetricasEstandar.ObtenerMetricasEstandar (codigoMetodologia,
                codigoPlataforma,codigoCriterio);
                Session[VariablesSesionLHS.METRICAS_ESTANDAR_MES] =
                metricasEstandar;

                if (!string.IsNullOrEmpty(codigoMetodologia))

                    Session[VariablesSesionLHS.METODOLOGIA_METRICAS_ESTANDAR_MES] =
                    GestorMetodologiasDesarrollo.ObtenerMetodologiaPorId(codigoMetod
                    ologia).DESCRIPCION;
                    var tablasReferencialesInfo =
                    GestorTablasReferenciales.ObtenerDatosTablaReferencial("CRITERIO
                    S_TAMANIO_REF").FirstOrDefault(m => m.CodigoDatoTabla ==
                    codigoCriterio);
                    if (tablasReferencialesInfo != null)
                    {
                        Session[VariablesSesionLHS.CRITERIO_METRICAS_ESTANDAR_MES] =
                        tablasReferencialesInfo.DescripcionDatoTabla;
                    }

                    var tablasReferenciales =
                    GestorTablasReferenciales.ObtenerDatosTablaReferencial("PLATAFOR
                    MAS_DESARROLLO_REF").FirstOrDefault(m => m.CodigoDatoTabla ==
                    codigoPlataforma);

                    if (tablasReferenciales != null)
                    {
                        Session[VariablesSesionLHS.PLATAFORMA_METRICAS_ESTANDAR_MES]
                        = tablasReferenciales.DescripcionDatoTabla;
                    }
                    return Json(metricasEstandar.ToDataSourceResult(request),
                    JsonRequestBehavior.AllowGet);
                }
            }
            catch (Exception exception)
            {
                ExceptionPolicy.HandleException(exception, "UI Policy");
                Response.StatusCode = 1;
                return Content(Mensajero.DevolverMensaje("5021"));
            }
        }
    }
}

```

Figura 3.43 Código generado para obtener las métricas estándar

Fuente: Sistema de la empresa Logiciel Cía. Ltda.

- ✓ Vista creada: mostrada en la Figura 3.44.

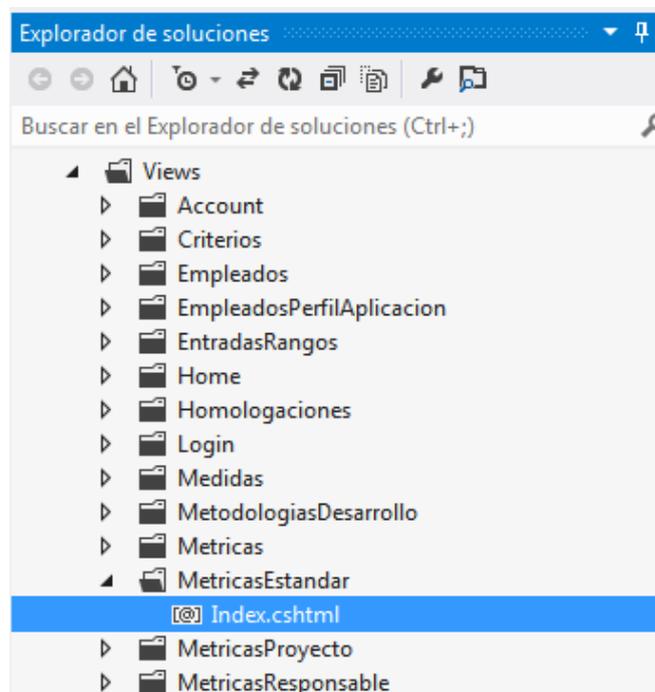


Figura 3.44 Vista de la pantalla Métricas Estándar  
Fuente: Sistema de la empresa Logiciel Cía. Ltda.

- ✓ En la Figura 3.45 se muestra un fragmento del código realizado para crear los elementos de la interfaz de usuario de la pantalla métricas estándar.

```

@using LogiHistoricos.Entidades.LogiHistoricosDesigner
@using LogiHistoricos.Recursos
@{
    Layout = null;
    var mensajeExportar = Mensajero.DevolverMensaje("3024");
}
<script type="text/javascript">
    var filtroMetricasEstandar = function () {
        var codigoMetodologia =
            $('#ddlMetodologiasDesarrolloME').data('kendoDropDownList').value();
        var codigoPlataforma =
            $('#ddlPlataformasME').data('kendoDropDownList').value();
        var codigoCriterio =
            $('#ddlCriteriosTamanoME').data('kendoDropDownList').value();
        return {
            codigoMetodologia: codigoMetodologia,
            codigoPlataforma: codigoPlataforma,
            codigoCriterio: codigoCriterio
        };
    };
    @(Html.Kendo().DropDownList()
        .Name("ddlMetodologiasDesarrolloME")

```

```

.OptionLabel("Seleccione una opción")
.SelectedIndex(0)
.BindTo(new
SelectList((System.Collections.IEnumerable)TempData[VariablesSesionLHS.L
ISTA_METODOLOGIAS_MES], "Codigo", "Descripcion"))
.HtmlAttributes(new {style="width:200px;vertical-
align:middle;font:menu;" })
</td>
</tr>
<tr>
<td>
Plataforma:
</td>
@(Html.Kendo().DropDownList()
.Name("ddlPlataformasME")
.OptionLabel("Seleccione una opción")
.SelectedIndex(0)
.BindTo(new SelectList((System.Collections.IEnumerable)TempData
[VariablesSesionLHS.LISTA_PLATAFORMAS_MES], "CodigoDatoTabla",
"DescripcionDatoTabla"))
.HtmlAttributes(new {style="width:200px;vertical-
align:middle;font:menu;" }))

```

Figura 3.45 Código para crear los elementos de la pantalla métricas estándar

Fuente: Sistema de la empresa Logiciel Cía. Ltda.

Una vez terminada la implementación, se puede observar como se ve la pantalla principal de la aplicación LogiHistoricos, apareciendo las nuevas opciones que se implementaron en el menú principal de la aplicación.

a) Pantalla principal del sistema: mostrada en la Figura 3.46.

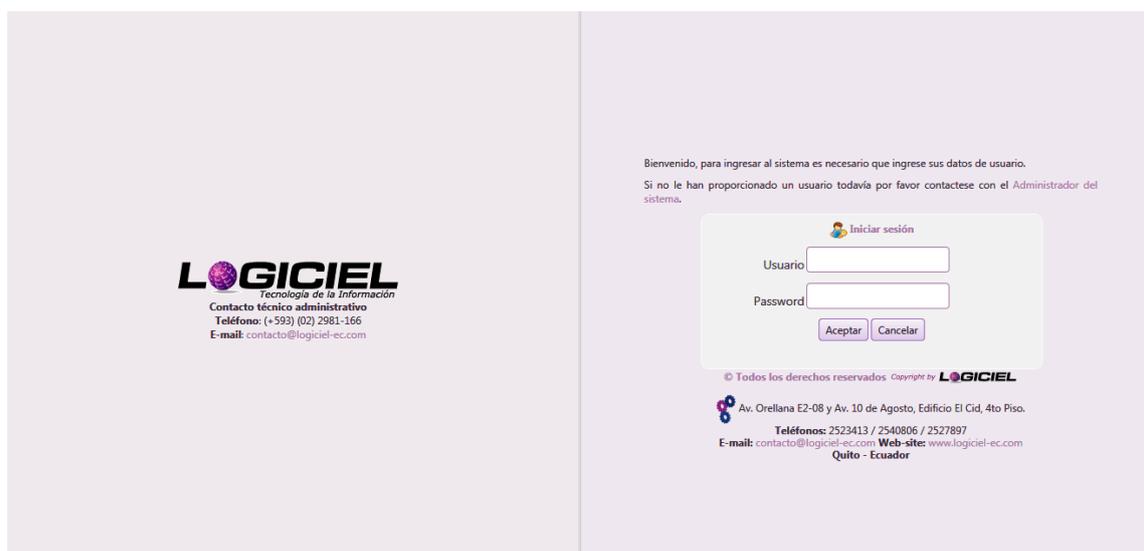


Figura 3.46 Pantalla principal de LogiHistoricos

Fuente: Sistema de la empresa Logiciel Cía. Ltda.

b) Menú principal del sistema: mostrado en la Figura 3.47.

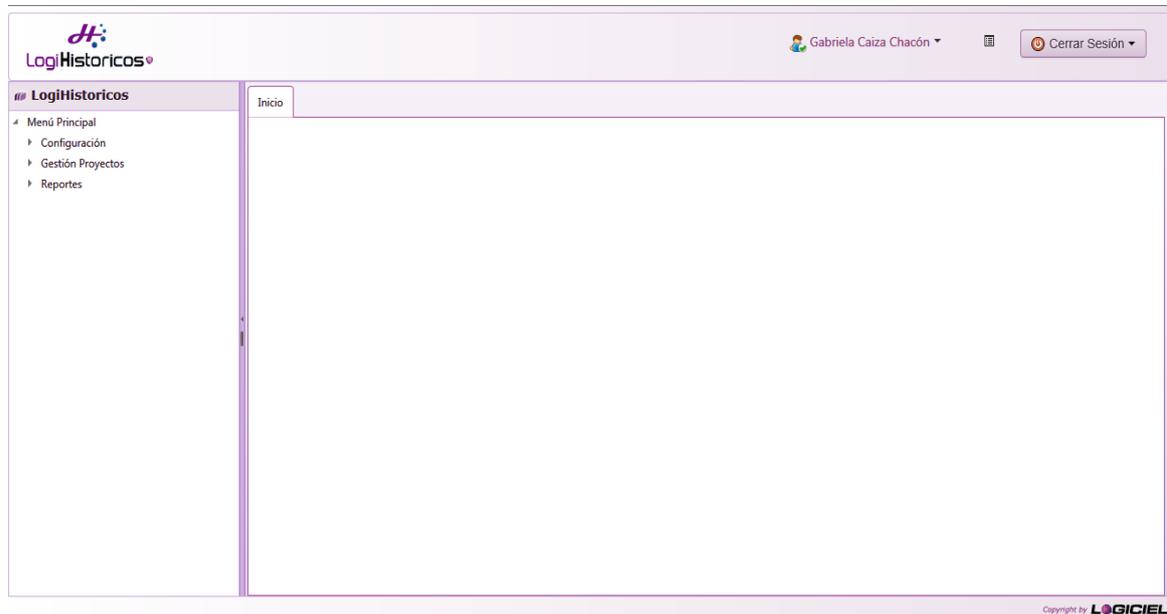


Figura 3.47 Menú principal de LogiHistoricos después del cambio

Fuente: Sistema de la empresa Logiciel Cía. Ltda.

c) Menú de la opción Reportes: mostrado en la Figura 3.48.

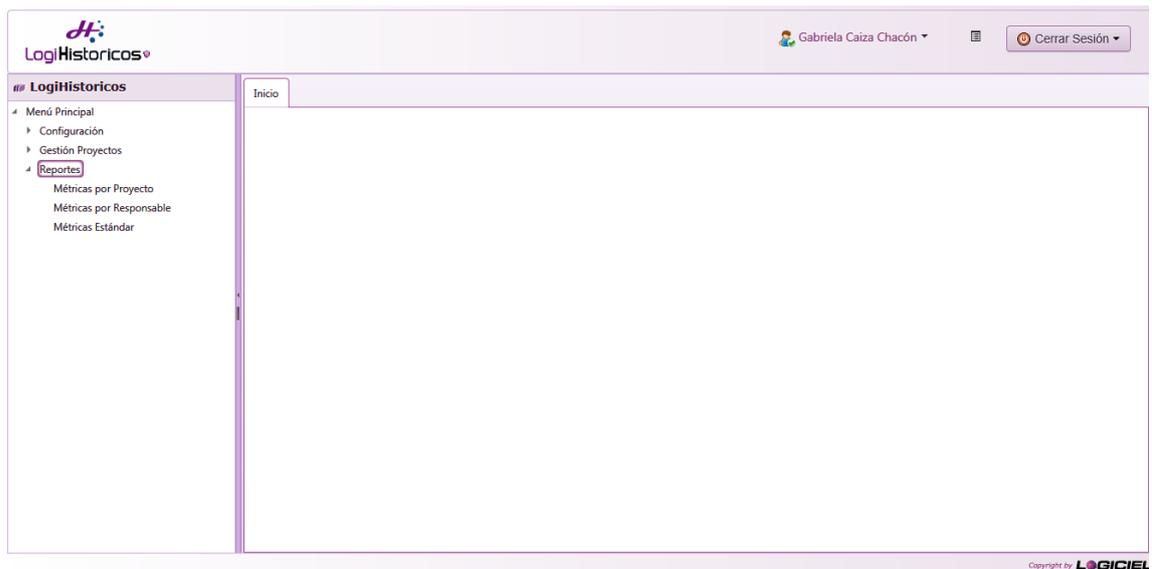


Figura 3.48 Menú de la opción Reportes

Fuente: Sistema de la empresa Logiciel Cía. Ltda.

d) Pantalla Métricas por Proyecto: mostrado en la Figura 3.49.

The screenshot shows the 'Métricas por Proyecto' screen in the LogiHistoricos application. The table displays the following data:

Proyecto	Líder de Proyecto	Responsables	Estado	Fecha Inicio	Fecha Fin
PRV_Repositorio de medidas y métricas	Isabel Vasco Aguas	Maria Eugenia Vines   Pablo Albuja   Gabriela Caiza Chacón   Claudia Arcos Obando	Cerrado	08/01/2013	18/02/2013
PRV_LogiGenDocsFasell	Santiago Falconi	Maria Eugenia Vines   Pablo Albuja   Gabriela Caiza Chacón   David Campaña Noboa   Cristian Pérez	Abierto	10/06/2013	26/12/2013
MNT_COOPROGRESO_20...	Nadia Karina Jaramillo	Isabel Vasco Aguas   Maria Eugenia Vines   Victor Aguilar   Andrea Guzman   Nestor Silverio   Santiago Falconi   Angel Cevallos   David Campaña Noboa   Edison Salas   Geovanna Aguinaga Villamar	Cerrado	18/03/2013	10/12/2013

At the bottom of the table, it indicates 'Elementos mostrados 1 - 6 de 6'.

**Figura 3.49 Pantalla de Métricas por Proyecto**  
Fuente: Sistema de la empresa Logiciel Cía. Ltda.

e) Pantalla Métricas por Responsable: mostrado en la Figura 3.50.

The screenshot shows the 'Métricas por Responsable' screen in the LogiHistoricos application. The table is currently empty, and the status at the bottom right indicates 'No hay datos.'.

Métrica	Valor	Indicador
No hay datos.		

**Figura 3.50 Pantalla de Métricas por Responsable**  
Fuente: Sistema de la empresa Logiciel Cía. Ltda.

f) Pantalla Métricas Estándar: mostrado en la Figura 3.51.

The screenshot displays the 'LogiHistoricos' web application interface. The top navigation bar shows the user 'Gabriela Caiza Chacón' and a 'Cerrar Sesión' button. The sidebar menu on the left includes 'Menú Principal', 'Configuración', 'Gestión Proyectos', and 'Reportes', with 'Métricas Estándar' selected under 'Reportes'. The main content area is titled 'Inicio Métricas Estándar' and features an 'Exportar' button, a 'Salir' button, and a 'Filtros' section with three dropdown menus: 'Metodología', 'Plataforma', and 'Criterio Tamaño', each with the text 'Seleccione una opción'. Below the filters is a table with three columns: 'Métrica', 'Valor Estándar', and 'Indicador'. The table is currently empty, and the status 'No hay datos.' is displayed at the bottom right. A pagination bar at the bottom left shows '0' items.

**Figura 3.51 Pantalla de Métricas Estándar**  
Fuente: Sistema de la empresa Logiciel Cía. Ltda.

18. El Equipo de mantenimiento se encarga de actualizar el Listado de elementos a modificar mostrado en la Figura 3.52.

							
DATOS GENERALES DEL PROYECTO O SISTEMA							
Nombre del Proyecto	LogiHistoricos						
Líder del Proyecto	Isabel Vasco						
Equipo de Mantenimiento	María Vincés, Gabriela Caiza, Marcela Balseca						
LISTADO DE ELEMENTOS A MODIFICAR							
No.	ELEMENTO A MODIFICAR	CAPA DE LA ARQUITECTURA	DESCRIPCIÓN DEL CAMBIO A IMPLEMENTAR	RESPONSABLE DE LA MODIFICACIÓN	DURACIÓN	ESTADO DE LA MODIFICACIÓN	OBSERVACIONES
1	Sistema	VISTA	Crear los elementos de la interfaz de usuario para las pantallas. Implementar tres pantallas una para cada reporte; los nombres de cada una de las pantallas serán: *Métricas por proyecto *Métricas por responsable *Métricas estándar	Analista-programador	8h	Terminado	
2		CONTROLADOR	Crear un control para cada reporte. Implementar las acciones que devuelven los datos para la vista.	Analista-programador	4h	Terminado	
3		GESTOR	Realizar el cálculo de las métricas por responsable.	Analista-programador	4h	Terminado	

Figura 3.52 Listado de elementos a modificar actualizado (PLANTILLA 12).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.

19. El Equipo de mantenimiento se encarga de actualizar el Listado de documentos a modificar mostrado en la Figura 3.53.

								
DATOS GENERALES DEL PROYECTO O SISTEMA								
Nombre del Proyecto	LogiHistoricos							
Líder del Proyecto	Isabel Vasco							
Equipo de Mantenimiento	María Vincés, Gabriela Caiza, Marcela Balseca							
LISTADO DE DOCUMENTOS A MODIFICAR								
FASE	No.	ID - DOCUMENTO	DESCRIPCIÓN	VERSIÓN	RESPONSABLE DE LA MODIFICACIÓN	DURACIÓN	ESTADO DE LA ACTUALIZACIÓN	OBSERVACIONES
Gestión de Requerimientos	1	LHS_ESP_FUN	Especificación de requerimientos	2.0	Documentador	1h	Terminado	
	2	LHS_FOR_GRQ	Gestión de requerimientos	2,0	Documentador	1h	Terminado	
Análisis y Diseño	3	LHS_ESP_CUS	Especificación de casos de uso	1,0	Documentador	3h	Terminado	
	4	LHS_DGA_PRY	Diagramas de casos de uso	1,0	Documentador	3h	Terminado	
Implementación	5	LHS_MAN_USR	Manual de usuario	1,0	Documentador	8h	Terminado	

Figura 3.53 Listado de documentos a modificar actualizado (PLANTILLA 15).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.

20. El Responsable de mantenimiento se encarga de actualizar el Registro del estudio de la solicitud de modificación mostrado en la Figura 3.54.

LogiHistoricos											
DATOS GENERALES DEL PROYECTO O SISTEMA											
Nombre del Proyecto	LogiHistoricos										
Líder del Proyecto	Isabel Vasco										
Gestor de Peticiones	Marcela Balseca										
REGISTRO DEL ESTUDIO DE LA SOLICITUD DE MODIFICACIÓN											
ID. SOLICITUD	DESCRIPCIÓN O DETALLE	RESPONSABLE DEL MANTENIMIENTO	FECHA RECEPCIÓN	FECHA APROBACIÓN	FECHA CIERRE	TIPO DE MANTENIMIENTO	MOTIVO DE SELECCIÓN	PRIORIDAD	ESTADO DEL CAMBIO	MOTIVO DEL RECHAZO	ESTADO DE IMPLEMENTACIÓN
SM01	Generar un reporte de las métricas por proyecto y exportarlo a Excel.	Gabriela Caiza	02/06/2014	03/06/2014		Perfectivo	Seleccionamos este tipo de mantenimiento porque se va añadir nuevos requerimientos al sistema existente.	Medio	Aprobado		En ejecución
	Calcular y generar un reporte de las métricas por responsable y exportarlo a Excel.										
	Generar un reporte de las métricas estándar y exportarlo a Excel.										

Figura 3.54 Registro del estudio de la solicitud de modificación actualizado (PLANTILLA 2).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.

## FASE 5: Pruebas del sistema

21. Los tester asignados se encargan de actualizar los resultados de los Casos de prueba de funcionalidad: mostrados en la Figuras 3.55, 3.56 y 3.57.



DATOS DEL CASO DE PRUEBA								
Nombre del Proyecto	LogiHistoricos							
Id. Requerimiento	RN1							
Id. Caso de prueba	CP01							
Tipo de prueba	Funcionalidad							
Módulo a probar	Registro de Medidas							
Autor del caso de prueba	Analista programador/ Tester							
Nombre del tester	Gabriela Caiza							
Objetivo	Probar la funcionalidad de la pantalla de métricas por proyecto.							
Descripción	Validar la funcionalidad de la pantalla métricas por proyecto.							
Fecha de creación	05/06/2014	Fecha de ejecución:	10/06/2014					
EJECUCIÓN Y RESULTADOS DEL CASO DE PRUEBA:								
ID.	ELEMENTO A PROBAR	PRECONDICIÓN	DATOS DE ENTRADA	RESULTADO ESPERADO	RESULTADO OBTENIDO	DOCUMENTOS DE RESPALDO	ESTADO DEL CASO DE PRUEBA	FECHA DE APROBACIÓN
1	Métricas por proyecto	Disponibilidad de la aplicación y de la pantalla mencionada.	Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Proyecto" 4.-Seleccionar del filtro Proyecto:PRY_Proceso de atención de incidencias	Se visualice en pantalla las métricas existentes del proyecto PRY_Proceso de atención de incidencias. En la pantalla se pueda ver el valor de las métrica y los indicadores.	Pendiente	LHS_DOC-RDP_20140610	Aprobado	11/06/2014
					Exitoso	LHS_DOC-RDP_20140611		
2			Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Proyecto" 4.-Seleccionar del filtro Proyecto:PRY_Proceso de atención de incidencias 5.-Presionar el botón Exportar	Se genera el reporte en Excel con las métricas existentes del proyecto PRY_Proceso de atención de incidencias. En el reporte se presenta el valor de las métricas, los indicadores y las observaciones.	Fallido	LHS_DOC-RDP_20140610		
				Exitoso	LHS_DOC-RDP_20140611			
3			Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Proyecto" 4.-Seleccionar del filtro Proyecto:PRY_Proceso de atención de incidencias 5.-Presionar el botón Salir	Se cierra la pantalla de Métricas por Proyecto.	Exitoso	LHS_DOC-RDP_20140610		

Figura 3.55 Resultado del Caso de Prueba 01 (PLANTILLA 16-CP01).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.



DATOS DEL CASO DE PRUEBA		
Nombre del Proyecto	LogiHistoricos	
Id. Requerimiento	RN2	
Id. Caso de prueba	CP02	
Tipo de prueba	Funcionalidad	
Módulo a probar	Registro de Medidas	
Autor del caso de prueba	Analista programador/ Tester	
Nombre del tester	Gabriela Caiza	
Objetivo	Probar la funcionalidad de la pantalla de métricas por responsable.	
Descripción	Validar la funcionalidad de la pantalla métricas por responsable.	
Fecha de creación	05/06/2014	Fecha de ejecución: 11/06/2014

#### EJECUCIÓN Y RESULTADOS DEL CASO DE PRUEBA:

ID.	ELEMENTO A PROBAR	PRECONDICIÓN	DATOS DE ENTRADA	RESULTADO ESPERADO	RESULTADO OBTENIDO	DOCUMENTOS DE RESPALDO	ESTADO DEL CASO DE PRUEBA	FECHA DE APROBACIÓN
1			Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Responsable" 4.-Seleccionar del filtro Proyecto: PRY_Proceso de atención de incidencias 5.-Seleccionar del filtro Responsable: Isabel Vasco Aguas.	Se visualice en pantalla las métricas existentes del proyecto PRY_Proceso de atención de incidencias del responsable Isabel Vasco Aguas . En la pantalla se pueda ver el valor de las métrica y los indicadores.	Fallido	LHS_DOC-RDP_20140611		
					Exitoso	LHS_DOC-RDP_20140612		
2	Métricas por Responsable	Disponibilidad de la aplicación y de la pantalla mencionada.	Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Responsable" 4.-Seleccionar del filtro Proyecto: PRY_Proceso de atención de incidencias 5.-Seleccionar del filtro Responsable: Isabel Vasco Aguas. 6.-Presionar el botón Exportar	Se generará el reporte en excel con las métricas existentes del proyecto PRY_Proceso de atención de incidencias del responsable Isabel Vasco Aguas. En el reporte el valor de las métricas y los indicadores.	Fallido	LHS_DOC-RDP_20140611	Aprobado	12/06/2014
					Exitoso	LHS_DOC-RDP_20140612		
3			Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Responsable" 4.-Seleccionar del filtro Proyecto: PRY_Proceso de atención de incidencias 5.-Seleccionar del filtro Responsable: Isabel Vasco Aguas. 6.-Presionar el botón Salir	Se cierra la pantalla de Métricas por Responsable.	Exitoso	LHS_DOC-RDP_20140611		

**Figura 3.56 Resultados del Caso de Prueba 02 (PLANTILLA 16-CP02).**

**Elaborado por:** Balseca Marcela, Caiza Gabriela

**Fuente:** Información obtenida de la Empresa Logiciel Cía. Ltda.



DATOS DEL CASO DE PRUEBA			
Nombre del Proyecto	LogiHistoricos		
Id. Requerimiento	RN3		
Id. Caso de prueba	CP03		
Tipo de prueba	Funcionalidad		
Módulo a probar	Registro de Medidas		
Autor del caso de prueba	Analista programador/ Tester		
Nombre del tester	Gabriela Caiza		
Objetivo	Probar la funcionalidad de la pantalla de métricas estándar.		
Descripción	Validar la funcionalidad de la pantalla métricas estándar.		
Fecha de creación	05/06/2014	Fecha de ejecución:	12/06/2014
EJECUCIÓN Y RESULTADOS DEL CASO DE PRUEBA:			

ID.	ELEMENTO A PROBAR	PRECONDICIÓN	DATOS DE ENTRADA	RESULTADO ESPERADO	RESULTADO OBTENIDO	DOCUMENTOS DE RESPALDO	ESTADO DEL CASO DE PRUEBA	FECHA DE APROBACIÓN
1	Métricas Estándar	Disponibilidad de la aplicación y de la pantalla mencionada.	Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas Estándar" 4.-Seleccionar del filtro Metodología:Proyectos Parametrización 5.-Seleccionar del filtro Plataforma: .Net 6.-Seleccionar del filtro Criterio Tamaño: PCO	Se visualice en pantalla las métricas estándar existentes para la metodología Proyectos Parametrización con plataforma .Net y criterio de tamaño PCO. En la pantalla se pueda ver el valor de las métrica y los indicadores.	Exitoso	LHS_DOC-RDP_20140612	Aprobado	12/06/2014
2			Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas Estándar" 3.- Seleccionar del filtro Metodología:Proyectos Parametrización 4.-Seleccionar del filtro Plataforma: .Net 5.-Seleccionar del filtro Criterio Tamaño: PCO 6.-Presionar el botón Exportar	Se genera el reporte en Excel con las métricas existentes del proyecto para la metodología Proyectos Parametrización con plataforma .Net y criterio de tamaño PCO. En el reporte se presenta el valor de las métricas y los indicadores.	Fallido	LHS_DOC-RDP_20140612		
3			Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas Estándar" 4.-Seleccionar del filtro Metodología:Proyectos Parametrización 5.-Seleccionar del filtro Plataforma: .Net 6.-Seleccionar del filtro Criterio Tamaño: PCO 7.-Presionar el botón Salir	Se cierra la pantalla de Métricas Estándar.	Exitoso	LHS_DOC-RDP_20140612		

Figura 3.57 Resultado del Caso de Prueba 03 (PLANTILLA 16-CP03).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.

22. Los tester asignados se encargan de actualizar los resultados de los Casos de prueba de rendimiento (Pruebas de estrés): mostrados en la Figuras 3.58, 3.59 y 3.60.

LogiHistoricos								
DATOS DEL CASO DE PRUEBA								
Nombre del Proyecto	LogiHistoricos							
Id. Requerimiento	RN1							
Id. Caso de prueba	CPR01							
Tipo de prueba	Estrés							
Módulo a probar	Registro de Medidas							
Autor del caso de prueba	Analista programador/ Tester							
Nombre del tester	Gabriela Caiza							
Objetivo	Probar la solidez de la aplicación en los momentos de carga.							
Descripción	Validar el rendimiento del sistema LogiHistoricos al generar varias reportes sobre las métricas por proyecto a la misma vez con diferentes usuarios.							
Fecha de creación	12/09/2014	Fecha de ejecución:	12/09/2014					
EJECUCIÓN Y RESULTADOS DEL CASO DE PRUEBA:								
ID.	ELEMENTO A PROBAR	PRECONDICIÓN	DATOS DE ENTRADA	RESULTADO ESPERADO	RESULTADO OBTENIDO	DOCUMENTOS DE RESPALDO	ESTADO DEL CASO DE PRUEBA	FECHA DE APROBACIÓN
1	Métricas por proyecto	Disponibilidad de la aplicación y de la pantalla mencionada.	Pasos a realizar usuario 1: 1.-Autenticarse en el sistema el usuario 1: ebalseca 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Proyecto" 4.-Seleccionar del filtro Proyecto:PRY_Proceso de atención de incidencias 5.-Presionar el botón Exportar	Se genera el reporte en Excel con las métricas existentes del proyecto PRY_Proceso de atención de incidencias.	Exitoso	LHS_DOC-RDP_20140912_EB	Aprobado	12/09/2014
2			Pasos a realizar usuario 2: 1.-Autenticarse en el sistema el usuario 2: esalas. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Proyecto" 4.-Seleccionar del filtro Proyecto:PRY_Repository de medidas y metricas 5.-Presionar el botón Exportar	Se genera el reporte en Excel con las métricas existentes del proyecto PRY_Proceso de atención de incidencias.	Exitoso	LHS_DOC-RDP_20140912_ES		
3			Pasos a realizar usuario 3: 1.-Autenticarse en el sistema el usuario 3: gcaiza. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Proyecto" 4.-Seleccionar del filtro Proyecto:PRY_Proceso de atención de incidencias 5.-Presionar el botón Exportar	Se genera el reporte en Excel con las métricas existentes del proyecto PRY_Proceso de atención de incidencias.	Exitoso	LHS_DOC-RDP_20140912_GC		
4			Pasos a realizar usuario 4: 1.-Autenticarse en el sistema el usuario 4: cperez. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Proyecto" 4.-Seleccionar del filtro Proyecto:PRY_Proceso de atención de incidencias 5.-Presionar el botón Exportar	Se genera el reporte en Excel con las métricas existentes del proyecto PRY_Proceso de atención de incidencias.	Exitoso	LHS_DOC-RDP_20140912_CP		

Figura 3.58 Resultado del Caso de Prueba de Estrés 01 (PLANTILLA 16-CPR01).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.



DATOS DEL CASO DE PRUEBA			
Nombre del Proyecto	LogiHistoricos		
Id. Requerimiento	RN2		
Id. Caso de prueba	CPR02		
Tipo de prueba	Estrés		
Módulo a probar	Registro de Medidas		
Autor del caso de prueba	Analista programador/ Tester		
Nombre del tester	Gabriela Caiza		
Objetivo	Probar la solidez de la aplicación en los momentos de carga.		
Descripción	Validar el rendimiento del sistema LogiHistoricos al generar varias reportes sobre las métricas estándar a la vez.		
Fecha de creación	12/09/2014	Fecha de ejecución:	12/09/2014

#### EJECUCIÓN Y RESULTADOS DEL CASO DE PRUEBA:

ID.	ELEMENTO A PROBAR	PRECONDICIÓN	DATOS DE ENTRADA	RESULTADO ESPERADO	RESULTADO OBTENIDO	DOCUMENTOS DE RESPALDO	ESTADO DEL CASO DE PRUEBA	FECHA DE APROBACIÓN
1	Métricas por Responsable	Disponibilidad de la aplicación y de la pantalla mencionada.	Pasos a realizar usuario 1: 1.-Autenticarse en el sistema el usuario 1: ebalseca. 2.-Seleccionar del Menú Principal la opción "Reportes". 3.-Seleccionar del menú de Reportes la opción "Métricas por Responsable" 4.-Seleccionar del filtro Proyecto: PRY_Proceso de atención de incidencias 5.-Seleccionar del filtro Responsable: Isabel Vasco Aguas. 6.-Presionar el botón Exportar	Se generó el reporte en excel con las métricas existentes del proyecto PRY_Proceso de atención de incidencias del responsable Isabel Vasco Aguas.	Exitoso	LHS_DOC-RDP_20140912_EB	Aprobado	12/09/2014
2			Pasos a realizar usuario 2: 1.-Autenticarse en el sistema el usuario 2: esalas. 2.-Seleccionar del Menú Principal la opción "Reportes". 3.-Seleccionar del menú de Reportes la opción "Métricas por Responsable" 4.-Seleccionar del filtro Proyecto: PRY_Proceso de atención de incidencias 5.-Seleccionar del filtro Responsable: Isabel Vasco Aguas. 6.-Presionar el botón Exportar	Se generó el reporte en excel con las métricas existentes del proyecto PRY_Proceso de atención de incidencias del responsable Isabel Vasco Aguas.	Exitoso	LHS_DOC-RDP_20140912_ES		
3			Pasos a realizar usuario 3: 1.-Autenticarse en el sistema el usuario 3: gcaiza. 2.-Seleccionar del Menú Principal la opción "Reportes". 3.-Seleccionar del menú de Reportes la opción "Métricas por Responsable" 4.-Seleccionar del filtro Proyecto: PRY_Proceso de atención de incidencias 5.-Seleccionar del filtro Responsable: Isabel Vasco Aguas. 6.-Presionar el botón Exportar	Se generó el reporte en excel con las métricas existentes del proyecto PRY_Proceso de atención de incidencias del responsable Isabel Vasco Aguas.	Exitoso	LHS_DOC-RDP_20140912_GC		
4			Pasos a realizar usuario 4: 1.-Autenticarse en el sistema el usuario 4: pperez. 2.-Seleccionar del Menú Principal la opción "Reportes". 3.-Seleccionar del menú de Reportes la opción "Métricas por Responsable" 4.-Seleccionar del filtro Proyecto: PRY_Proceso de atención de incidencias 5.-Seleccionar del filtro Responsable: Isabel Vasco Aguas. 6.-Presionar el botón Exportar	Se generó el reporte en excel con las métricas existentes del proyecto PRY_Proceso de atención de incidencias del responsable Isabel Vasco Aguas.	Exitoso	LHS_DOC-RDP_20140912_CP		

Figura 3.59 Resultado del Caso de Prueba de Estrés 02 (PLANTILLA 16-CPR02).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.



## DATOS DEL CASO DE PRUEBA

Nombre del Proyecto	LogiHistoricos		
Id. Requerimiento	RN3		
Id. Caso de prueba	CPR03		
Tipo de prueba	Estrés		
Módulo a probar	Registro de Medidas		
Autor del caso de prueba	Analista programador/ Tester		
Nombre del tester	Gabriela Caiza		
Objetivo	Probar la solidez de la aplicación en los momentos de carga.		
Descripción	Validar el rendimiento del sistema LogiHistoricos al generar varios reportes sobre las métricas estándar a la vez.		
Fecha de creación	12/09/2014	Fecha de ejecución:	12/09/2014

## EJECUCIÓN Y RESULTADOS DEL CASO DE PRUEBA:

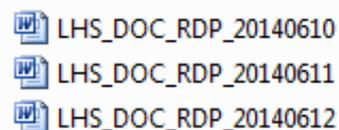
ID.	ELEMENTO A PROBAR	PRECONDICIÓN	DATOS DE ENTRADA	RESULTADO ESPERADO	RESULTADO OBTENIDO	DOCUMENTOS DE RESPALDO	ESTADO DEL CASO DE PRUEBA	FECHA DE APROBACIÓN
1	Métricas Estándar	Disponibilidad de la aplicación y de la pantalla mencionada.	Pasos a realizar usuario 1: 1.-Autenticarse en el sistema usuario 1: ebalseca. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas Estándar" 3.- Selección del filtro Metodología:Proyectos Parametrización 4.-Seleccionar del filtro Plataforma: .Net 5.-Seleccionar del filtro Criterio Tamaño: PCO 6.-Presionar el botón Exportar	Se genera el reporte en Excel con las métricas existentes del proyecto para la metodología Proyectos Parametrización con plataforma .Net y criterio de tamaño PCO.	Exitoso	LHS_DOC-RDP_20140912_EB	Aprobado	12/09/2014
2			Pasos a realizar usuario 2: 1.-Autenticarse en el sistema usuario 2: esalas. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas Estándar" 3.- Selección del filtro Metodología:Proyectos Parametrización 4.-Seleccionar del filtro Plataforma: .Net 5.-Seleccionar del filtro Criterio Tamaño: PCO 6.-Presionar el botón Exportar	Se genera el reporte en Excel con las métricas existentes del proyecto para la metodología Proyectos Parametrización con plataforma .Net y criterio de tamaño PCO.	Exitoso	LHS_DOC-RDP_20140912_ES		
3			Pasos a realizar usuario 3: 1.-Autenticarse en el sistema usuario 3: gcaiza. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas Estándar" 3.- Selección del filtro Metodología:Proyectos Parametrización 4.-Seleccionar del filtro Plataforma: .Net 5.-Seleccionar del filtro Criterio Tamaño: PCO 6.-Presionar el botón Exportar	Se genera el reporte en Excel con las métricas existentes del proyecto para la metodología Proyectos Parametrización con plataforma .Net y criterio de tamaño PCO.	Exitoso	LHS_DOC-RDP_20140912_GC		
4			Pasos a realizar usuario 4: 1.-Autenticarse en el sistema usuario 4: pperez. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas Estándar" 3.- Selección del filtro Metodología:Proyectos Parametrización 4.-Seleccionar del filtro Plataforma: .Net 5.-Seleccionar del filtro Criterio Tamaño: PCO 6.-Presionar el botón Exportar	Se genera el reporte en Excel con las métricas existentes del proyecto para la metodología Proyectos Parametrización con plataforma .Net y criterio de tamaño PCO.	Exitoso	LHS_DOC-RDP_20140912_CP		

Figura 3.60 Resultado del Caso de Prueba de Estrés 03 (PLANTILLA 16-CPR03).

Elaborado por: Balseca Marcela, Caiza Gabriela

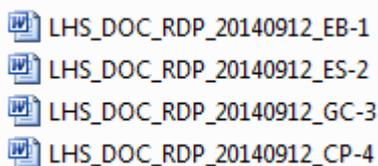
Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.

**23.** Los tester ejecutan los casos de pruebas CP01, CP02, CP03 y obtienen los reportes de pruebas de funcionalidad generados que se observan en el ANEXO B en la carpeta Reporte de pruebas. Los reportes de pruebas generados son los siguientes:



**Figura 3.61** Nombres de los reportes de pruebas de funcionalidad generados

**24.** Los tester ejecutan los casos de pruebas CPR01, CPR2, CPR3 y obtienen los reportes de pruebas de estrés generados que se observan en el ANEXO B en la carpeta Reporte de pruebas de estrés. Los reportes de pruebas generados son los siguientes:



**Figura 3.62** Nombres de los reportes de pruebas de estrés generados

## FASE 6: Pruebas de aceptación

25. Los tester se encargan de actualizar los resultados de los Casos de prueba de aceptación: mostrados en las Figuras 3.63, 3.64 y 3.65.

EJECUCIÓN Y RESULTADOS DEL CASO DE PRUEBA:							
ID.	ELEMENTO A PROBAR	PRECONDICIÓN	DATOS DE ENTRADA	RESULTADO ESPERADO	RESULTADO OBTENIDO	ESTADO DEL CASO DE PRUEBA	FECHA DE APROBACIÓN
1	Métricas por proyecto	Disponibilidad de la aplicación y de la pantalla mencionada.	Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Proyecto" 4.-Seleccionar del filtro un proyecto determinado.	Se visualice en la pantalla las métricas existentes del proyecto seleccionada y se pueda visualizar el valores de las métricas y los indicadores.	Pendiente	Aprobado	12/06/2014
2			Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Proyecto" 4.-Seleccionar del filtro un proyecto determinado 5.-Presionar el botón Exportar	Se genera el reporte en Excel con las métricas existentes del proyecto seleccionado y se pueda visualizar el valor de las métricas, los indicadores y las observaciones.	Fallido		
3			Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Proyecto" 4.-Seleccionar del filtro un proyecto determinado 5.-Presionar el botón Salir	Se cierre la pantalla de Métricas por Proyecto.	Exitoso		

Figura 3.63 Resultado del Caso de Prueba de Aceptación 01 (PLANTILLA 16-CPA01).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.



DATOS DEL CASO DE PRUEBA			
Nombre del Proyecto	LogiHistoricos		
Id. Requerimiento	RN2		
Id. Caso de prueba	CPA02		
Tipo de prueba	Aceptación		
Módulo a probar	Registro de Medidas		
Autor del caso de prueba	Analista programador/ Tester		
Nombre del tester	Marcela Balseca		
Objetivo	Probar la funcionalidad de la pantalla de métricas por proyecto.		
Descripción	Validar la funcionalidad de la pantalla métricas por proyecto.		
Fecha de creación	06/06/2014	Fecha de ejecución:	12/06/2014
EJECUCIÓN Y RESULTADOS DEL CASO DE PRUEBA:			

ID.	ELEMENTO A PROBAR	PRECONDICIÓN	DATOS DE ENTRADA	RESULTADO ESPERADO	RESULTADO OBTENIDO	ESTADO DEL CASO DE PRUEBA	FECHA DE APROBACIÓN
1	Métricas por proyecto	Disponibilidad de la aplicación y de la pantalla mencionada.	Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Proyecto" 4.-Seleccionar del filtro un proyecto determinado.	Se visualice en la pantalla las métricas existentes del proyecto seleccionada y se pueda visualizar el valores de las métricas y los indicadores.	Exitoso	Aprobado	12/06/2014
2			Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Proyecto" 4.-Seleccionar del filtro un proyecto determinado 5.-Presionar el botón Exportar	Se genera el reporte en Excel con las métricas existentes del proyecto seleccionado y se pueda visualizar el valor de las métricas, los indicadores y las observaciones.	Pendiente  Exitoso		
3			Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas por Proyecto" 4.-Seleccionar del filtro un proyecto determinado 5.-Presionar el botón Salir	Se cierre la pantalla de Métricas por Proyecto.	Exitoso		

Figura 3.64 Resultado del Caso de Prueba de Aceptación 02 (PLANTILLA 16-CPA02).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.



DATOS DEL CASO DE PRUEBA			
Nombre del Proyecto	LogiHistoricos		
Id. Requerimiento	RN3		
Id. Caso de prueba	CPA03		
Tipo de prueba	Aceptación		
Módulo a probar	Registro de Medidas		
Autor del caso de prueba	Analista programador/ Tester		
Nombre del tester	Marcela Balseca		
Objetivo	Probar la funcionalidad de la pantalla de métricas estándar.		
Descripción	Validar la funcionalidad de la pantalla métricas estándar.		
Fecha de creación	06/06/2014	Fecha de ejecución:	12/06/2024
Ejecución y Resultados del Caso de Prueba:			

ID.	ELEMENTO A PROBAR	PRECONDICIÓN	DATOS DE ENTRADA	RESULTADO ESPERADO	RESULTADO OBTENIDO	ESTADO DEL CASO DE PRUEBA	FECHA DE APROBACIÓN
1	Métricas Estándar	Disponibilidad de la aplicación y de la pantalla mencionada.	Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas Estándar" 4.-Seleccionar del panel de filtros: una Metodología, una Plataforma y un Criterio Tamaño.	Se visualice en pantalla las métricas estándar existentes de acuerdo a los filtros seleccionados y se pueda visualizar el valor de las métrica y los indicadores.	Exitoso	Aprobado	13/06/2014
2			Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas Estándar" 4.-Seleccionar del panel de filtros: una Metodología, una Plataforma y un Criterio Tamaño. 5.-Presionar el botón Exportar	Se genera el reporte en Excel de acuerdo a los filtros seleccionados y se pueda visualizar el valor de las métricas y los indicadores.	Fallido  Exitoso		
3			Pasos a realizar: 1.-Autenticarse en el sistema. 2.-Seleccionar del Menú Principal la opción Reportes. 3.-Seleccionar del menú de Reportes la opción "Métricas Estándar" 4.-Seleccionar del panel de filtros: una Metodología, una Plataforma y un Criterio Tamaño. 5.-Presionar el botón Salir	Se cierra la pantalla de Métricas Estándar.	Exitoso		

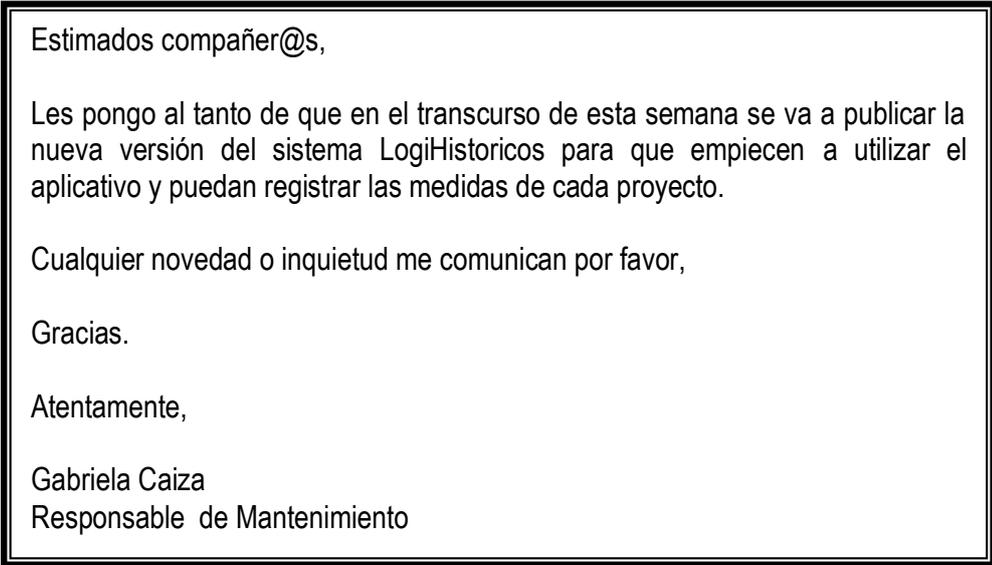
Figura 3.65 Resultado del Caso de Prueba de Aceptación 03 (PLANTILLA 16-CPA03).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.

## FASE 7: Puesta en producción

26. El Responsable de mantenimiento se encarga de realizar la Notificación de puesta en producción del sistema LogiHistoricos mostrada en la Figura 3.66.



Estimados compañer@s,

Les pongo al tanto de que en el transcurso de esta semana se va a publicar la nueva versión del sistema LogiHistoricos para que empiecen a utilizar el aplicativo y puedan registrar las medidas de cada proyecto.

Cualquier novedad o inquietud me comunican por favor,

Gracias.

Atentamente,

Gabriela Caiza  
Responsable de Mantenimiento

**Figura 3.66 Notificación de puesta en producción del sistema.**

**Elaborado por:** Balseca Marcela, Caiza Gabriela

**Fuente:** Información obtenida de la Empresa Logiciel Cía. Ltda.

27. Copia del software: El producto software se empaquetó en un archivo comprimido llamado WebLogiHistoricosKendo\_Backup\_2014.06.13, que contiene los archivos del sistema y que reposa en las instalaciones de la empresa.
28. Software en producción: El sistema está listo para que pueda ser utilizado por los usuarios finales de la misma empresa.

29. El Responsable de mantenimiento se encargará de actualizar el Registro de la solicitud de modificación mostrado en la Figura 3.67.

LogiHistoricos											
DATOS GENERALES DEL PROYECTO O SISTEMA											
Nombre del Proyecto	LogiHistoricos										
Líder del Proyecto	Isabel Vasco										
Gestor de Peticiones	Marcela Balseca										
REGISTRO DEL ESTUDIO DE LA SOLICITUD DE MODIFICACIÓN											
ID. SOLICITUD	DESCRIPCIÓN O DETALLE	RESPONSABLE DEL MANTENIMIENTO	FECHA RECEPCIÓN	FECHA APROBACIÓN	FECHA CIERRE	TIPO DE MANTENIMIENTO	MOTIVO DE SELECCIÓN	PRIORIDAD	ESTADO DEL CAMBIO	MOTIVO DEL RECHAZO	ESTADO DE IMPLEMENTACIÓN
SM01	Generar un reporte de las métricas por proyecto y exportarlo a Excel.	Gabriela Caiza	02/06/2014	03/06/2014	13/06/2014	Perfectivo	Seleccionamos este tipo de mantenimiento porque se va añadir nuevos requerimientos al sistema existente.	Medio	Aprobado		Terminado
	Calcular y generar un reporte de las métricas por responsable y exportarlo a Excel.										
	Generar un reporte de las métricas estándar y exportarlo a Excel.										

Figura 3.67 Registro del estudio de la solicitud de modificación actualizado (PLANTILLA 2).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.

30. Archivo de la documentación del sistema: la documentación del sistema actualizada y generada durante el proceso de mantenimiento se la organizó en carpetas y se procedió a guardarla en un repositorio de la empresa para su control: mostrado en la Figura 3.68.



Figura 3.68 Documentación del sistema LogiHistoricos  
Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.

## Aseguramiento de Calidad

31. El Analista SQA se encarga de llenar y actualizar el Registro de las revisiones de aseguramiento de calidad mostrado en la Figura 3.69.

DATOS GENERALES DEL PROYECTO O SISTEMA								
Nombre del Proyecto	LogiHistoricos							
Líder del Proyecto	Isabel Vasco							
Responsable de SQA	Gabriela Caiza							
REGISTRO DE LAS REVISIONES DE ASEGURAMIENTO DE CALIDAD								
ID	ENTREGABLES DEL PROCESO DE MANTENIMIENTO	AUDITADO	AUDITOR	FECHA DE REVISIÓN	CALIFICACIÓN	MOTIVO RECHAZO	ACCIÓN RECOMENDADA	OBSERVACIONES
1	Solicitud de modificación	Gestor de peticiones	Analista - SQA	03/06/2014	Aprobado			
2	Registro del estudio de la solicitud de modificación	Responsable de mantenimiento	Analista - SQA	03/06/2014	Pendiente			Este documento se lo va a ir completando durante todo el proceso de mantenimiento
				09/06/2014	Pendiente			Actualizar el resultado del estado de la implementación
				13/06/2014	Aprobado			
3	Inventario de la documentación del sistema	Responsable de mantenimiento	Analista - SQA	04/06/2014	Aprobado			
4	Registro de asignación de recursos	Responsable de mantenimiento	Analista - SQA	04/06/2014	Aprobado			
5	Definición del impacto del cambio	Equipo de mantenimiento	Analista - SQA	04/06/2014	Aprobado			
6	Matriz de rastreabilidad	Responsable de mantenimiento	Analista - SQA	04/06/2014	Aprobado			
7	Estimación de costo y esfuerzo	Equipo de mantenimiento	Analista - SQA	04/06/2014	Aprobado			
8	Lista de elementos a modificar	Equipo de mantenimiento	Analista - SQA	05/06/2014	Pendiente		Se recomienda llenar todos los campos de la plantilla	Este documento se lo va a ir completando durante todo el proceso de mantenimiento
				11/06/2014	Pendiente			Actualizar el resultado del estado de la implementación
				12/06/2014	Aprobado			
9	Lista de documentos a modificar	Responsable de mantenimiento	Analista - SQA	05/06/2014	Pendiente			Este documento se lo va a ir completando durante todo el proceso de mantenimiento
				11/06/2014	Pendiente			Actualizar el resultado del estado de la actualización
				12/06/2014	Aprobado			
10	Plan de pruebas	Equipo de mantenimiento	Analista - SQA	05/06/2014	Pendiente		Actualizar el plan	Falta actualizar con los últimos cambios
				06/06/2014	Aprobado			
11	Plan de implementación	Responsable de mantenimiento	Analista - SQA	05/06/2014	Pendiente		Actualizar fechas del plan	
				06/06/2014	Aprobado			
12	Casos de prueba	Analista-programador Tester	Analista - SQA	05/06/2014	Pendiente			Falta registrar los resultados de los casos de prueba
				12/06/2014	Pendiente			Falta registrar los resultados del caso de prueba de aceptación
				13/06/2014	Aprobado			
13	Reporte de pruebas	Tester	Analista - SQA	11/06/2014	Rechazado	Falta plantilla	Se recomienda utilizar la plantilla establecida para los	
				12/06/2014	Aprobado			

Figura 3.69 Registro de las revisiones de aseguramiento de calidad (PLANTILLA 18).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.

## Gestión de la configuración

32. El Analista SQA se encarga de llenar y actualizar el Registro del control de versión del sistema: mostrado en la Figura 3.70.



**DATOS GENERALES DEL PROYECTO O SISTEMA**

<b>Nombre del Proyecto</b>	LogiHistoricos
<b>Líder del Proyecto</b>	Isabel Vasco
<b>Responsable SQA</b>	Gabriela Caiza

**REGISTRO DEL CONTROL DE VERSIONES DEL SISTEMA**

ID. SOLICITUD	SISTEMA	VERSIÓN ACTUAL	NUEVA VERSIÓN	OBSERVACIONES
SM01	Generar 3 reportes de métricas: por proyecto, por responsable y estándar.	V1.0	V1.1	

Figura 3.70 Registro del control de versiones del sistema (PLANTILLA 19).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.

33. El Analista SQA se encarga de llenar y actualizar el Registro de la auditoría de gestión de la configuración: mostrado en la Figura 3.71.

													
DATOS GENERALES DEL PROYECTO O SISTEMA													
Nombre del Proyecto	LogiHistoricos												
Líder del Proyecto	Isabel Vasco												
Responsable de SQA	Gabriela Caiza												
REGISTRO DE LA AUDITORÍA DE GESTIÓN DE LA CONFIGURACIÓN													
FASE	ELEMENTOS DE CONFIGURACIÓN	DESCRIPCIÓN	VERSIÓN ACTUAL	NUEVA VERSIÓN	FECHA REVISIÓN	RESPONSABLE	PRODUCTO ACABADO	APROBADO FORMALMENTE	CUMPLE ESTÁNDARES	RESULTADO REVISIÓN	RECOMENDACIONES	OBSERVACIONES	
Gestión de Requerimientos	LHS_ESP_FUN	Especificación de requerimientos	2.0	3.0	12/06/2014	Analista - SQA	SI	SI	SI	Aprobado			
	LHS_FOR_GRQ	Gestión de requerimientos	2.0	3.0	12/06/2014	Analista - SQA	SI	SI	SI	Aprobado			
Planeación del Proyecto	LHS_FOR_ESC	Listado de elementos de la configuración	2.0	N/A									
	LHS_PLN_PRY	Plan de gestión de la configuración	2.0	N/A									
	LHS_DOC_RCO	Riesgos y contingencias del proyecto	2.0	N/A									
	LHS_PLN_ADM	Plan de administración del proyecto	3.0	N/A									
Análisis y Diseño	LHS_DGA_ERC	Modelo conceptual	12.0	N/A									
	LHS_DGA_ERF	Modelo físico	12.0	N/A									
	LHS_SCT_BDD	Script de generación de la base de datos	6.0	N/A									
	LHS_DGA_PRY	Diagramas de casos de uso		1.0	2.0	12/06/2014	Analista - SQA	No	SI	SI	Pendiente	Falta actualizar el documento	
						13/06/2014	Analista - SQA	SI	SI	SI	Aprobado		
LHS_ESP_CUS	Especificación de casos de uso		1.0	2.0	12/06/2014	Analista - SQA	SI	No	SI	Pendiente	Ordenar el documento correctamente		
					13/06/2014	Analista - SQA	SI	SI	SI	Aprobado			
Implementación		Medidas-Métricas	1.0	N/A									
		Diagrama de n-capas lógicas	1.0	N/A									
	LHS_MAN_USR	Manual de usuario		1.0	2.0	12/06/2014	Analista - SQA	SI	No	SI	Pendiente	Incluir las imágenes de todas las pantallas	
						13/06/2014	Analista - SQA	SI	SI	SI	Aprobado		
	LHS_ESP_TEC	Documento de especificaciones técnicas	2.0	N/A									
		LogiHistoricosRecursos	1.0	N/A									
		Plantilla estimación tamaño	1.0	N/A									
LHS_FOR_AUD	Formulario de tablas a auditar	2.0	N/A										
Administración del Proyecto	LHS_ACT_SEI	Acta de seguimiento interno	13.0	N/A									

Figura 3.71 Registro de la auditoría de gestión de la configuración (PLANTILLA 20).

Elaborado por: Balseca Marcela, Caiza Gabriela

Fuente: Información obtenida de la Empresa Logiciel Cía. Ltda.

### 3.3 ANÁLISIS DE RESULTADOS

Al aplicar las actividades y tareas de la fase 1 definidas en la guía, no se presentaron inconvenientes. Sin embargo al ir realizando las actividades y tareas de las siguientes fases de la guía se presentaron varias dificultades, entre las cuales se pueden mencionar las siguientes:

- Existían actividades y tareas que no se aplicaban en algunos casos.
- El nombre de las plantillas definidas en la guía no concordaban con su contenido.
- Existían plantillas que no contenían los campos necesarios para ingresar la información recopilada.
- Existían tareas que no estaban claramente definidas para poder ejecutarlas correctamente.

Debido a estas dificultades, se realizaron varios cambios en el contenido de la guía, como cambios en las plantillas propuestas y modificaciones en algunas actividades y tareas del proceso.

El proceso de mantenimiento aplicado permitió definir qué se debe realizar durante todo la implementación del cambio, lo cual se refleja en los cambios que se obtuvieron al final del proceso de mantenimiento del sistema LogiHistoricos.

La aplicación de esta guía ha sido de mucha utilidad en la empresa, ya que está ayudando a tener un control de todo el proceso de mantenimiento, así como está permitiendo tener un registro de los cambios realizados, lo que ayuda a que el sistema y su documentación se mantengan actualizados en todo momento bajo una misma versión.

## **CAPÍTULO 4**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **4.1 CONCLUSIONES**

- El mantenimiento de software tiene que ver con la necesidad permanente de cambio al producto elaborado, donde los cambios se dan para corregir errores, mejorar sus características, prevenir futuras fallas o adaptar el producto a un nuevo entorno, una vez que el sistema software ha sido puesto en producción.
- En este proyecto se ha podido concluir de manera evidente, que el proceso de mantenimiento de software no es una tarea fácil, lo que sugiere la urgente necesidad de seguir procedimientos claramente definidos para mantener productos software.
- En la actualidad existen algunas metodologías de mantenimiento de software que deberían ser utilizadas por aquellos profesionales que deseen realizar cambios a los productos software.
- Se ha visto que las metodologías más conocidas para realizar el mantenimiento de software son Mantema y Métrica V3 ya que cubren todos los aspectos del mantenimiento en cuanto a procesos, tareas, actividades y documentación.
- Para realizar el mantenimiento, Mantema demanda mucho tiempo y esfuerzo para ser implementada en una pequeña empresa, en cambio Métrica V3 tiene un flujo corto y fue ideada pensando en ayudar a las pequeñas organizaciones.

## 4.2 RECOMENDACIONES

- Se recomienda establecer un procedimiento para analizar los fallos obtenidos y manejar una base de conocimientos sobre los problemas reportados al dar mantenimiento a un sistema software.
- Se recomienda dar a conocer y concienciar a todo el personal de las empresas que desarrollan software, de los beneficios que implica tener una guía de procedimientos bien definida para realizar el mantenimiento de software.
- Se recomienda llevar un control estricto de cada uno de los cambios realizados sobre un sistema software para poder controlar sus versiones y facilitar nuevos cambios.
- Se recomienda almacenar toda la información recopilada a lo largo de los procesos de mantenimiento realizados sobre un producto software, a fin de resguardar toda aquella información útil e indispensable para procesos similares a futuro.

## GLOSARIO

**Actividad:** Grupo de operaciones afines ejecutadas por una misma persona o una misma unidad administrativa, como parte de una función determinada.

**Análisis:** Consiste en identificar los componentes de un todo, separarlos y examinarlos para lograr acceder a sus principios más elementales.

**Analogía:** Es una comparación entre objetos, conceptos o experiencias. Al establecer una analogía, se indican características particulares y generales y se establecen las semejanzas y diferencias entre los elementos contrastados.

**Auditoría:** Es un proceso sistemático para obtener y evaluar de manera objetiva las evidencias relacionadas con informes sobre actividades económicas y otros acontecimientos relacionados.

**Cambio:** Acción y efecto de cambiar.

**Código:** Es un texto desarrollado en un lenguaje de programación y que debe ser compilado o interpretado para poder ejecutarse en una computadora.

**Coherencia:** Es la existencia de relación o lógica entre las diferentes partes de una afirmación o entre las diferentes afirmaciones o posturas de un discurso.

**Complejidad:** Tiene que ver con la diversidad de elementos que componen una situación; un todo que se compone de partes que interactúan y que estas a su vez se encuentran en contacto con su medio ambiente.

**Control:** Es un mecanismo preventivo y correctivo adoptado por la entidad que permite la oportuna detección y corrección de ineficiencias o incongruencias en el curso de la ejecución y evaluación de las acciones, con el propósito de procurar el cumplimiento de la normatividad que las rige, y las estrategias, políticas, objetivos, metas y asignación de recursos.

**Defecto:** Resultado de un fallo o deficiencia durante el proceso de creación de programas de un computador.

**Desarrollador:** Es un programador que se dedica a uno o más aspectos del proceso de desarrollo de software. El desarrollador puede contribuir a la visión general del proyecto más a nivel de aplicación que a nivel de componentes o en las tareas de programación individuales.

**Equipo:** Es un grupo de seres humanos que se reúnen y trabajan en conjunto para alcanzar una meta en común. Para esto, el equipo mantiene una cierta organización que le permita conseguir sus objetivos.

**Estándar:** Sirve como tipo, norma, modelo, patrón o referencia por ser corriente, de serie.

**Falla:** Es la discrepancia visible que se produce al ejecutar un programa con un defecto, el cual es incapaz de funcionar correctamente (no sigue su curso normal).

**Función:** Conjunto de actividades afines y coordinadas necesarias para alcanzar los objetivos de la Empresa, se definen a partir de las disposiciones jurídicas administrativas.

**Guía Técnica:** Documento que contiene la metodología, las instrucciones y la información esencial para dirigir u orientar la elaboración de diversos tipos de documentos normativos.

**Hardware:** Equipo utilizado para el funcionamiento de una computadora. El hardware se refiere a los componentes materiales de un sistema informático.

**Implementación:** Es la etapa donde efectivamente se programa el sistema.

**Ingeniería Inversa:** Reconstruir el proceso de ingeniería de un producto a partir de ciertos artefactos de dicho producto.

**Interfaz:** Hace referencia al conjunto de métodos para lograr interactividad entre un usuario y el computador. / Conexión que se da de manera física y a nivel de utilidad entre dispositivos o sistemas.

**Mantenimiento:** Capacidad de un producto software a ser modificado. Estas modificaciones incluyen correcciones, mejoras, o adaptaciones a cambios en el entorno, los requisitos o las especificaciones funcionales.

**Mantenimiento de Software:** Es la modificación de un sistema después de su liberación, para: corregir errores, mejorar el desempeño y otras propiedades, prevenir futuras fallas o adaptar el producto a un nuevo ambiente.

**Método:** Un método es una serie de pasos sucesivos que conducen a una meta. El objetivo es llegar a tomar decisiones y una teoría que permita generalizar y resolver de la misma forma problemas semejantes en el futuro.

**Metodología:** El conjunto de procedimientos que determinan una investigación de tipo científico o marcan el rumbo de una exposición doctrinal.

**Módulos:** Forman parte de un sistema y suele estar conectado de alguna manera con el resto de los componentes.

**Operabilidad / Operatividad:** Capacidad para realizar una función.

**Proceso:** Se denomina proceso a la consecución de determinados actos, acciones, sucesos o hechos que deben necesariamente sucederse para completar un fin específico. Todos estos pasos o instancias que componen un proceso deben ser organizados, coordinados y realizados de manera sistemática, de a uno por vez (secuencia alternativa) o pueden incluso superponerse las instancias (secuencia simultánea).

**Producto:** Es el punto central de la oferta que realiza toda empresa u organización (ya sea lucrativa o no) a su mercado meta para satisfacer sus necesidades y deseos, con la finalidad de lograr los objetivos que persigue.

**Recursos:** Conjunto de elementos interdependientes e interactuantes, cuyas relaciones entre sí, o entre sus atributos, determinan un todo unitario que realiza un efecto, una función o un objetivo.

**Reestructuración:** Modificación del software para hacerlo más fácil de entender y cambiar o menos susceptible de incluir errores en cambios posteriores.

**Refactorización:** Proceso de cambio de un sistema software de forma que su comportamiento externo se vea afectado pero que mejora su estructura interna.

**Registro:** Documento que presenta resultados obtenidos o proporciona evidencias de actividades desempeñadas.

**Reingeniería:** Es el rediseño de un proceso en un negocio o un cambio drástico de un proceso. Es comenzar de cero, es un cambio de todo o nada, además ordena la empresa alrededor de los procesos.

**Rendimiento:** Es la proporción que surge entre los medios empleados para obtener algo y el resultado que se consigue. El beneficio o el provecho que brinda algo o alguien también se conocen como rendimiento.

**Requerimientos:** Es una exigencia que tiene un software para que pueda funcionar de manera correcta.

**Riesgo:** Se define como la combinación de la probabilidad de que se produzca un evento y sus consecuencias negativas. Los factores que lo componen son la amenaza y la vulnerabilidad.

**Rol:** Papel que desempeña una persona o un grupo en cualquier actividad.

**Sistema:** Es un conjunto de partes o elementos organizados y relacionados que interactúan entre sí para lograr un objetivo. Los sistemas reciben (entrada) datos, y proveen (salida) información.

**Sistema heredado:** Es un método, tecnología, computador o aplicación antiguo que continúa en uso porque aún satisface las necesidades de los usuarios, aun existiendo nuevas tecnologías o métodos más eficientes disponibles.

**Software:** Comúnmente se utiliza este término para referirse de una forma muy genérica a los programas de un dispositivo informático, es decir, son las instrucciones que el computador necesita para funcionar, no existen físicamente, o lo que es igual, no se pueden ver ni tocar.

**Subcontratación:** Es la contratación que una empresa hace de otra empresa, para que ésta última realice parte de los servicios por los que la primera ha sido contratada directamente.

**Transformaciones:** Manipulación, por medio de transformaciones automáticas ejecutadas por un computador, del código o modelo de un sistema para añadirle / modificarle / eliminarle elementos.

**Usuario:** Es la persona que utiliza algún tipo de objeto o que es destinataria de un servicio, ya sea privado o público.

**Versión:** Es un número o nombre que se asigna a un programa informático para mencionar su nivel de desarrollo y su actualización.

## BIBLIOGRAFÍA

- [1] R. S. Pressman, *Ingeniería de Software*, Séptima ed., New York: Mac Graw Hill, 2010.
- [2] J. A. Martínez Párraga, «Planificación y Gestión de Sistemas de Información,» de *Estándar IEEE 1219 de Mantenimiento de software*, 1999.
- [3] «Tecnología de la información. Proceso del ciclo de vida del software,» de *Norma Técnica Peruana. NTP-ISO/IEC 12207*, AENOR, 2006.
- [4] IEEE Computer Society, «Software Maintenance,» [En línea]. Available: <http://www.computer.org/portal/web/swebok/html/ch6#Ref1>. [Último acceso: 19 Agosto 2013].
- [5] I. Sommerville, *Ingeniería del Software*, Novena ed., Addison-Wesley, 2011.
- [6] M. G. Piattini Velthuis, *Mantenimiento del Software, modelos, técnicas y métodos para la gestión del cambio*, México: Alfa-Omega, 2001.
- [7] S. Lamayzi, «Planificación y Gestión de Sistemas de Información,» de *La norma ISO 14764*, 1999.
- [8] M. Polo Usaola, *MANTEMA: Una metodología para mantenimiento de software*, España, 2000.
- [9] Ministerio de Administración Pública de España, «Metodología de Planificación, Desarrollo y Mantenimiento de Sistemas de Información,» de *Metodología Métrica Versión 3*, 2000.
- [10] P. F. Ruiz Macario, «Mantenimiento del Software S1,» 2001. [En línea]. Available: <http://alarcos.esi.uclm.es/per/fruiz/curs/mso/trans/s1.pdf>. [Último acceso: 3 Enero 2014].
- [11] C. Contreras, «Guía técnica para elaboración de manuales de procedimientos,» [En línea]. Available: [http://www.up.ac.pa/ftp/2010/d\\_planificacion/documentos/guia\\_tecnica\\_para\\_al\\_elaboracion%20de%20manuales%20de%20procedimientos.pdf](http://www.up.ac.pa/ftp/2010/d_planificacion/documentos/guia_tecnica_para_al_elaboracion%20de%20manuales%20de%20procedimientos.pdf). [Último acceso: 25 Mayo 2014].

## **ANEXOS**

Los anexos que se enumeran a continuación se encuentran en el disco adjunto.

**ANEXO A** - PLANTILLAS DE LA GUÍA

**ANEXO B** - PLANTILLAS DEL CASO DE ESTUDIO