



A.F. 133747



## **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

### **MAESTRÍA EN SISTEMAS DE INFORMACIÓN GERENCIAL FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN**

"Desarrollo e implementación de un Sistema de Información para la Gestión de Defectos en el diseño y modelamiento conceptual de proyectos informáticos que utilizan herramientas de ingeniería de software asistida por computadora."

#### **TESIS DE POSTGRADO**

Previa a la obtención del Título de:

#### **MAGISTER EN SISTEMAS DE INFORMACIÓN GERENCIAL**

Presentado por:

Ing. Daniel Andrés Silva Palacios

**Guayaquil - Ecuador**

**2014**

## **AGRADECIMIENTOS**

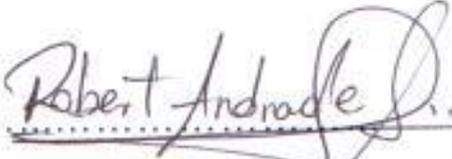
Mis más sinceros agradecimientos a María Fernanda Granda Juca por su apoyo y guía en la realización de esta tesis. A mi director de tesis y a los profesores de la maestría por haberme instruido en estos años de estudio.

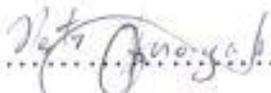
## DEDICATORIA

A mi esposa Delia Gabriela Granda Juca por todo su apoyo incondicional en la realización de esta tesis. A mi familia por darme siempre todo su apoyo. Gracias a Dios que me da fortaleza, sabiduría y perseverancia para poder lograr mis objetivos.

## TRIBUNAL GRADUACIÓN

.....  
MSIG. LENIN FREIRE  
DIRECTOR DE LA MSIG

  
.....  
MGS. ROBERT ANDRADE  
DIRECTOR DE TESIS

  
.....  
MGS. NÉSTOR ARREAGA  
MIEMBRO DEL TRIBUNAL

.....  
MSc. GUIDO CAICEDO R.  
MIEMBRO DEL TRIBUNAL

## DECLARACIÓN EXPRESA

"La responsabilidad por los hechos, ideas y doctrinas expuesto en este proyecto, nos corresponden exclusivamente; y el Patrimonio intelectual de la misma, a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL".

(Reglamento de Exámenes y Títulos Profesionales de la ESPOL)



.....  
Ing. Daniel Andrés Silva Palacios

## RESUMEN

La evolución constante de las tecnologías de la información y comunicación, el aumento de empresas dedicadas a producir software en Ecuador y el apoyo gubernamental a productos de software nacionales hacen el marco ideal para la creación de emprendimientos y empresas que basen su modelo de negocio en desarrollar productos informáticos. Estos negocios son vistos como una oportunidad de trabajo para los futuros técnicos e ingenieros de sistemas. Estas empresas buscan implementar estándares de calidad, mecanismos de control en sus procesos de desarrollo y ofrecer productos de alta calidad que satisfaga las necesidades de sus clientes.

La presente tesis plantea el diseño y desarrollo de una herramienta para la administración de defectos de software de forma que ayude a gestionar la información sobre los defectos encontrados al realizar proyectos informáticos. Especialmente esta herramienta da soporte para el desarrollo a través de la Ingeniería Basada en Modelos que es una técnica que permite generar automáticamente un sistema informático partiendo de modelos conceptuales.

Esta herramienta permite realizar un análisis de los defectos a través de reportes y gráficos estadísticos que presentan la información de forma que pueda ser fácilmente analizada, además incluye una serie de herramientas para la gestión de defectos por parte de los jefes y gerentes de proyectos. Se puede mantener un registro histórico ordenado de la información de encontrada, realizar análisis de causas efectos, es posible utilizar la información histórica como fuente de

información para solucionar defectos y crear medidas de prevención. En consecuencia este sistema permite aprender del pasado y así evitar errores en futuros proyectos informáticos.

## CONTENIDO

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1</b> .....	<b>4</b>
GENERALIDADES .....	4
1.1 <i>Antecedentes</i> .....	4
1.2 <i>Objetivo General</i> .....	5
1.3 <i>Objetivos Específicos</i> .....	5
1.4 <i>Descripción del Problema que Resuelve</i> .....	6
1.5 <i>Solución Propuesta</i> .....	8
<b>CAPÍTULO 2</b> .....	<b>11</b>
MARCO TEÓRICO .....	11
2.1 <i>Ingeniería Dirigida por Modelos</i> .....	11
2.2 <i>Anomalias de Software</i> .....	15
2.3 <i>Esquemas de Clasificación de Defectos</i> .....	17
2.3.1 <i>IEEE Estándar de Clasificación de Anomalias en Software</i> .....	18
2.3.2 <i>Esquema de Hewlett-Packard</i> .....	19
2.3.3 <i>Clasificación de Defectos Ortogonal (ODC)</i> .....	20
2.4 <i>Impacto de los defectos en la Calidad de Software</i> .....	23
2.5 <i>Análisis de los Datos Obtenidos por la Clasificación de Defectos</i> .....	24
2.6 <i>Conclusiones</i> .....	24
<b>CAPÍTULO 3</b> .....	<b>26</b>
LEVANTAMIENTO DE REQUERIMIENTOS Y DEFINICIÓN DEL PROYECTO .....	26
3.1 <i>Características de los Sistemas de Gestión de Defectos</i> .....	26
3.2 <i>Requerimientos de Sistema</i> .....	28
3.3 <i>Requerimientos de Usuario</i> .....	28
3.4 <i>Roles y Permisos de Usuarios</i> .....	29
3.5 <i>Conclusiones</i> .....	30
<b>CAPÍTULO 4</b> .....	<b>33</b>
DISEÑO DE LA SOLUCIÓN .....	33
4.1 <i>Arquitectura de la Aplicación</i> .....	33
4.2 <i>Infraestructura</i> .....	34
4.2.1 <i>Base de Datos Oracle XE</i> .....	35
4.2.2 <i>Máquina Virtual en la Nube de AWS</i> .....	35
4.2.3 <i>Servidor de Aplicaciones Glassfish</i> .....	36
4.2.4 <i>Tecnología Web</i> .....	36
4.2.5 <i>Conectividad en Aplicaciones Web</i> .....	37
4.3 <i>Diseño de Modelos Conceptuales</i> .....	38
4.3.1 <i>Diseño de Datos</i> .....	39
4.3.2 <i>Diseño de Clases</i> .....	40
4.3.3 <i>Diseño de Casos de Uso</i> .....	41
4.4 <i>Diseño de Interfaces</i> .....	72
4.4.1 <i>Interfaz de Ingreso al Sistema</i> .....	72
4.4.2 <i>Interfaz de Página Principal de la Aplicación</i> .....	72
4.4.3 <i>Interfaz del Menú de Opciones de la Aplicación</i> .....	73
4.4.4 <i>Interfaz Para Listados del Sistema</i> .....	74
4.4.5 <i>Interfaz de Creación y Edición de Registros</i> .....	75

4.4.6 Interfaz de Reportes .....	78
4.5 <i>Diseño del Esquema de Seguridad de la Aplicación</i> .....	79
4.5.1 Autenticación.....	79
4.5.2 Autorización.....	79
4.5.3 Seguridad en la Conexión.....	80
4.6 <i>Diseño de Reportes y Paneles de Control</i> .....	80
4.6.1 Diseño de Panel de Control Principal.....	80
4.6.2 Diseño de Reporte para Proyectos Individuales.....	82
4.7 <i>Mecanismos de Gestión</i> .....	83
4.7.1 Análisis de Causalidad.....	83
4.7.2 Base de Acumulada de Conocimientos.....	88
4.7.3 Medidas de Prevención.....	92
4.8 <i>Conclusiones</i> .....	95
<b>CAPÍTULO 5</b> .....	<b>96</b>
DESARROLLO Y PRUEBAS.....	96
5.1 <i>Planificación del Proyecto</i> .....	96
5.1.1 Recursos.....	96
5.1.2 Calendario de Actividades.....	96
5.1.3 Costos del Desarrollo.....	97
5.1.4 Flujos y Proyección del Proyecto.....	97
5.2 <i>Desarrollo del Sistema de Gestión de Defectos</i> .....	98
5.2.1 Descripción del Proceso de Desarrollo.....	98
5.2.2 Estructura de la Aplicación Web.....	99
5.2.3 Tecnologías Utilizadas para el Desarrollo.....	100
5.2.4 Herramientas Utilizadas en el Desarrollo.....	102
5.3 <i>Despliegue del Sistema de Gestión de Defectos</i> .....	102
5.3.1 Configuración de Entorno.....	102
5.3.2 Optimización de la Aplicación.....	103
5.4 <i>Pruebas de la Aplicación</i> .....	104
5.4.1 Configuración de las Pruebas.....	104
5.4.2 Análisis de Resultados.....	105
5.5 <i>Análisis de Mejoras</i> .....	109
5.6 <i>Conclusiones</i> .....	110
<b>CONCLUSIONES Y RECOMENDACIONES</b> .....	<b>111</b>
<b>GLOSARIO</b> .....	<b>114</b>
<b>REFERENCIAS</b> .....	<b>121</b>

## ÍNDICE DE FIGURAS

FIGURA 1 : Tipos de Defectos de Software (Group, 2013).....	16
FIGURA 2 : Esquema de Clasificación de Hewlett-Packard.....	20
FIGURA 3 : Gráfica de Análisis de Información Bajo Esquema IBM.....	22
FIGURA 4 : Arquitectura de una Aplicación Web JSF .....	33
FIGURA 5 : Conectividad de la Aplicación Web en la Nube. ....	37
FIGURA 6 : Base de Datos del Sistema de Gestión de Defectos.....	39
FIGURA 7 : Clases del Sistema de Gestión de Defectos.....	40
FIGURA 8 : Diagrama de Jerarquías de Actores del Sistema .....	41
FIGURA 9 : Estereotipo <<CRUD>> .....	42
FIGURA 10 : Caso de Uso para Usuario No Registrado.....	42
FIGURA 11 : Diagrama de Caso de Uso para Usuario Registrado .....	43
FIGURA 12 : Diagrama de Caso de Uso para Usuario Registrado .....	45
FIGURA 13 : Diagrama de Casos de Uso de un Colaborador.....	66
FIGURA 14 : Diseño de Página de Inicio de Sesión.....	72
FIGURA 15 : Diseño de Panel de Control.....	73
FIGURA 16 : Diseño de Menú de Opciones del Sistema.....	74
FIGURA 17 : Diseño de Listado de un Mantenimiento del Sistema. ....	75
FIGURA 18 : Diseño de Formulario de Edición de una Empresa. ....	76
FIGURA 19 : Diseño de Formulario de Edición de un Proyecto.....	76
FIGURA 21 : Diseño de Formulario de Edición de un Esquema.....	77
FIGURA 20 : Diseño de Formulario de Modelo de Calidad. ....	78
FIGURA 22 : Diseño de Formulario de Generación de Reportes. ....	79
FIGURA 23 : Gráficas Estadísticas de la Cuenta de Usuario. ....	80
FIGURA 24 : Gráfica De Defectos Encontrados vs Solucionados por Día. .	81
FIGURA 25 : Gráfica de Proyectos y Datos de la Cuenta.....	81
FIGURA 26 : Gráfica de Defectos Encontrados Versus Solucionados.....	82
FIGURA 27 : Análisis por Atributos de Calidad. ....	83
FIGURA 28 : Diagrama de Causalidad Típica de un Proyecto Informático .	84
FIGURA 29 : Tipos de Relaciones entre Defectos.....	85
FIGURA 30 : Ejemplo de Red de Defectos.....	85
FIGURA 31 : Ejemplo de Red Aislada de Defectos.....	87
FIGURA 32: Opción de Búsqueda de Soluciones de un Defecto. ....	89
FIGURA 33 : Defectos Similares Encontrados .....	90
FIGURA 34 : Estructura de un Proceso de Solución de Defecto .....	91
FIGURA 35 : Diseño de Creación de una Medida de Prevención.....	93
FIGURA 36 : Listado de Medidas de Prevención.....	94
FIGURA 37 : Estructura de la Aplicación Basada en Plantilla Maven.....	100
FIGURA 38 : Tiempo Promedio de Respuesta de la Página.....	105
FIGURA 39 : El Tiempo Promedio Respuesta por Petición.....	106

FIGURA 40 : Peticiones por Segundo .....	106
FIGURA 41 : Tráfico.....	107
FIGURA 42 : Tiempo de Respuesta Versus Carga .....	108
FIGURA 43 : Errores de la Aplicación para 15 Usuarios Simultáneos.....	109

## ÍNDICE DE TABLAS

TABLA 1: Atributos del Esquema de Clasificación de la IEEE.....	19
TABLA 2: Esquema de Clasificación de IBM (Freimut, 2001).....	21
TABLA 3 : Usuario No Registrado .....	41
TABLA 4 : Usuario Registrado.....	41
TABLA 5 : Usuario Colaborador .....	41
TABLA 6 : Caso de Uso Creación de una Cuenta de Usuario.....	43
TABLA 7 : Caso de Uso Autenticar el Usuario.....	44
TABLA 8 : Caso de Uso Cambiar de Clave de Usuario.....	45
TABLA 9 : Caso de Uso Crear una Empresa.....	46
TABLA 10 : Caso de Uso Listar las Empresas .....	46
TABLA 11 : Caso de Uso Editar una Empresa .....	47
TABLA 12 : Caso de Uso Eliminar una Empresa.....	47
TABLA 13 : Caso de Uso Imprimir Empresas.....	48
TABLA 14 : Caso de Uso Crear un Proyecto .....	48
TABLA 15 : Caso de Uso Listar los Proyectos.....	49
TABLA 16 : Caso de Uso Editar un Proyecto.....	50
TABLA 17 : Caso de Uso Eliminar un Proyecto.....	50
TABLA 18 : Caso de Uso Imprimir Proyectos .....	51
TABLA 19 : Caso de Uso Crear un Esquema de Defectos.....	51
TABLA 20 : Caso de Uso Listar Esquemas de Defectos.....	52
TABLA 21 : Caso de Uso Editar un Esquema de Defectos .....	52
TABLA 22 : Caso de Uso Eliminar un Esquema de Defectos .....	53
TABLA 23 : Caso de Uso Imprimir Esquemas de Defectos.....	53
TABLA 24 : Caso de Uso Crear Atributos de Esquema .....	54
TABLA 25 : Caso de Uso Editar un Atributo del Esquema .....	55
TABLA 26 : Caso de Uso Eliminar un Atributo del Esquema .....	55
TABLA 27 : Caso de Uso Crear un Modelo de Calidad .....	56
TABLA 28 : Caso de Uso Listar los Modelos de Calidad .....	57
TABLA 29 : Caso de Uso Editar un Modelo de Calidad .....	57
TABLA 30 : Caso de Uso Eliminar un Modelo de Calidad .....	58
TABLA 31 : Caso de Uso Imprimir Modelos de Calidad.....	58
TABLA 32 : Caso de Uso Crear Atributo de Modelo .....	59
TABLA 33 : Caso de Uso Editar un Atributo de Modelo .....	60
TABLA 34 : Caso de Uso Eliminar un Atributo de Modelo.....	60
TABLA 35 : Caso de Uso Crear un Defecto .....	61
TABLA 36 : Caso de Uso Listar los Defectos.....	61
TABLA 37 : Caso de Uso Editar un Defecto.....	62
TABLA 38 : Caso de Uso Eliminar un Defecto.....	63
TABLA 39 : Caso de Uso Imprimir Defectos .....	63

TABLA 40 : Caso de Uso Listar Colaboradores .....	63
TABLA 41 : Caso de Uso Solicitar un Colaborador .....	64
TABLA 42 : Caso de Uso Eliminar un Colaborador .....	65
TABLA 43 : Caso de Uso Asignar de Defecto a Colaborador .....	66
TABLA 44 : Caso de Uso Analizar la información de Defectos .....	66
TABLA 45 : Caso de Uso Aceptar Solicitud de Colaboración .....	67
TABLA 46 : Caso de Uso Listar Defectos Asignados.....	68
TABLA 47 : Caso de Uso Crear una Solución .....	68
TABLA 48 : Caso de Uso Listar los Soluciones.....	69
TABLA 49 : Caso de Uso Editar Solución .....	69
TABLA 50 : Caso de Uso Eliminar una Solución.....	70
TABLA 51 : Caso de Uso Imprimir Soluciones .....	70
TABLA 52 : Caso de Uso Enviar una Solución .....	71
TABLA 53 : Orden Óptimo para Solucionar Red de Defectos.....	86
TABLA 54 : Orden Óptimo para Solucionar Red Aislada .....	88
TABLA 55 : Calendario de Trabajo .....	96
TABLA 56 : Costos de Desarrollo .....	97
TABLA 57 : Inversión Inicial del Proyecto .....	97
TABLA 58 : Flujo Proyectado para 8 años.....	98
TABLA 59 : Micro Instancias de Bajo Costo de Nube Elástica de Amazon	103
TABLA 60 : Resultados Estadísticos del Tiempo Promedio de Respuesta	105
TABLA 63 : Resultados Estadísticos del Tiempo Promedio de Petición ....	106
TABLA 62 : Resultados Estadísticos de las Peticiones por Segundo .....	107
TABLA 63 : Resultados Estadísticos del Tráfico .....	107

# INTRODUCCIÓN

Actualmente el proceso de desarrollo de software ha sufrido muchos cambios con la aparición de nuevas tecnologías, una de estas es la Ingeniería Basada en Modelos (MDE, Model-Driven Engineering), utilizada en la construcción de sistemas de información. MDE permite la generación de aplicaciones partiendo de modelos conceptuales lo que mejora la productividad y facilita su mantenimiento. Sin embargo la rigurosidad y efectividad que deben presentar estos modelos juegan un papel muy importante en el desarrollo de aplicaciones.

MDE va más allá que las simples actividades de desarrollo, de hecho combina otras tecnologías como por ejemplo el uso de lenguajes de modelado específicos de dominio y generadores de códigos o transformadores de modelos. Los lenguajes de modelado se basan en meta-modelos (modelos de modelos), permitiendo la descripción de la estructura, comportamiento y requisitos de un sistema. Por otro lado, las transformaciones de modelos permiten obtener código de aplicación (transformaciones modelo-texto) o bien a otros modelos (transformaciones modelo-modelo).

Los desarrolladores tienen el reto de crear modelos conceptuales que permitan la generación de código de aplicación, por lo tanto el registro, seguimiento y resolución de defectos en el proceso de desarrollo de software bajo el paradigma MDE tiene un papel importante en el desarrollo de sistemas de información. La calidad del software en sistemas construidos bajo MDE está directamente relacionada con el número de defectos que tenga sus modelos

conceptuales en las fases de modelado y diseño. Existen varias herramientas de software utilizadas para la detección de defectos de forma que presenten a los desarrolladores una retroalimentación de la información de defectos encontrada.

Sin embargo cabe señalar que muchos de estos sistemas realizan un análisis puramente sintáctico y no dicen con exactitud si el software podrá satisfacer todas las necesidades del cliente. Por lo antes señalado es necesaria la creación de una herramienta que permita el registro, clasificación y análisis de defectos encontrados al desarrollar productos informáticos. Esta herramienta permitirá dar soluciones oportunas y generar sistemas de calidad.

Los errores que se comenten en las fases de diseño y modelado de sistemas de software tienen un impacto fundamental en el resto de las etapas de desarrollo. El hecho de mejorar la calidad de modelos conceptuales también mejora la calidad del producto final. Esto se puede observar claramente al utilizar ingeniería basada en modelos.

Se plantea la implementación de un Sistema de Gestión de Defectos de Software que permita registrar los defectos, organizar su información, analizar sus causas e influir en la forma que los desarrolladores toman decisiones en las etapas de modelado y diseño de sistemas. Con esta herramienta los diseñadores, jefes de proyectos y en general todo el equipo de trabajo podrán realizar un seguimiento de los defectos encontrados en el proceso de desarrollo y utilizarlos como herramienta para solucionar y gestionar errores.

Gracias a la evolución de las tecnologías web se plantea la creación de una solución que tome ventaja de las mismas. Por lo que se desarrollará la solución bajo el esquema de Software como Servicio (SAS, Software as Service), esto permitirá sacar el mayor provecho a las oportunidades de desarrollo sin mayor inversión en infraestructura ni en mantenimiento. Además con las tecnologías basadas en Web 2.0, es posible llegar a todos usuarios directamente y ofrecerles un producto de bajo costo y altas prestaciones.

Al ser un sistema web es posible una fácil integración con los sistemas de información internos de la empresa. Gracias a la implementación de esta propuesta las empresas desarrolladoras de software podrán contar con una herramienta que registre, organice, clasifique, resuelva y gestione la información de los defectos de software encontrados al desarrollar productos informáticos.

# CAPÍTULO 1

## Generalidades

### 1.1 Antecedentes

El estudio de defectos de software no es nuevo, ya que a través del tiempo se han realizado investigaciones y planteamientos en este campo. Este trabajo parte de conceptos teóricos ya desarrollados y propone un sistema de información que ponga en práctica esta teoría. La definición de anomalías de software y el desarrollo de esquemas de clasificación de defectos son fundamentales para la realización de esta tesis. Ya que dan el soporte teórico y las pautas en el desarrollo del sistema de gestión de defectos propuesto.

Los esquemas de clasificación de defectos han sido desarrollados por varios actores e implementados por varias empresas. Entre los más importantes se pueden mencionar: el estándar de clasificación de anomalías de software presentado por la IEEE, el esquemas de clasificaciones planteado por Hewlett-Packard y el esquema Ortogonal son de los más conocidos. Estos esquemas buscan aplicar una clasificación a defectos de software encontrados en las etapas de desarrollo, la diferencia entre los esquemas mencionados se debe a que cada autor creó un esquema que responde a sus necesidades y metodologías de desarrollo. Otro de los esquemas de clasificación planteado es el esquema de clasificación multidimensional de defectos para esquemas conceptuales planteado por Fernanda Granda Juca para su tesis doctoral en la Universidad Politécnica de

Valencia, este estudio y los anteriores han sido una base fundamental ya que muchos de estos son llevados a la práctica a través de este proyecto de tesis.

## **1.2 Objetivo General**

Implementar un sistema de información que permita gestionar los defectos encontrados en el diseño y modelamiento conceptual de proyectos informáticos que utilizan herramientas de ingeniería en software asistida por computador.

## **1.3 Objetivos Específicos**

- Establecer las bases teóricas y metodológicas que permitan la implementación de un sistema de información para la gestión de defectos en proyectos informáticos. Se analizará los conceptos relacionados con los defectos de software, tipos de defectos, esquemas de clasificación y las metodologías de análisis existentes.
- Analizar y definir los requerimientos de los sistemas de información para la gestión de defectos de software. Se analizará los servicios, restricciones y metas del sistema, además se especificará los requerimientos de hardware, software, las herramientas utilizadas y las tecnologías que serán empleadas en el desarrollo de la aplicación.
- Establecer un arquitectura completa del sistema a través de la definición de modelos conceptuales y diseños previos a la construcción de la solución, se especificará los usuarios, la seguridad y la arquitectura utilizada para la construcción de un sistema de información para la gestión de defectos.

- Validar la funcionalidad del sistema de información para la gestión de defectos a través de una serie de pruebas que evalúen los diferentes componentes de la aplicación y se definirá las posibles mejoras.

#### **1.4 Descripción del Problema que Resuelve**

Hoy en día el proceso de desarrollo de software ha sufrido cambios gracias a la aparición de nuevas tecnologías, una de estas es la Ingeniería Basada en Modelos (MDE, Model-Driven Engineering) que es una técnica muy utilizada para la construcción de sistemas de información. MDE permite la generación de aplicaciones partiendo de modelos conceptuales, por lo que se mejora la productividad y se facilita el mantenimiento. Sin embargo bajo esta metodología de desarrollo la rigurosidad y efectividad que deben presentar estos modelos juegan un papel importante.

Se podría decir que independientemente de la metodología utilizada, los errores que se cometen en las fases de diseño y modelado del proceso de desarrollo, tienen un impacto fundamental en el resto de las etapas. El hecho de mejorar la calidad de los modelos conceptuales también mejora la calidad del producto final, lo cual nos brinda oportunidades de optimización y ayuda a prevenir la aparición de defectos en futuras etapas de desarrollo.

MDE va más allá que las simples actividades de desarrollo, de hecho combina otras tecnologías como por ejemplo: el uso de lenguajes de modelado específicos de dominio y generadores de código o transformadores de modelos. Por un lado, los lenguajes de modelado se basan en meta-modelos (modelos de modelos),

permitiendo la descripción de la estructura, comportamiento y requisitos de un sistema. Por otro lado, las transformaciones de los modelos permiten la obtención de código fuente (transformaciones modelo-texto), o bien a otros modelos (transformaciones modelo-modelo).

Los desarrolladores tienen el reto de crear modelos conceptuales que permitan la generación eficiente de sistemas de información, por lo cual el registro, seguimiento y resolución de defectos encontrados en los procesos de desarrollo de software es fundamental y aún más bajo el paradigma MDE que representa el objeto de estudio de este proyecto de tesis. La calidad del software en los sistemas construidos bajo MDE está directamente relacionada con el número de defectos que tenga sus modelos conceptuales.

Existen varias herramientas de software utilizadas para la detección de defectos, que por lo general presentan a los desarrolladores una retroalimentación continua sobre los defectos encontrados. Cabe señalar que muchos de estos sistemas realizan un análisis puramente sintáctico y no dicen con exactitud si el software podrá satisfacer las necesidades del usuario final.

Por lo antes señalado es necesario la creación de una herramienta que permita el registro, la clasificación y el análisis de defectos encontrados en el desarrollo de proyectos informáticos, de forma que se optimice el código, se evite errores futuros y se mejore la calidad de los sistemas de información desarrollados.

## 1.5 Solución Propuesta

Se plantea la implementación de un Sistema de Gestión de Defectos de Software que sirva especialmente para su utilización con herramientas de ingeniería asistida por computadora (CASE), el cual se empleó para realizar el registro y análisis de defectos encontrados en la fases de diseño y modelado conceptual. Esta herramienta permitirá registrar defectos, analizar sus causas, e implementar medidas que prevengan su ocurrencia.

Con esta herramienta los diseñadores, jefes de proyectos y en general todo el equipo de desarrolladores podrán realizar un seguimiento y análisis de los defectos encontrados en los procesos de desarrollo de software. Se tomará como base fundamental la implementación de esquemas de clasificación de defectos, por lo que el sistema debe presentar flexibilidad y adaptabilidad ya que los esquemas podrían variar en el tiempo y de un proyecto a otro.

Gracias a la evolución de las plataformas y tecnologías web utilizadas para el desarrollo de aplicaciones, se plantea la creación de una solución que aproveche estas capacidades. Por lo que se desarrollará la solución bajo una plataforma web en la nube, ya que permitirá sacar el mayor provecho a las oportunidades de desarrollo, sin mayor inversión en infraestructura ni en mantenimiento. Además con la aparición de las tecnologías basadas en Web 2.0, es posible llegar directamente a los posibles usuarios y ofrecerles un producto de bajo costo y altas prestaciones.

Para el desarrollo del sistema de información se utilizarán JavaServer Faces (JSF) que es una tecnología y framework para aplicaciones Java basadas en la web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java empresariales. Bajo esta plataforma se empleará Richfaces ya que permite el uso de librerías de componentes especializados que utilizan AJAX (Javascript Asíncrono y XML) y finalmente se utilizara HTML5 para dar interacciones al sistema web. Como herramienta de desarrollo se utilizará Netbeans IDE para la codificación, Subversion en el control de versiones e Hibernate como manejador de persistencia de la aplicación. El sistema se implementará bajo un servidor virtual Windows Server 2003 a través de los Servicios Web de Amazon. Se utilizará el servidor de aplicaciones Glassfish, y la base de datos será Oracle XE que es una herramienta gratuita utilizada frecuentemente para la generación de prototipos.

Dentro del proyecto se contempla el diseño de una serie de reportes que permitan a los usuarios a través de paneles de control y vistas responder a preguntas como: el cuándo, cómo, porqué y dónde se encontraron los defectos en el desarrollo de un proyecto informático. Este sistema analizará las relaciones entre los defectos y los clasificará de forma que sirvan como una fuente de conocimiento que permita a los administradores o jefes de proyectos identificarlos, clasificarlos y prevenir su ocurrencia.

Gracias a esta información se podrá encontrar y corregir debilidades en las fases del proceso de desarrollo, además permitirá a los directivos tomar decisiones oportunas basadas en la información histórica encontrada. Se podrá comunicar la

información relacionada con los defectos entre el equipo de desarrollo ya que al ser un sistema web tendrá la capacidad de ser integrada fácilmente con otros sistemas informáticos internos de la empresa.

# CAPÍTULO 2

## Marco Teórico

Este capítulo presenta la definición de los principales conceptos, estándares y teorías necesarias para el desarrollo de un sistema de gestión de defectos, además se presenta conceptos teóricos relacionados con las técnicas de identificación, clasificaciones y análisis de defectos.

### 2.1 Ingeniería Dirigida por Modelos

Actualmente las tecnologías de información y comunicación cumplen un papel esencial en nuestra sociedad, en particular los sistemas de software cumplen funciones tan importantes como la gestión documental, control de asistencia, gestión estratégica, etc. Las tecnologías web basadas en servicios y cloud computing están siendo implementadas exitosamente y presentan nuevas herramientas y tecnologías que permiten el desarrollo de software de una forma eficiente y confiable. Sin embargo a pesar de los avances en los entornos de desarrollo integrados (IDE: Integrated Development Environments), el desarrollo de sistemas de software requiere mucho esfuerzo e inversión de recursos.

Dada la complejidad creciente en el desarrollo de software aparece la ingeniería dirigida por modelos (MDE: Model Driven Engineering), con la intención de disminuir la dificultad en el desarrollo de sistemas de software complejos. MDE es una metodología de desarrollo de software que se enfoca en crear y explotar modelos de dominio (representaciones abstractas del conocimiento y actividades

de un particular dominio de aplicación), se describen a través de modelos en diferentes niveles de abstracción y se basa en herramientas de análisis y transformación. Partiendo de modelos conceptuales abstractos para generar un sistema funcional.

Los modelos son abstracciones de algún aspecto de un sistema, el mismo que puede existir o no. Cada modelo tiene un propósito definido y se usa para permitir la comunicación de un aspecto específico del sistema. MDE usa el lenguaje de modelado unificado (UML: Unified Modeling Language)<sup>1</sup> para representar y construir sus modelos conceptuales.

El estándar UML permite modelar la arquitectura, los objetos, las interacciones, los datos y procesos del ciclo de vida de una aplicación, así como aspectos relacionados con el diseño de componentes incluyendo su construcción y despliegue. Los elementos del modelado UML permiten realizar especificaciones estáticas y dinámicas de un sistema orientado a objetos. Los modelos estáticos incluyen la definición de clases, atributos, operaciones, interfaces y relaciones entre clases, como son: herencia, asociación, dependencia, etc. Los modelos semánticos o del comportamiento son representados a través de diagramas de secuencia y colaboración.

Las tecnologías MDE aparecen como una solución a la complejidad que representa desarrollar sistemas de software, al principio se trató de abstraer esta complejidad con la aparición de los primeros lenguajes de programación como

---

<sup>1</sup> UML: Lenguaje de Modelado Unificado estándar es una técnica estándar para la especificación sistemas en todas sus fases.

Fortran y el lenguaje ensamblador, que reducían la complejidad de la codificación. Después aparecieron las herramientas de ingeniería computacional asistida (CASE: Computer-Aided Software Engineering), las cuales entregaron a los desarrolladores herramientas de diseño en términos de representaciones gráficas de propósito general como: diagramas de estructuras, flujo de datos; Estos permiten el uso de convenciones y además reducen la codificación y depuración manual. Los problemas se dieron gracias a la alta dificultad para integrar el código generado con otras plataformas y la falta de opciones de personalización del código generado, lo que significó que esta tecnología no se lleve a la práctica.

Las plataformas que hoy son utilizadas para el desarrollo de aplicaciones presentan una mayor complejidad en su aplicación, como son: J2EE, Punto Net y CORBA, las cuales nos ofrecen librerías y dependencias que requieren un razonable esfuerzo de configuración. Sin mencionar la implementación de sistemas grandes con arquitectura distribuida, que requieran integración a nivel de componentes. MDE se presenta como la solución a estos problemas ya que combina las siguientes tecnologías (Shmidt, 2006):

- Lenguaje de modelado de dominio específico (DSML: Domain-specific modeling languages). DSML utiliza meta modelos para definir las relaciones entre los conceptos de un dominio, donde especifica con precisión la semántica y las restricciones de asociación con el objetivo de formalizar la estructura de la aplicación, el comportamiento y las necesidades dentro de dominios particulares.

- Motores de Transformación y Generadores. Los cuales toman un aspecto del modelo para producir artefactos como son: código fuente, entradas de simulación, descriptores XML, etc. De forma que asegura la consistencia entre la implementación de aplicaciones y la funcionalidad que es representada en los modelos.

La evolución de la tecnología MDE se puede observar a través del desarrollo de DSML, que ajusta con precisión la sintaxis y la semántica del dominio vía el meta modelamiento, el cual permite a través de elementos gráficos familiares reducir la curva de aprendizaje y ayuda a que los arquitectos aseguren que el software producido cumplirá las expectativas del usuario. Gracias a esto los desarrolladores e ingenieros están en capacidad de realizar análisis, validación y verificación de los modelos en etapas iniciales de desarrollo para evitar errores en el futuro.

MDE permite maximizar la compatibilidad entre sistemas a través del reúso de modelos estándares, simplifica los procesos de diseño para lo cual usa modelos y diseños de patrones recurrentes en el dominio de la aplicación y también promueve la uniformidad del trabajo del equipo de desarrollo a través de la estandarización de la terminología y las establecimiento de las mejores prácticas. (Jubair Basha, 2011)

MDE tiene por objeto aumentar la rentabilidad de una empresa disminuyendo el esfuerzo en el desarrollo de software. Este beneficio se entrega en dos formas básicas: mediante la mejora de la productividad a corto plazo de los desarrolladores, incrementando el valor de los primeros artefactos de software y

mediante la mejora de la productividad a largo plazo con lo cual se evita la rápida obsolescencia en los objetos primarios de software. (Colin Atkinson, 2003)

Entre las ventajas del uso de MDE se pueden mencionar:

- Reduce la sensibilidad al cambio de los primeros objetos producidos.
- Garantizar que la información debe ser de fácil acceso para cualquier persona aunque no sean los creadores del artefacto.
- Mantener un bajo impacto en el sistema por los cambios en los requerimientos.

## 2.2 Anomalías de Software

Defectos, anomalías, fallas y errores pueden ser introducidos en cualquier etapa del desarrollo de software, por ejemplo: por errores humanos al momento de la codificación, al realizar los modelos conceptuales y en la etapa de implementación de un sistema. Llegar a desarrollar un sistema de software sin ningún defecto es muy difícil y es el resultado de un proceso de constantes pruebas, depuraciones y correcciones.

Se puede decir que un sistema que presente una cantidad despreciable de defectos es de alta calidad. A través de los años se ha hablado mucho de los defectos en software y la importancia que tiene en las etapas de desarrollo. Existen estándares definidos que hablan de las metodologías para la detección y clasificación de anomalías en software. Antes de profundizar estas metodologías

se debe definir algunos conceptos importantes que son presentados en el siguiente diagrama.

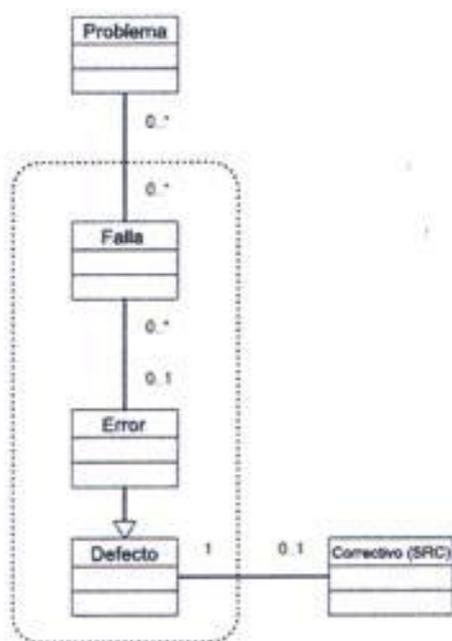


FIGURA 1 : Tipos de Defectos de Software (Group, 2013).

Como se puede observar en la figura anterior los defectos podrían ser removidos a través de una solicitud de cambio correctivo (SCR: Corrective Change Request). Un error es el resultado de un defecto que no se detectó hasta la ejecución del software. Uno o muchos errores conforman una falla y finalmente un conjunto de fallas forman un problema. Los defectos tienen un ciclo de vida bien definido, primero están en un estado de introducción, luego pasan a un estado de detección y finalmente son removidos.

Han existido varios intentos de clasificar los defectos de software y se ha desarrollado un estándar para la clasificación de anomalías por la IEEE, muchas de las clasificaciones presentan atributos y análisis similares, pero ninguna de

estas ha sido aceptada como una herramienta genérica que debería ser utilizada en todos los proyectos informáticos.

Se puede dividir las investigaciones en tres tipos de categorías: (1) taxonomías de defectos, (2) análisis de la causa primaria y (3) clasificaciones de defectos. (Wagner, 2008) La taxonomía de defectos está relacionada con la categorización de fallas, por ejemplo: errores en las declaraciones de las variables o un erróneo manejo de interrupciones. La segunda categoría es análisis de la causa qué produjo un error, como su nombre lo indica se busca el origen del error y prevenir su recurrencia. Finalmente los esquemas de clasificación que a través de categorizaciones permiten organizar la información de defectos y realizar análisis.

### **2.3 Esquemas de Clasificación de Defectos**

Los esquemas de clasificación de defectos tratan de categorizar los defectos encontrados en el desarrollo de un sistema de software, identificar los atributos necesarios que definen estos defectos y poder realizar análisis sobre ellos. Muchas empresas e investigadores han realizado estudios tratando de plantear un esquema de clasificación completo y eficiente pero ninguno de ellos ha sido aceptado y llevado oficialmente a la práctica. Actualmente existe un estándar para la Clasificación de Anomalías de Software desarrollado por: IEEE, HP, IBM y otros los cuales son aplicables a situaciones particulares.

La estructura de los esquemas de clasificación varía en función de sus atributos y la forma en que se relacionan entre sí. Por lo general estos esquemas

tienen la forma de un árbol organizado jerárquicamente. Existen serios problemas con los esquemas de clasificación ya que muchos tienen demasiados atributos, son ambiguos, algunos atributos guardan la misma información, mezclan causas y síntomas. Algunas de las propiedades que deben contener los esquemas de clasificación para que no presenten los problemas citados anteriormente son (Freimut, 2001):

- Los esquemas de clasificación deben tener atributos y valores, por lo cual solamente un valor es apropiado para un defecto específico.
- Los valores de los atributos deben ser completos de forma que para todos los defectos se puede seleccionar un valor de atributo apropiado.
- Es necesario que un esquema presente pocos atributos para facilitar su seguimiento.
- Los atributos deben estar bien definidos dentro de sus descripciones para no dar lugar a confusiones.

Estas características permiten que los esquemas de clasificación estén menos propensos a sufrir errores y facilitan su implementación.

### **2.3.1 IEEE Estándar de Clasificación de Anomalías en Software.**

Este estándar fue presentado en 1994 por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE: Institute of Electrical and Electronics Engineers). El estándar organiza los atributos del esquema en 4 pasos: El primer paso es el reconocimiento del defecto, el segundo es el proponer una solución, el tercer paso es la acción que es el plan donde se resuelve el problema. Finalmente la

terminación del proceso donde todas las acciones han sido completadas. (Freimut, 2001)

Estado del Proceso	Nombre de los Atributos	Significado de los Atributos
<b>Reconocimiento</b>	Actividad del Proyecto	¿Qué estabas haciendo cuando ocurrió el defecto?
	Fase del Proyecto	¿En fase del ciclo de vida estaba el producto?
	Causa Sospechada	¿Cuál se piensa que podría ser la causa?
	Frecuencia	¿Puede este defecto aparecer más de una vez?
	Síntoma Estatus del Producto	¿Cómo se manifestó el defecto? ¿Qué tan usable es el producto si no se realizan cambios?
<b>Investigación</b>	Causa Actual	¿Que causa que la anomalía ocurra?
	Fuente	¿Dónde se originó el defecto?
	Tipo	¿Qué tipo de defecto/mejora a nivel de código?
<b>Acción</b>	Resolución	¿Qué hacer para prevenir que el defecto pase nuevamente?
	Acciones Correctivas	¿Qué acción tomar para resolver el defecto?
<b>Identificación del Impacto</b>	Severidad	¿Cuán malo fue el defecto en términos de ingeniería?
	Prioridad	Clasificar la importancia de resolver el defecto.
	Valor del Cliente	¿Cuán importante es el arreglo para el usuario?
	Misión de Seguridad	¿Cuánto afecto el defecto a los objetivos del proyecto?
	Calendario del Proyecto	Efecto relativo en el calendario del proyecto
	Costo del Proyecto	Efecto relativo en el presupuesto del proyecto
	Riesgo del Proyecto	Riesgo asociado con la implementación de la solución
	Calidad del Proyecto	Impacto del producto sobre la calidad y confianza para hacer el arreglo.
	Socialización	Impacto en la sociedad de implementación de la solución.
<b>Disposición</b>	Disposición	Que paso para cerrar la anomalía?

TABLA 1: Atributos del Esquema de Clasificación de la IEEE

### 2.3.2 Esquema de Hewlett-Packard

Este esquema fue desarrollado en 1986, trata de reducir el número de defectos sobre el tiempo y los agrupa en tres fases: Origen, Tipo y Moda, la

primera representa el lugar donde el defecto fue introducido, la segunda es el tipo que guarda las características específicas del defecto y la tercera es la moda que es la representación del estado. El resultado de la fase en la que se encuentra el defecto depende de los valores asignado a la fase anterior. (Freimut, 2001)

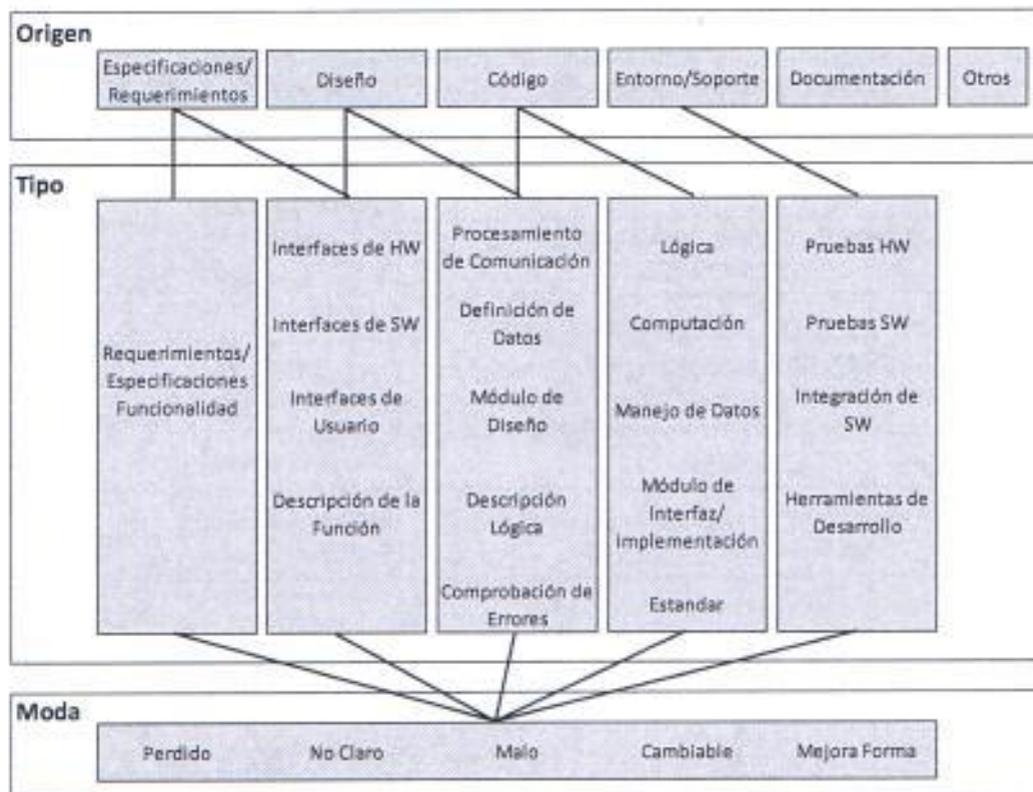


FIGURA 2 : Esquema de Clasificación de Hewlett-Packard

### 2.3.3 Clasificación de Defectos Ortogonal (ODC)

Fue desarrollada por IBM (International Business Machines) en 1996 y trata de dar una retroalimentación al equipo del proyecto sobre la calidad del producto desarrollado. El ciclo de vida de los defectos se los divide en dos procesos que son: el proceso de apertura donde se descubre el defecto, se detecta el disparador

y el impacto, y el cierre que se da cuando el defecto se ha sido corregido y el reporte ha finalizado. (Freimut, 2001)

A continuación se presenta los principales atributos de este esquema:

Proceso	Nombre del Atributo	Significado del Atributo	Valor del Atributo
Abierto	Actividad	Cuando se detecto el defecto?	Inspección del diseño, código de incción, pruebas unitarias, prueba de integración, pruebas del sistema.
	Disparador	Como detectaron el defecto	Diseño Conformidad, Logica / Flujo, compatibilidad con versiones anteriores, compatibilidad Lateral, concurrencia, documento interno, Dependencia Idioma, efecto secundario, Situacion anormal.
			Camino sencillo, ruta compleja, Cobertura, Variación, secuenciación, Interaction
Impacto	Que podria haber notado el usuario si el defecto se hubiera producido?	Capacidad de instalación, facilidad de mantenimiento, normas, Integridad / Seguridad, Migración, fiabilidad, rendimiento, documentación, requisitos, mantenimiento, usabilidad, accesibilidad, capacidad	
Cerrar	Objetivo	Que entidad de alto nivel fue arreglada?	Requisitos, diseño, código, construcción / paquete / fusión, la Información, la interfaz de usuario
	Fuente	Quien desarrollo el objetivo?	En casa, biblioteca, fuera de origen, portado
	Edad	Cual es la historia del objetivo?	Base, nuevo, re-escrito, re-fijado
	Tipo del Defecto	Que tiene que ser arreglado?	Asignación, control, algoritmo, la función, el tiempo, la interfaz, la relación
	Calificador del Defecto		Perdido, incorrecta, extraño

TABLA 2: Esquema de Clasificación de IBM (Freimut, 2001)

Como se puede ver, muchos de los atributos realizan un análisis semántico con excepción del atributo tipo de defecto y disparador por lo cual son los más importantes del esquema, ya que tratan de categorizar dimensionalmente los defectos y son utilizados para realizar varios análisis.

El tipo de defecto se basa en la naturaleza del cambio que se utiliza para arreglar el mismo. Este permite medir el progreso del producto a través de los procesos de

desarrollo. Mientras el disparador caracteriza el defecto basándose en el evento ejecutado para que el defecto salga a la luz y genere un error. Finalmente el atributo impacto del defecto mide al mismo en términos de la falta de funcionalidad ofrecida al usuario. (Bridge, 1996)

ODC trata de relacionar los tipos de defectos con las actividades de detección (por ejemplo: las pruebas unitarias, pruebas funcionales y pruebas de sistemas) (Freimut, 2001). Ya que cada tipo de defecto debería aparecer de una forma predictiva, como por ejemplo los defectos de tipo funcional debería ir disminuyendo según el avance de las actividades de desarrollo, si no se muestra así y no sigue el patrón estándar se debe investigar más a detalle su origen de esta anomalía.

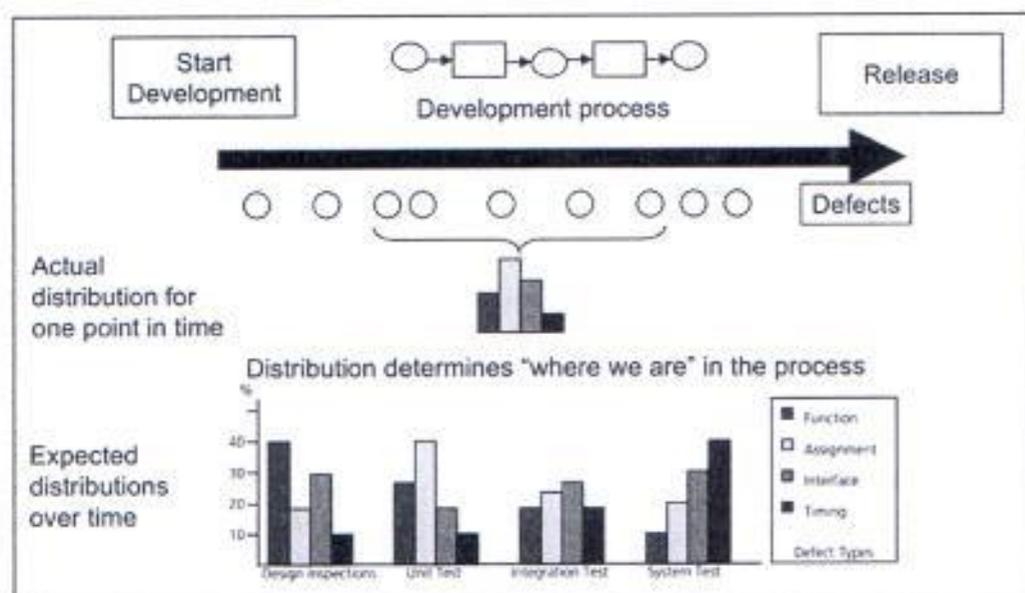


FIGURA 3 : Gráfica de Análisis de Información Bajo Esquema IBM

## 2.4 Impacto de los defectos en la Calidad de Software

La calidad de un sistema informático mide el grado que un producto específico satisface las necesidades de un usuario. Se puede decir que la calidad está directamente relacionada con los defectos ya que mientras mayor es la calidad del software menor es el número de defectos encontrados. Se han elaborado varios modelos de calidad aplicables a sistemas de software entre ellos se encuentra: el modelo de McCall, el modelo de Boehm y el modelo ISO 9126 entre otros. Estos modelos tratan de medir la calidad de un producto de software en función de los valores de sus atributos. La agrupación de atributos de calidad forma un factor calidad que describe la calidad de un sistema en un nivel mayor de abstracción. Si es posible satisfacer la mayor cantidad de factores de calidad se podría decir que un sistema es de alta calidad.

El sistema de información para la gestión de defectos de software propuesto trata de integrar estos conceptos de calidad y relacionarlos directamente. Si partimos de la premisa que cada defecto está relacionado con un atributo de un modelo de calidad, podremos categorizar los defectos por su calidad. Si logramos disminuir los defectos relacionados con un factor de calidad es posible incrementar la calidad total del sistema ya que la misma se encuentra definida por el conjunto total de factores de calidad.

## 2.5 Análisis de los Datos Obtenidos por la Clasificación de Defectos

Al implementar el sistema de gestión de defectos de software planteado en este proyecto de tesis será posible realizar los siguientes análisis a la información almacenada:

1. Análisis histórico de la información de defectos de software que permita describir exactamente cuándo y con qué frecuencia se encontraron defectos en un proyecto de software.
2. Análisis histórico de la información de soluciones presentadas para los defectos de software. Lo cual permite analizar el grado de eficiencia que tiene la empresa al construir proyectos de software. Además estas soluciones permiten crear una base de conocimiento que permita generar soluciones a defectos encontrados posteriormente.
3. Análisis histórico y frecuencia de la aparición de defectos de software categorizado por los atributos del esquema de defectos planteado.
4. Análisis gráfico causal de los defectos encontrados en un proyecto de software.

## 2.6 Conclusiones

Existen muchos estudios previos realizados sobre la definición y clasificación de defectos de software, los mismos que son la base del sistema de gestión de defectos planteado en esta tesis. Los conceptos y definiciones de anomalías de software nos permiten distinguir los tipos de defectos que existen y nos presentan el correcto uso de su terminología.

Los esquemas de clasificación han sido adaptados y definidos por varias empresas a través de los años, por lo cual la diferencia entre cada esquema de clasificación es que responde a las necesidades puntuales de las empresas que los desarrollaron. Los esquemas de clasificación tratan de categorizar los defectos de software de forma que sea fácilmente analizables por lo que se utiliza un esquema de clasificación como medio para registrar, clasificar y analizar defectos.

Es posible a través del análisis de defectos evaluar las tecnologías utilizadas en el desarrollo de sistemas informáticos y encontrar mejoras. El objetivo principal de realizar un análisis de la información de defectos es brindar al equipo de desarrollo herramientas que les permitan encontrar los motivos y patrones que causan la existencia de defectos. Así se puede tomar mejores decisiones al momento de buscar soluciones y evitar su aparición en el futuro.

# CAPÍTULO 3

## Levantamiento de Requerimientos y

### Definición del Proyecto

Este capítulo analiza y define los requerimientos de un sistema de información para la gestión de defectos de software, se define los servicios, restricciones y metas del sistema. Además se especificará los requerimientos de hardware, software, las herramientas utilizadas y las tecnologías que se emplearán en el desarrollo de la aplicación.

#### 3.1 Características de los Sistemas de Gestión de Defectos

La competencia global en el desarrollo de aplicaciones y sistemas empresariales está creciendo de forma acelerada, cada vez más se puede encontrar software especializado, adaptados para diferentes dispositivos y que tratan de resolver problemas comunes de un grupo de usuarios. Hoy en día los sistemas más robustos, más caros y más difíciles de implementar están perdiendo terreno frente a las aplicaciones sencillas, ágiles y con un valor relativamente pequeño. El mercado objetivo hoy en día es la comunidad global y no solamente las empresas locales como fue hace algunos años.

Un sistema moderno de información debe necesariamente poseer un conjunto de características que son dictadas por los avances tecnológicos. Con la aparición de tecnologías como: computación distribuida, computación en la nube y desarrollo de tecnologías móviles, se ha estandarizado los requisitos mínimos que

un sistema computacional debe poseer para ser competitivo. Requisitos como: la capacidad de accesibilidad que nos permite ingresar desde cualquier lugar del mundo, debe ser flexible para adaptarse a diferentes entornos o empresas, además debe tener la capacidad de crecer en sus recursos para poder brindar este servicio a un mayor número de usuarios simultáneos, ofreciendo un servicio veloz de altas prestaciones.

Existen miles de aplicaciones de negocios promocionadas y ofrecidas en los nuevos mercados informáticos, un ejemplo claro de la existencia de estos mercados son las tiendas de aplicaciones como: App Store, Windows Store, Google Play, Adroid Pit entre otras, estas tiendas se presentan ante la comunidad mundial con aplicaciones sencillas, flexibles y a muy bajos precios de lanzamiento. Se pretende que estas aplicaciones sean acogidas por un gran número de usuarios en el mundo de forma que establece una estrategia basada en altos volúmenes de ventas a precios bajos.

La capacidad que tiene un sistema de información de llegar y ofrecer productos informáticos a la comunidad mundial es cada vez más sencilla de implementar, en el caso que sean soluciones para múltiples dispositivo, para un sistema operativo específico o a través del internet. Con el desarrollo de servicios web y aplicaciones en la nube es cada vez más fácil que los nuevos desarrolladores tengan acceso a recursos y herramientas que les permitan crear sistemas web eficientes, flexibles y escalables. Por esta razón se plantea que el Sistema de Gestión de Defectos de Software posea todas las características de los sistemas de web de última generación.

### 3.2 Requerimientos de Sistema

Entre los requerimientos del usuario se plantearon los siguientes:

1. El sistema debe ser accesible desde cualquier lugar del mundo incluso desde computadoras personales, dispositivos celulares.
2. El sistema debe proveer una respuesta rápida y un poder de procesamiento escalable.
3. El sistema debe ser flexible ya que podrá ser configurado para diferentes entornos empresariales.
4. El sistema debe soportar el acceso concurrente de un número creciente de usuarios.
5. El sistema debe ser fácil de utilizar y aprender.
6. El sistema debe ser de fácil de actualizar y dar mantenimiento.

### 3.3 Requerimientos de Usuario

Los requerimientos del sistema son:

1. Un usuario debe tener la capacidad de crear una cuenta en el Sistema de Gestión de Defectos.
2. Un usuario puede crear empresas y proyectos de software dentro de su cuenta.
3. Un usuario puede agregar un conjunto de usuarios para que sean sus colaboradores en los proyectos.
4. Un usuario puede asignar el rol de administrador o desarrollador a cada uno de los colaboradores dentro de su proyecto informático.

5. Un usuario puede configurar un esquema de clasificación de defectos y sus atributos que luego se usarán en varios proyectos informáticos.
6. Un usuario puede definir modelos de calidad relacionados con un esquema de defectos.
7. Un usuario puede registrar los defectos encontrados en un proyecto informático.
8. Un usuario puede obtener reportes que analicen la información de los defectos.
9. Un usuario puede gestionar los defectos encontrados a través de herramientas automatizadas.

### **3.4 Roles y Permisos de Usuarios**

Los permisos y roles específicos de un usuario dentro de un sistema web moderno por lo general toman la forma de un permiso único. Como por ejemplo las aplicaciones de correo electrónico como Hotmail, Gmail y Facebook dan las mismas capacidades a los usuarios dentro de sus sistemas. En el caso de Facebook todos los usuarios tienen la capacidad de publicar una foto y compartirla al resto del mundo, mientras que un usuario externo puede interactuar y ver las fotos del primero solamente si este es parte de alguno de su círculo de amistad. El usuario decide el tipo de permisos que otro tiene sobre sus recursos.

Se plantea usar este tipo de usuario en el sistema de gestión de defectos de forma que pueda compartir información de proyectos con otros usuarios a los cuales se

los denomina colaboradores. Por lo cual un usuario del sistema tiene la capacidad de incluir a otro usuario para que ocupe algún cargo o realice alguna actividad específica dentro de sus proyectos. Este colaborador podría tener diferentes roles para varios proyectos en los que participa. Los roles de usuario pueden ser:

**Administrador.-** Los creadores del proyecto informático son por defecto administradores del proyecto. Un usuario administrador tiene las siguientes capacidades de gestión y administración de los proyectos informáticos.

- Acceder y modificar la información de proyectos.
- Asignar los colaboradores de un proyecto.
- Administrar los defectos encontrados en un proyecto informático.
- Acceder a los reportes y análisis del proyecto.
- Crear de medidas de prevención de defectos.
- Ejecutar de opciones de gestión.

**Usuario.-** Los integrantes que tengan el carácter de colaboradores de un proyecto informático solamente podrán ver los defectos que le han sido asignados. Entre las actividades que un usuario puede realizar están:

- Visualizar los defectos que les han sido asignados por proyectos.
- Pueden responder a estas asignaciones creando soluciones al defecto.

### 3.5 Conclusiones

Gracias a las tecnologías emergentes los estándares y requerimientos necesarios para que un sistema de información sea competitivo han cambiado, hoy en día es necesario que las aplicaciones empresariales respondan a las

nuevas necesidades de los usuarios como son: la accesibilidad total, la facilidad de uso e integración, altas niveles de poder de procesamiento y un producto con bajo costo.

Se ha definido un conjunto de requerimientos de usuarios y de sistemas que son necesarios para realizar la implementación del sistema de defectos de software, estos se basan principalmente en varios factores entre los que están: la accesibilidad ya que un usuario debería poder acceder al sistema desde cualquier lugar del mundo. Esto beneficia a los equipos de desarrollo distribuido en diferentes lugares geográficos, además estos usuarios deberían tener la capacidad de acceder al sistema a través de sus dispositivos móviles y tabletas digitales. Por lo cual se plantea que el sistema sea un sistema web con un formato estándar que permita el ingreso a través de varios dispositivos móviles y que la aplicación esté disponible a la comunidad mundial a través del internet.

La velocidad y bajos tiempos de respuesta es otra de las metas de los sistemas de información modernos, lo cual se puede conseguir si el sistema es cargado en servidores que pertenezcan con servicios de Computación en la Nube<sup>2</sup>, donde es posible escalar los niveles de cargas de procesamiento del sistema en función de los usuarios. Por esta razón se ha seleccionado Amazon Web Services como nuestro proveedor de servicios para la aplicación.

La estructura de roles y permisos de usuarios para el sistema es simple ya que está constituido por los de administrador y de usuario que permiten definir las

---

<sup>2</sup> Computación en la nube es un concepto conocido también bajo los términos servicios en la nube, informática en la nube, nube de cómputo o nube de conceptos, del inglés "Cloud Computing", es un paradigma que permite ofrecer servicios de computación a través de Internet.

capacidades que tienen los usuarios sobre los proyectos en los que participan. Un usuario con un rol administrativo puede realizar cambios en los defectos del proyecto, mientras que un usuario con el rol de tipo usuario únicamente puede trabajar sobre los defectos que se le han sido asignados y crear soluciones.

Se han definido varios requerimientos para describir las capacidades que los usuarios tienen dentro del Sistema de Defectos de Software, y los requerimientos mínimos que debería tener el Sistema de Gestión de Defectos para que cumpla con los estándares mínimos de calidad que posee un sistema web de última tecnología.

# CAPÍTULO 4

## Diseño de la Solución

En este capítulo se realiza el diseño de todos los diagramas y modelos que son la base en la construcción de sistema de gestión de defectos. Además se especifica la arquitectura y metodologías que se emplearán en el diseño del sistema de información.

### 4.1 Arquitectura de la Aplicación

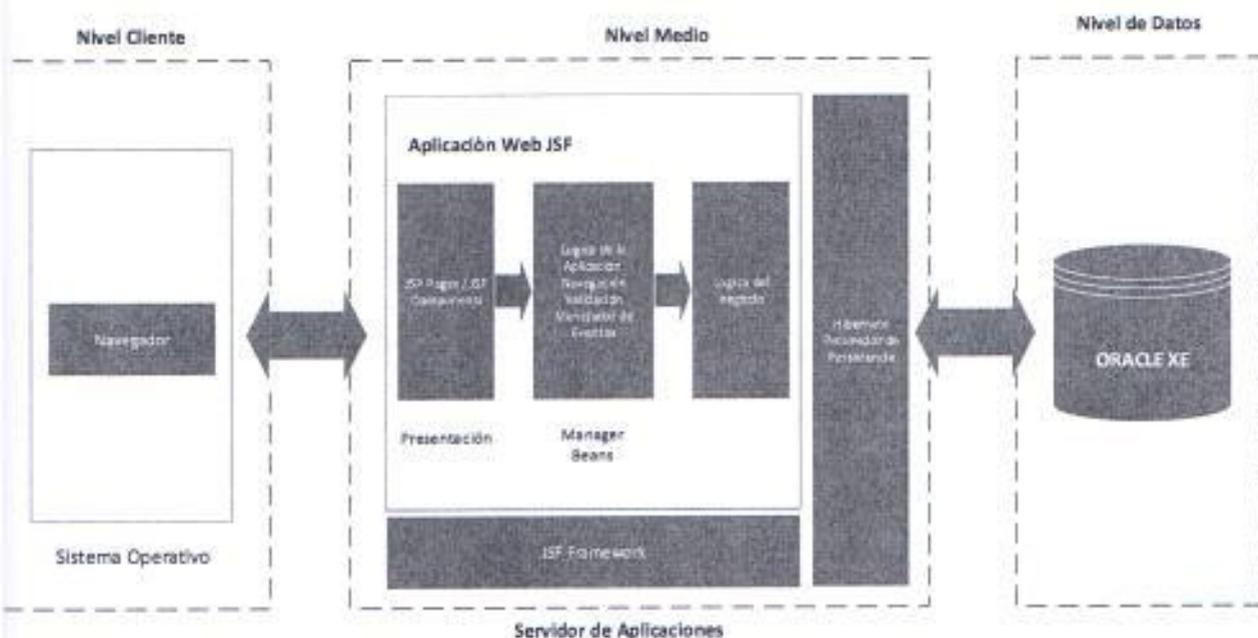


FIGURA 4 : Arquitectura de una Aplicación Web JSF

Las aplicaciones web generalmente son multinivel ya que dividen su funcionalidad en varios niveles y residen en computadoras separadas. En el caso

del Sistema de Gestión de Defectos se incluirá el servidor de aplicaciones y base de datos en un mismo servidor físico por las limitaciones en el presupuesto. La arquitectura de la aplicación está definida por tres niveles que son:

- **Nivel Cliente.-** Que representa cualquier dispositivo que a través de su navegador dentro de su sistema operativo puede acceder al sitio web donde se encuentra la aplicación.
- **Nivel Medio.-** Está representado por el servidor de aplicaciones que es donde se cargan las aplicaciones web. Dentro de la Aplicación JSF se pueden diferenciar tres componentes principales: la presentación en la cual están todas las páginas web JSP y componentes JSF, los POJO's del servidor conocidos como "Managed Beans" y la lógica del negocio.

La aplicación a su vez se conecta a través de un proveedor de persistencia a la base de datos que en este caso es Hibernate.

- **Nivel de Datos.-** Representa la base de datos accedida por la aplicación, para la implementación se utilizará el servidor de base datos Oracle XE donde se almacenará toda la información.

## 4.2 Infraestructura

La infraestructura requerida para el funcionamiento de la aplicación comprende los siguientes componentes:

#### **4.2.1 Base de Datos Oracle XE**

La Base de Datos Oracle es un sistema gestor de base de datos objeto-relacional desarrollado por la Corporación Oracle. Esta base de datos permite el desarrollo de prototipos y aplicaciones pequeñas con el objeto de probar su funcionamiento. Oracle Express Edition es gratuita y es compatible con las ediciones 10g y 11g de la base de datos comercial, esta edición permite usar todas las funcionalidades de las ediciones comerciales. En esta base de datos se pretende almacenar toda la información recolectada por la aplicación.

#### **4.2.2 Máquina Virtual en la Nube de AWS.**

Los Servicios Web de Amazon (AWS), son una colección de servicios de escritorio remoto o servicios web que conforman una plataforma de computación en la nube. Entre el conjunto de productos ofrecidos por Amazon se encuentra el servicio de Nube Elástica (EC2 Amazon Elastic Cloud), que permite adquirir una capacidad informática de tamaño modificable en la nube. Entre las ventajas de usar este servicio para la construcción de aplicaciones empresariales esta:

- Capacidad de controlar completamente el uso de recursos informáticos y pagar solamente por la cantidad de procesamiento utilizado.
- Permite la manipulación completa de instancias de servidor virtual.
- Permite la recuperación de errores gracias a su infraestructura.

### 4.2.3 Servidor de Aplicaciones Glassfish

GlassFish es un servidor de aplicaciones de software libre de Oracle Corporation, que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen estas especificaciones. Es gratuita, de código libre y se distribuye bajo un licenciamiento dual a través de la licencia CDDL<sup>3</sup> y la GNU<sup>4</sup> GPL. La versión utilizada para la aplicación es la versión 4 de Glassfish Server, esta versión es ideal ya que permite el despliegue de aplicaciones con tecnología Java Server Faces versión 2.2.

### 4.2.4 Tecnología Web

Existe varias tecnologías que convergen para construir una aplicación web funcional, estas tecnologías también simplifican la creación de interfaces de usuario para aplicaciones Java EE. JSF permite el desarrollo rápido de aplicaciones web ya que ofrece una arquitectura rica en componentes y abstrae mucho la complejidad al manejar directamente el código. Java Sever Faces incluye las siguientes características:

- Ofrece un conjunto de APIs<sup>5</sup> para representar componentes de una interfaz de usuario, administrar estados, manejar eventos, validar entradas, definir un esquema de navegación, además soporta internacionalización y accesibilidad.

---

<sup>3</sup> Licencia Común de Desarrollo y Distribución. Es una licencia de código abierto y libre, producida por Sun Microsystems, basada en la Mozilla Public License o MPL, versión 1.1.

<sup>4</sup> Licencia Pública General de GNU. Es la licencia que garantiza a los usuarios finales (personas, organizaciones, compañías) la libertad de usar, estudiar, compartir (copiar) y modificar el software.

<sup>5</sup> Interfaz de Programación de Aplicaciones (API).- Es un conjunto de funciones y procedimientos que ofrece una librería para ser utilizado por otro software como una capa de abstracción.

- Posee componentes pre configurados para la creación de interfaces de usuarios.
- Soporta interacciones Ajax.
- Administra JavaBeans<sup>6</sup> y estados.

#### 4.2.5 Conectividad en Aplicaciones Web

Es posible acceder a la aplicación web a través del internet ya que las aplicaciones estarán en uno de los servidores virtuales de Amazon. Por lo cual podrá ser accedido fácilmente con cualquier dispositivo que tenga acceso al internet. Entre los posibles dispositivos que tienen la capacidad de acceder a la aplicación están: teléfonos inteligentes, tablets, computadoras portátiles y de escritorio. A continuación se muestra como se realiza los accesos típicos a aplicaciones ubicadas en la nube.

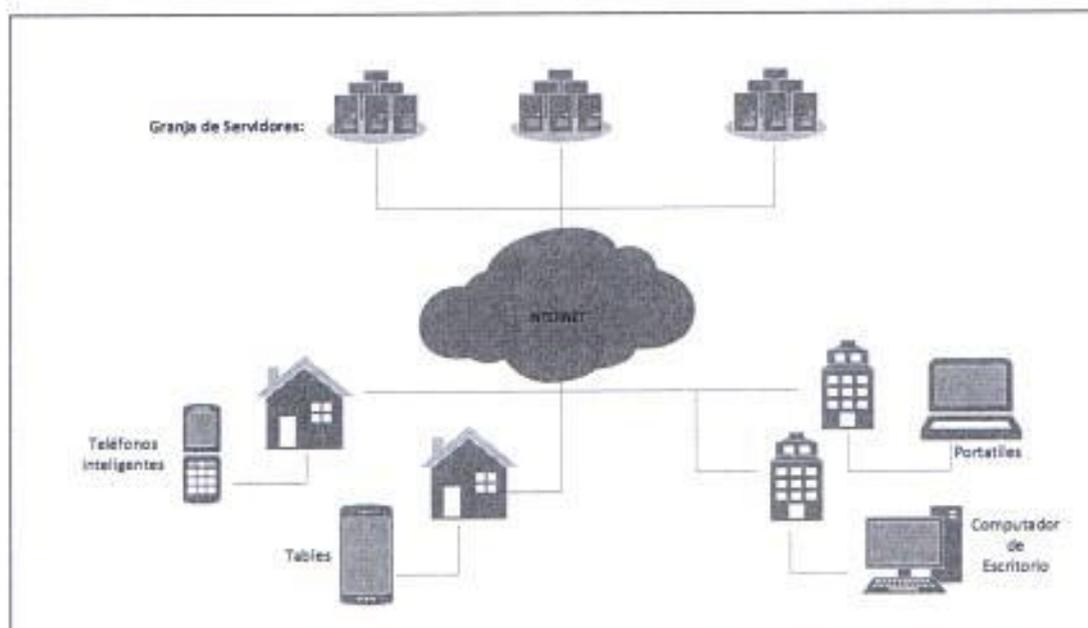


FIGURA 5 : Conectividad de la Aplicación Web en la Nube.

<sup>6</sup> JavaBeans (Beans).- Son modelos de componentes creados para la construcción de aplicaciones Java que se utilizan para encapsular varios objetos en un único objeto.

### 4.3 Diseño de Modelos Conceptuales.

Los diseños permiten definir las características de la aplicación a través de sus diferentes diagramas de diseño. Entre los principales modelos que se utilizarán están:

- Diseño de Datos.
- Diseño de Interfaces.
- Diseño de Clases.
- Diseño de Seguridad.
- Diseño de Casos de Uso.
- Diseño de Reportes.

## 4.3.1 Diseño de Datos

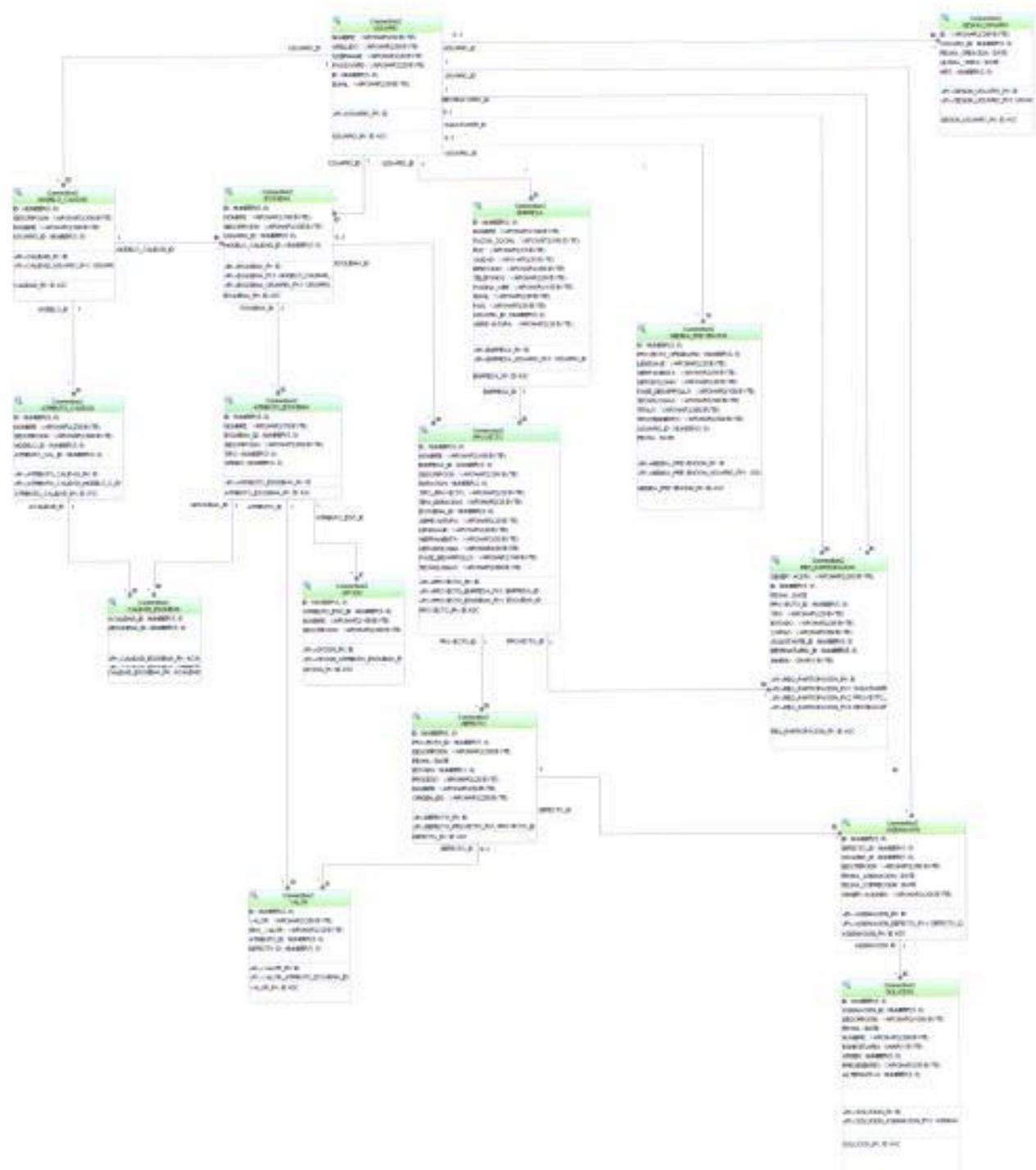


FIGURA 6 : Base de Datos del Sistema de Gestión de Defectos



### 4.3.3 Diseño de Casos de Uso

#### 4.3.3.1 Definición de los Actores del Sistema



FIGURA 8 : Diagrama de Jerarquías de Actores del Sistema

<b>Nombre:</b>	Usuario no Registrado
<b>Descripción:</b> Representa un usuario que no se ha registrado en el sistema, por lo tanto no tiene acceso a los módulos del sistema. Generalmente son posibles usuarios del sistema que pueden ver la página de ingreso del sistema pero no tienen el nombre de usuario o contraseña para iniciar sesión en el sistema.	

TABLA 3 : Usuario No Registrado

<b>Nombre:</b>	Usuario Registrado
<b>Descripción:</b> Representa un usuario que ha registrado sus datos en el sistema y por lo tanto tiene una cuenta de usuario y una contraseña que les permite autenticarse y empezar a utilizar el sistema. Generalmente estos usuarios pueden usar toda la funcionalidad del sistema.	

TABLA 4 : Usuario Registrado

<b>Nombre:</b>	Usuario Colaborador
<b>Descripción:</b> Representa un usuario que es parte del equipo de trabajo dentro de un proyecto de desarrollo de software. Generalmente estos usuarios pueden ingresar a su cuenta de usuario y pueden ver las opciones específicas de colaborador para los proyectos en los que estén involucrados.	

TABLA 5 : Usuario Colaborador

#### 4.3.3.2 Definición de Casos de Uso

Los diagramas de casos de uso nos permiten especificar los requerimientos de usuarios a un mayor nivel de detalle. Para facilitar la representación de un mantenimiento se ha creado el estereotipo <<Crud>> que incluye todas las opciones típicas de creación, borrado, edición e impresión de una entidad. A continuación se presenta la representación de los casos de uso de un mantenimiento en el sistema.

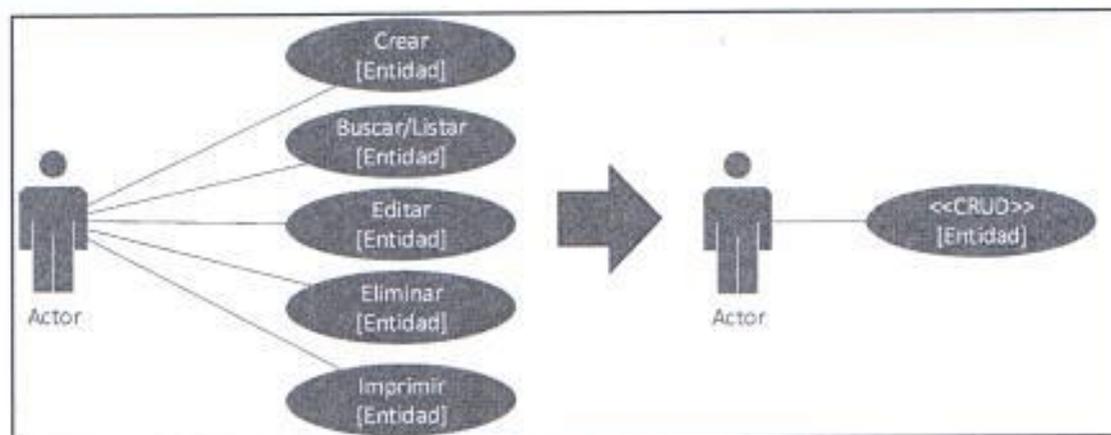


FIGURA 9 : Estereotipo <<CRUD>>

Este estereotipo representa las operaciones básicas que corresponden a un mantenimiento. Estas opciones permiten el registro, creación, modificación y borrado de registros. La gráfica anterior representa el modelamiento de estos diagramas con el estándar definido.



FIGURA 10 : Caso de Uso para Usuario No Registrado.

#### C01: Creación de una Cuenta de Usuario

<b>Código:</b>	C01
<b>Nombre:</b>	Creación de una Cuenta de Usuario
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	7 de noviembre del 2013
<b>Actores:</b>	Usuario no Registrado
<b>Descripción:</b>	Permite crear una nueva cuenta de usuario en el sistema.
<b>Prioridad:</b>	5

<b>Precondiciones:</b>
<b>Flujo Normal:</b> <ol style="list-style-type: none"> <li>1. El usuario ingresa en la opción "Registro de Usuario" del formulario de ingreso al sistema.</li> <li>2. El sistema presenta el formulario para la creación de un nuevo usuario.</li> <li>3. El usuario completa los campos del formulario y presiona el botón registrar.</li> <li>4. El sistema valida la información ingresada.</li> <li>5. El sistema realiza la creación del usuario.</li> </ol>
<b>Flujo Alternativo:</b> <p>4.A.- El sistema encuentra problemas de validación de la información ingresada por el usuario y presenta un mensaje con el error cometido al ingresar la información del nuevo usuario.</p>
<b>Poscondiciones:</b> <p>La cuenta de usuario ha sido creada en el sistema.</p>

TABLA 6 : Caso de Uso Creación de una Cuenta de Usuario

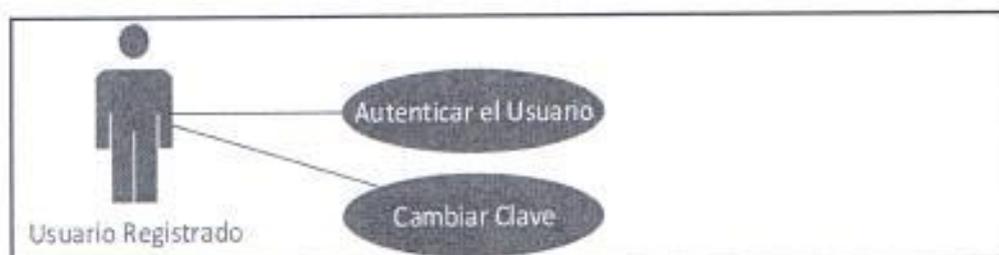


FIGURA 11 : Diagrama de Caso de Uso para Usuario Registrado

C02: Autenticar el Usuario

<b>Código:</b>	C02
<b>Nombre:</b>	Autenticar el Usuario
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	7 de noviembre del 2013
<b>Actores:</b>	Usuario no Registrado
<b>Descripción:</b>	Permite autenticar un usuario ante el sistema.
<b>Prioridad:</b>	5
<b>Precondiciones:</b>	El usuario debe poseer una cuenta de usuario.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. El usuario ingresa a través de un navegador a la dirección de la página de acceso al sistema.</li> <li>2. El sistema presenta el formulario de ingreso.</li> </ol>

3. El usuario ingresa su nombre de usuario y contraseña.
4. El sistema verifica la información ingresada.
5. El sistema re direcciona a la página de inicio del sistema.

**Flujo Alternativo:**

5.A.- El sistema encuentra que la información de autenticación no corresponde a un usuario presenta un mensaje de información con esta información al usuario. Regresa al punto 2.

**Poscondiciones:**

El usuario ha ingresado exitosamente al sistema.

TABLA 7 : Caso de Uso Autenticar el Usuario

C03: Cambiar de Clave de Usuario

<b>Código:</b>	C03
<b>Nombre:</b>	Cambiar clave de usuario
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b> Usuario Registrado	
<b>Descripción:</b> Permite cambiar la clave usada para ingresar al sistema.	
<b>Prioridad:</b> 2	
<b>Precondiciones:</b> El usuario debe poseer una cuenta de usuario.	
<b>Flujo Normal:</b>	
<ol style="list-style-type: none"> <li>1. El usuario accede a la página de ingreso al sistema.</li> <li>2. El sistema presenta una opción en el formulario de ingreso al sistema para cambiar la clave de usuario.</li> <li>3. El usuario selecciona la opción para cambiar la clave de usuario registrado.</li> <li>4. El sistema presenta un formulario donde se requerirá el correo electrónico del usuario.</li> <li>5. El usuario ingresa su correo electrónico.</li> <li>6. El sistema verifica la información ingresada por el usuario.</li> <li>7. El sistema envía un correo electrónico con un enlace para cambiar la clave de usuario.</li> <li>8. El usuario abre su correo electrónico y accede al enlace de cambio de clave.</li> <li>9. El sistema presenta el formulario para el cambio de clave.</li> <li>10. El usuario ingresa su clave nueva y confirmación.</li> <li>11. El sistema verifica que la cumpla con los criterios de seguridad.</li> <li>12. El sistema registra la nueva clave del usuario y re direcciona a la página de ingreso al sistema.</li> </ol>	
<b>Flujo Alternativo:</b>	
<p>11.A.- El sistema encuentra que la clave ingresada por el usuario no cumple con los requerimientos de seguridad por lo que presenta al usuario un mensaje que pide el reingreso de la nueva clave. Regresa al punto 10.</p>	

**Poscondiciones:**

El usuario cambia la clave utilizada para el ingreso al sistema.

TABLA 8 : Caso de Uso Cambiar de Clave de Usuario

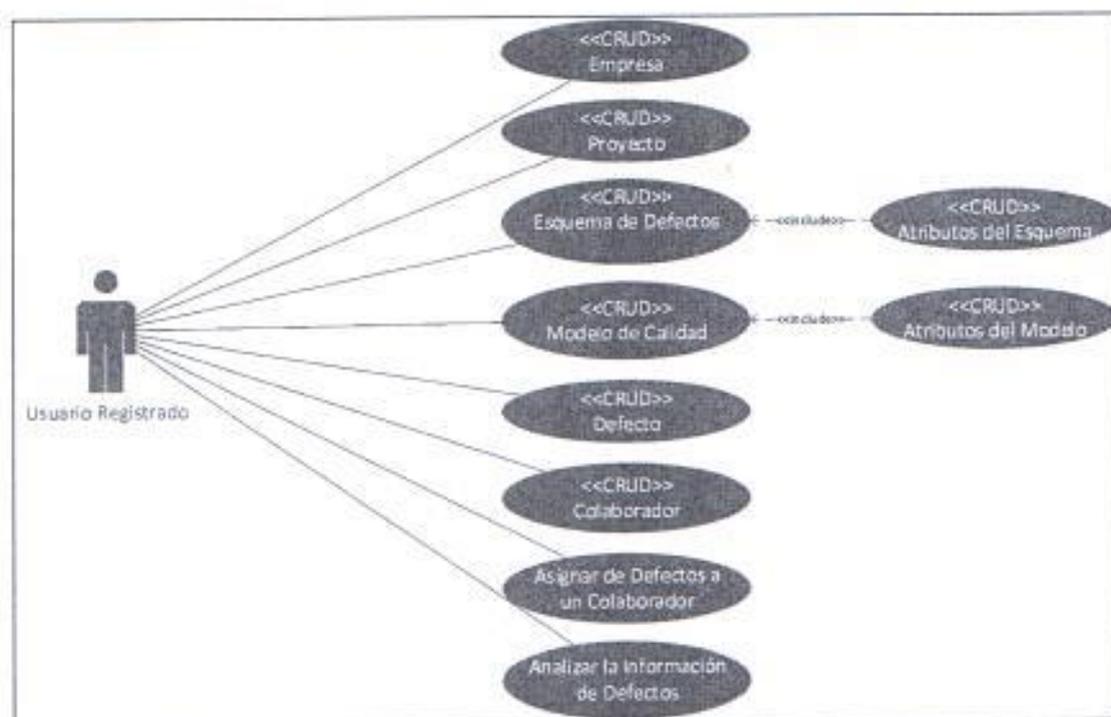


FIGURA 12 : Diagrama de Caso de Uso para Usuario Registrado

**C04: Crear una Empresa**

<b>Código:</b>	C04
<b>Nombre:</b>	Creación de una Empresa.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite el registro de una empresa al sistema.
<b>Prioridad:</b>	5
<b>Precondiciones:</b>	El usuario debe haberse autenticado ante el sistema.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. El usuario ingresa en la opción de creación de empresa.</li> <li>2. El sistema presenta el formulario de ingreso de una nueva empresa.</li> <li>3. El usuario llena los datos del formulario.</li> <li>4. El sistema verifica que la información ingresada por el usuario cumpla con los criterios de validación y presenta un mensaje para confirmar el ingreso.</li> </ol>

5. El usuario confirma el ingreso de los datos ingresados al sistema.
6. El sistema registra los datos ingresados por el usuario.

**Flujo Alternativo:**

4.A.- El sistema encuentra que los datos ingresados no cumplen con los criterios de validación por lo que le presenta al usuario un mensaje que pide el reingreso de los campos errados. Regresa al punto 3.

**Poscondiciones:**

El usuario registra una nueva empresa.

TABLA 9 : Caso de Uso Crear una Empresa

## C05: Listar las Empresas

<b>Código:</b>	C05
<b>Nombre:</b>	Listar las Empresas.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite listar las empresas creadas en el sistema.
<b>Prioridad:</b>	4
<b>Precondiciones:</b>	
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción de listar las empresas creadas.</li> <li>2. El sistema muestra a través de un listado las empresas registradas por el usuario en formato PDF.</li> </ol>
<b>Flujo Alternativo:</b>	
<b>Poscondiciones:</b>	El usuario puede visualizar las empresas creadas en el sistema.

TABLA 10 : Caso de Uso Listar las Empresas

## C06: Editar una Empresa

<b>Código:</b>	C06
<b>Nombre:</b>	Edición de una Empresa.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite la modificación de la información de una empresa.
<b>Prioridad:</b>	5
<b>Precondiciones:</b>	El usuario debe haber listado las empresas creadas.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la empresa que desea cambiar la información.</li> </ol>

2. El sistema presenta un formulario con la información de la empresa seleccionada.
3. El usuario cambia los datos de la empresa.
4. El sistema verifica que la información ingresada por el usuario cumpla con los criterios de validación y presenta un mensaje para confirmar la modificación.
5. El usuario confirma la modificación de los datos ingresados.
6. El sistema modifica los datos de la empresa seleccionada.

**Flujo Alternativo:**

4.A.- El sistema encuentra que los datos ingresados no cumplen con los criterios de validación por lo que le presenta al usuario un mensaje que pide el reintegro de los campos errados. Regresa al punto 3.

**Poscondiciones:**

El usuario actualiza la información de la empresa seleccionada.

TABLA 11 : Caso de Uso Editar una Empresa

## C07: Eliminar una Empresa

<b>Código:</b>	C07
<b>Nombre:</b>	Eliminar una Empresa.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite la eliminación de una empresa.
<b>Prioridad:</b>	5
<b>Precondiciones:</b>	El usuario debe haber listado las empresas creadas.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la empresa que desea eliminar y escoge la opción de eliminar.</li> <li>2. El sistema muestra la confirmación de la eliminación de la empresa.</li> <li>3. El usuario confirma la eliminación de la empresa seleccionada.</li> <li>4. El sistema verifica los criterios para permitir la eliminación.</li> <li>5. El sistema realiza la eliminación de la empresa seleccionada y re direcciona al usuario a la página de listado de empresas.</li> </ol>
<b>Flujo Alternativo:</b>	4.A.- El sistema encuentra que existe algún criterio por el cual no debe eliminarse el registro por lo cual presenta un mensaje al usuario con la información respectiva. Se regresa al paso 1.
<b>Poscondiciones:</b>	El usuario elimina una empresa registrada.

TABLA 12 : Caso de Uso Eliminar una Empresa

## C08: Imprimir Empresas

<b>Código:</b>	C08
<b>Nombre:</b>	Imprimir Empresas.

<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite la impresión del listado de empresas creadas por el usuario.
<b>Prioridad:</b>	4
<b>Precondiciones:</b>	El usuario debe haber listado las empresas creadas.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción imprimir las empresas creadas.</li> <li>2. El sistema muestra a través de un listado las empresas registradas por el usuario.</li> </ol>
<b>Flujo Alternativo:</b>	
<b>Poscondiciones:</b>	El usuario lista las empresas registradas por el usuario.

TABLA 13 : Caso de Uso Imprimir Empresas

## C09: Crear un Proyecto

<b>Código:</b>	C09
<b>Nombre:</b>	Creación de una Proyecto.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite el registro de un proyecto al sistema.
<b>Prioridad:</b>	5
<b>Precondiciones:</b>	El usuario debe haberse autenticado ante el sistema.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. El usuario ingresa en la opción de creación de proyecto.</li> <li>2. El sistema presenta el formulario de ingreso del nuevo proyecto.</li> <li>3. El usuario llena los datos del formulario.</li> <li>4. El sistema verifica que la información ingresada por el usuario cumpla con los criterios de validación y presenta un mensaje para confirmar el ingreso.</li> <li>5. El usuario confirma el ingreso de los datos ingresados al sistema.</li> <li>6. El sistema registra los datos ingresados por el usuario.</li> </ol>
<b>Flujo Alternativo:</b>	4.A.- El sistema encuentra que los datos ingresados no cumplen con los criterios de validación por lo que le presenta al usuario un mensaje que pide el reingreso de los campos errados. Regresa al punto 3.
<b>Poscondiciones:</b>	El usuario registra una nueva empresa.

TABLA 14 : Caso de Uso Crear un Proyecto

## C10: Listar los Proyectos

<b>Código:</b>	C10
<b>Nombre:</b>	Listar los Proyectos.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite listar los proyectos creados en el sistema.
<b>Prioridad:</b>	4
<b>Precondiciones:</b>	
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción de listar los proyectos creados.</li> <li>2. El sistema muestra a través de un listado los proyectos registradas por el usuario en formato PDF.</li> </ol>
<b>Flujo Alternativo:</b>	
<b>Poscondiciones:</b>	El usuario puede visualizar los proyectos creados en el sistema.

TABLA 15 : Caso de Uso Listar los Proyectos

## C11: Editar un Proyecto

<b>Código:</b>	C11
<b>Nombre:</b>	Edición de un Proyecto.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite la modificación de la información de un proyecto.
<b>Prioridad:</b>	5
<b>Precondiciones:</b>	El usuario debe haber listado los proyectos creados.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona el proyecto que desea cambiar la información.</li> <li>2. El sistema presenta un formulario con la información del proyecto seleccionado.</li> <li>3. El usuario cambia los datos del proyecto.</li> <li>4. El sistema verifica que la información ingresada por el usuario cumpla con los criterios de validación y presenta un mensaje para confirmar la modificación.</li> <li>5. El usuario confirma la modificación de los datos ingresados.</li> <li>6. El sistema modifica los datos de la empresa seleccionada.</li> </ol>
<b>Flujo Alternativo:</b>	<p>4.A.- El sistema encuentra que los datos ingresados no cumplen con los criterios de validación por lo que le presenta al usuario un mensaje que pide el reingreso de los campos errados. Regresa al punto 3.</p>

**Poscondiciones:**

El usuario actualiza la información de un proyecto.

TABLA 16 : Caso de Uso Editar un Proyecto

## C12: Eliminar un Proyecto

<b>Código:</b>	C12
<b>Nombre:</b>	Eliminar un Proyecto.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b> Usuario Registrado	
<b>Descripción:</b> Permite la eliminación de un proyecto.	
<b>Prioridad:</b> 5	
<b>Precondiciones:</b> El usuario debe haber listado los proyectos creados.	
<b>Flujo Normal:</b>	
6. El usuario selecciona el proyecto que desea eliminar y escoge la opción de eliminar.	
7. El sistema muestra la confirmación de la eliminación del proyecto.	
8. El usuario confirma la eliminación del proyecto seleccionado.	
9. El sistema verifica los criterios para permitir la eliminación.	
10. El sistema realiza la eliminación de la empresa seleccionada y re direcciona al usuario a la página de listado de proyectos.	
<b>Flujo Alternativo:</b>	
4.A.- El sistema encuentra que existe algún criterio por el cual no debe eliminarse el registro por lo cual presenta un mensaje al usuario con la información respectiva. Se regresa al paso 1.	
<b>Poscondiciones:</b> El usuario elimina un proyecto registrado.	

TABLA 17 : Caso de Uso Eliminar un Proyecto

## C13: Imprimir Proyectos

<b>Código:</b>	C13
<b>Nombre:</b>	Imprimir Proyectos.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b> Usuario Registrado	
<b>Descripción:</b> Permite la impresión del listado de proyectos creados por el usuario.	
<b>Prioridad:</b> 4	
<b>Precondiciones:</b> El usuario debe haber listado los proyectos creados.	
<b>Flujo Normal:</b>	
3. El usuario selecciona la opción imprimir los proyectos creados.	
4. El sistema muestra a través de un listado los proyectos registrados por	

el usuario.
<b>Flujo Alternativo:</b>
<b>Poscondiciones:</b> El usuario lista los proyectos registrados por el usuario.

TABLA 18 : Caso de Uso Imprimir Proyectos

## C14: Crear un Esquema de Defectos

<b>Código:</b>	C14
<b>Nombre:</b>	Creación de un Esquema de Defectos.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite el registro de un esquema de defectos.
<b>Prioridad:</b>	5
<b>Precondiciones:</b>	El usuario debe haberse autenticado ante el sistema.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>7. El usuario ingresa en la opción de creación de un esquema de defectos.</li> <li>8. El sistema presenta el formulario de ingreso del nuevo esquema.</li> <li>9. El usuario llena los datos del formulario.</li> <li>10. El usuario realiza la creación de uno o más atributos que forman el esquema.</li> <li>11. El sistema verifica que la información ingresada por el usuario cumpla con los criterios de validación y presenta un mensaje para confirmar el ingreso.</li> <li>12. El usuario confirma el ingreso de los datos ingresados al sistema.</li> <li>13. El sistema registra los datos ingresados por el usuario.</li> </ol>
<b>Flujo Alternativo:</b>	4.A.- El sistema encuentra que los datos ingresados no cumplen con los criterios de validación por lo que le presenta al usuario un mensaje que pide el reingreso de los campos errados. Regresa al punto 3.
<b>Poscondiciones:</b>	El usuario registra un nuevo esquema de defectos.

TABLA 19 : Caso de Uso Crear un Esquema de Defectos

## C15: Listar los Esquemas de Defectos

<b>Código:</b>	C15
<b>Nombre:</b>	Listar los Esquemas de Defectos.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	

Permite listar los esquemas creados en el sistema.
<b>Prioridad:</b> 4
<b>Precondiciones:</b>
<b>Flujo Normal:</b>
3. El usuario selecciona la opción de listar los esquemas de defectos creados.
4. El sistema muestra a través de un listado los esquemas registradas por el usuario en un formato PDF.
<b>Flujo Alternativo:</b>
<b>Poscondiciones:</b>
El usuario puede visualizar los esquemas de defectos creados en el sistema.

TABLA 20 : Caso de Uso Listar Esquemas de Defectos

## C16: Editar un Esquema de Defectos

<b>Código:</b>	C16
<b>Nombre:</b>	Edición de un Esquema de Defectos.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite la modificación de la información de un esquema de defectos.
<b>Precondiciones:</b>	El usuario debe haber listado los esquemas de defectos creados.
<b>Flujo Normal:</b>	
7.	El usuario selecciona el esquema de defectos del cual desea cambiar la información.
8.	El sistema presenta un formulario con la información del esquema de defectos seleccionado.
9.	El usuario cambia los datos del esquema.
10.	El sistema verifica que la información ingresada por el usuario cumpla con los criterios de validación y presenta un mensaje para confirmar la modificación.
11.	El usuario confirma la modificación de los datos ingresados.
12.	El sistema modifica los datos del esquema seleccionado.
<b>Flujo Alternativo:</b>	
4.A.-	El sistema encuentra que los datos ingresados no cumplen con los criterios de validación por lo que le presenta al usuario un mensaje que pide el reingreso de los campos errados. Regresa al punto 3.
<b>Poscondiciones:</b>	El usuario actualiza la información del esquema de defectos.

TABLA 21 : Caso de Uso Editar un Esquema de Defectos

## C17: Eliminar un Esquema de Defectos

<b>Código:</b>	C17
----------------	-----

<b>Nombre:</b>	Eliminar un Esquema de Defectos.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite la eliminación de un esquema de defectos.
<b>Precondiciones:</b>	El usuario debe haber listado los esquemas creados.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>11. El usuario selecciona el esquema que desea eliminar y escoge la opción de eliminar.</li> <li>12. El sistema muestra la confirmación de la eliminación del esquema.</li> <li>13. El usuario confirma la eliminación del esquema seleccionado.</li> <li>14. El sistema verifica los criterios para permitir la eliminación.</li> <li>15. El sistema realiza la eliminación del esquema seleccionado y re direcciona al usuario a la página de listado de esquemas.</li> </ol>
<b>Flujo Alternativo:</b>	4.A.- El sistema encuentra que existe algún criterio por el cual no debe eliminarse el registro por lo cual presenta un mensaje al usuario con la información respectiva. Se regresa al paso 1.
<b>Poscondiciones:</b>	El usuario elimina un esquema registrado.

TABLA 22 : Caso de Uso Eliminar un Esquema de Defectos

## C18: Imprimir Esquemas de Defectos

<b>Código:</b>	C18
<b>Nombre:</b>	Imprimir Esquemas de Defectos.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite la impresión del listado de esquemas creados por el usuario.
<b>Precondiciones:</b>	El usuario debe haber listado los esquemas creados.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>5. El usuario selecciona la opción imprimir los esquemas creados.</li> <li>6. El sistema muestra a través de un listado los esquemas registrados por el usuario.</li> </ol>
<b>Flujo Alternativo:</b>	
<b>Postcondiciones:</b>	El usuario lista los esquemas de defectos registrados por el usuario.

TABLA 23 : Caso de Uso Imprimir Esquemas de Defectos

## C19: Crear Atributos de Esquema

<b>Código:</b>	C19
<b>Nombre:</b>	Crear de Atributos de Esquema.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite el registro de atributos de un esquema de defectos.
<b>Prioridad:</b>	5
<b>Precondiciones:</b>	El usuario debe haber ingresado a crear o editar un esquema de defectos.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>14. El usuario ingresa en la opción de agregar atributos a un esquema de defectos.</li> <li>15. El sistema presenta el formulario de ingreso del nuevo atributo.</li> <li>16. El usuario llena los datos del formulario.</li> <li>17. El sistema verifica que la información ingresada por el usuario cumpla con los criterios de validación y presenta un mensaje para confirmar el ingreso.</li> <li>18. El usuario confirma el ingreso de los datos ingresados al sistema.</li> <li>19. El sistema agrega atributos al esquema seleccionado y agrega este atributo al listado de atributos del esquema.</li> </ol>
<b>Flujo Alternativo:</b>	4.A.- El sistema encuentra que los datos ingresados no cumplen con los criterios de validación por lo que le presenta al usuario un mensaje que pide el reingreso de los campos errados. Regresa al punto 3.
<b>Poscondiciones:</b>	El usuario registra un nuevo atributo a un esquema de defectos.

TABLA 24 : Caso de Uso Crear Atributos de Esquema

## C20: Editar un Atributo del Esquema

<b>Código:</b>	C20
<b>Nombre:</b>	Edición de un Atributo del Esquema.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite la modificación de la información de un atributo de esquema.
<b>Precondiciones:</b>	El usuario debe haber ingresado a crear o editar un esquema de defectos.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>13. El usuario selecciona el atributo del esquema del cual desea cambiar la información y presiona el botón actualizar.</li> </ol>

14. El sistema presenta un formulario con la información del atributo del esquema seleccionado.
15. El usuario cambia los datos del atributo.
16. El sistema verifica que la información ingresada por el usuario cumpla con los criterios de validación y presenta un mensaje para confirmar la modificación.
17. El usuario confirma la modificación de los datos ingresados.
18. El sistema modifica los datos del esquema seleccionado.

**Flujo Alternativo:**

4.A.- El sistema encuentra que los datos ingresados no cumplen con los criterios de validación por lo que le presenta al usuario un mensaje que pide el reintegro de los campos errados. Regresa al punto 3.

**Poscondiciones:**

El usuario actualiza la información del atributo de esquema.

TABLA 25 : Caso de Uso Editar un Atributo del Esquema

## C21: Eliminar un Atributo del Esquema

<b>Código:</b>	C21
<b>Nombre:</b>	Eliminar un Atributo del Esquema.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite la eliminación de un atributo del esquema.
<b>Precondiciones:</b>	El usuario debe haber listado los atributos creados.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>16. El usuario selecciona el atributo que desea eliminar y escoge la opción de eliminar.</li> <li>17. El sistema muestra la confirmación de la eliminación del atributo.</li> <li>18. El usuario confirma la eliminación del atributo seleccionado.</li> <li>19. El sistema verifica los criterios para permitir la eliminación y que no esté siendo usado en un proyecto existente.</li> <li>20. El sistema realiza la eliminación del atributo seleccionado y re direcciona al usuario a la página de listado de atributos.</li> </ol>
<b>Flujo Alternativo:</b>	<p>4.A.- El sistema encuentra que existe algún criterio por el cual no debe eliminarse el registro por lo cual presenta un mensaje al usuario con la información respectiva. Se regresa al paso 1.</p>
<b>Poscondiciones:</b>	El usuario elimina un atributo del esquema registrado.

TABLA 26 : Caso de Uso Eliminar un Atributo del Esquema

## C22: Crear un Modelo de Calidad

<b>Código:</b>	C22
<b>Nombre:</b>	Creación de un Modelo de Calidad.

<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite el registro de un modelo de calidad.
<b>Prioridad:</b>	5
<b>Precondiciones:</b>	El usuario debe haberse autenticado ante el sistema.
<b>Flujo Normal:</b>	<p>20. El usuario ingresa en la opción de creación de un modelo de calidad.</p> <p>21. El sistema presenta el formulario de ingreso del nuevo modelo de calidad.</p> <p>22. El usuario llena los datos del formulario.</p> <p>23. El usuario realiza la creación de uno o más atributos que forman el modelo.</p> <p>24. El sistema verifica que la información ingresada por el usuario cumpla con los criterios de validación y presenta un mensaje para confirmar el ingreso.</p> <p>25. El usuario confirma el ingreso de los datos ingresados al sistema.</p> <p>26. El sistema registra los datos ingresados por el usuario.</p>
<b>Flujo Alternativo:</b>	<p>4.A.- El sistema encuentra que los datos ingresados no cumplen con los criterios de validación por lo que le presenta al usuario un mensaje que pide el reingreso de los campos errados. Regresa al punto 3.</p>
<b>Poscondiciones:</b>	El usuario registra una nueva empresa.

TABLA 27 : Caso de Uso Crear un Modelo de Calidad

## C23: Listar los Modelos de Calidad

<b>Código:</b>	C23
<b>Nombre:</b>	Listar los Modelos de Calidad.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite listar los modelos de calidad creados en el sistema.
<b>Prioridad:</b>	4
<b>Precondiciones:</b>	
<b>Flujo Normal:</b>	<p>5. El usuario selecciona la opción de listar los modelos de calidad creados.</p> <p>6. El sistema muestra a través de un listado los modelos registrados por el usuario en un formato PDF.</p>
<b>Flujo Alternativo:</b>	

**Poscondiciones:**  
El usuario puede visualizar los modelos de calidad creados en el sistema.

TABLA 28 : Caso de Uso Listar los Modelos de Calidad

C24: Editar un Modelo de Calidad

<b>Código:</b>	C24
<b>Nombre:</b>	Edición de un Modelo de Calidad.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite la modificación de la información de un modelo de calidad.
<b>Precondiciones:</b>	El usuario debe haber listado los modelos de calidad creados.
<b>Flujo Normal:</b>	<p>19. El usuario selecciona el modelo de calidad del cual desea cambiar la información.</p> <p>20. El sistema presenta un formulario con la información del modelo de calidad seleccionado.</p> <p>21. El usuario cambia los datos del modelo.</p> <p>22. El sistema verifica que la información ingresada por el usuario cumpla con los criterios de validación y presenta un mensaje para confirmar la modificación.</p> <p>23. El usuario confirma la modificación de los datos ingresados.</p> <p>24. El sistema modifica los datos del modelo seleccionado.</p>
<b>Flujo Alternativo:</b>	<p>4.A.- El sistema encuentra que los datos ingresados no cumplen con los criterios de validación por lo que le presenta al usuario un mensaje que pide el reintegro de los campos errados. Regresa al punto 3.</p>
<b>Poscondiciones:</b>	El usuario actualiza la información del modelo de calidad.

TABLA 29 : Caso de Uso Editar un Modelo de Calidad

C25: Eliminar un Modelo de Calidad

<b>Código:</b>	C25
<b>Nombre:</b>	Eliminar un Modelo de Calidad.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite la eliminación de un modelo de calidad.
<b>Precondiciones:</b>	El usuario debe haber listado los modelos creados.
<b>Flujo Normal:</b>	<p>21. El usuario selecciona el modelo que desea eliminar y escoge la opción</p>

de eliminar.
22. El sistema muestra la confirmación de la eliminación del proyecto.
23. El usuario confirma la eliminación del proyecto seleccionado.
24. El sistema verifica los criterios para permitir la eliminación.
25. El sistema realiza la eliminación de la empresa seleccionada y re direcciona al usuario a la página de listado de modelos.
<b>Flujo Alternativo:</b> 4.A.- El sistema encuentra que existe algún criterio por el cual no debe eliminarse el registro por lo cual presenta un mensaje al usuario con la información respectiva. Se regresa al paso 1.
<b>Poscondiciones:</b> El usuario elimina un modelo registrado.

TABLA 30 : Caso de Uso Eliminar un Modelo de Calidad

## C26: Imprimir Modelos de Calidad

<b>Código:</b>	C26
<b>Nombre:</b>	Imprimir Modelos de Calidad.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite la impresión del listado de modelos creados por el usuario.
<b>Precondiciones:</b>	El usuario debe haber listado los modelos creados.
<b>Flujo Normal:</b>	7. El usuario selecciona la opción imprimir los modelos creados. 8. El sistema muestra a través de un listado los modelos registrados por el usuario en formato PDF.
<b>Flujo Alternativo:</b>	
<b>Postcondiciones:</b>	El usuario lista los proyectos registrados por el usuario.

TABLA 31 : Caso de Uso Imprimir Modelos de Calidad

## C27: Crear Atributo de Modelo

<b>Código:</b>	C27
<b>Nombre:</b>	Crear de Atributo de Modelo.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite el registro de atributos de un modelo de calidad.
<b>Prioridad:</b>	5
<b>Precondiciones:</b>	El usuario debe haber ingresado a crear o editar un modelo de calidad.

<p><b>Flujo Normal:</b></p> <p>27. El usuario ingresa en la opción de agregar atributos a un modelo de calidad.</p> <p>28. El sistema presenta el formulario de ingreso del nuevo atributo.</p> <p>29. El usuario llena los datos del formulario.</p> <p>30. El sistema verifica que la información ingresada por el usuario cumpla con los criterios de validación y presenta un mensaje para confirmar el ingreso.</p> <p>31. El usuario confirma el ingreso de los datos ingresados al sistema.</p> <p>32. El sistema agrega atributos al esquema seleccionado y agrega este atributo al listado de atributos de modelo.</p>
<p><b>Flujo Alternativo:</b></p> <p>4.A.- El sistema encuentra que los datos ingresados no cumplen con los criterios de validación por lo que le presenta al usuario un mensaje que pide el reingreso de los campos errados. Regresa al punto 3.</p>
<p><b>Poscondiciones:</b></p> <p>El usuario registra un nuevo atributo a un modelo de calidad.</p>

TABLA 32 : Caso de Uso Crear Atributo de Modelo

## C28: Editar un Atributo de Modelo

<b>Código:</b>	C28
<b>Nombre:</b>	Edición de un Atributo de Modelo.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite la modificación de la información de un atributo de modelo.
<b>Precondiciones:</b>	El usuario debe haber ingresado a crear o editar un modelo de calidad.
<b>Flujo Normal:</b>	<p>25. El usuario selecciona el atributo del modelo del cual desea cambiar la información y presiona el botón actualizar.</p> <p>26. El sistema presenta un formulario con la información del atributo del esquema seleccionado.</p> <p>27. El usuario cambia los datos del atributo.</p> <p>28. El sistema verifica que la información ingresada por el usuario cumpla con los criterios de validación y presenta un mensaje para confirmar la modificación.</p> <p>29. El usuario confirma la modificación de los datos ingresados.</p> <p>30. El sistema modifica los datos de un atributo del modelo de calidad.</p>
<b>Flujo Alternativo:</b>	<p>4.A.- El sistema encuentra que los datos ingresados no cumplen con los criterios de validación por lo que le presenta al usuario un mensaje que pide el reingreso de los campos errados. Regresa al punto 3.</p>

**Poscondiciones:**

El usuario actualiza la información del atributo de un modelo.

TABLA 33 : Caso de Uso Editar un Atributo de Modelo

C29: Eliminar un Atributo de Modelo

<b>Código:</b>	C29
<b>Nombre:</b>	Eliminar un Atributo de Modelo.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b> Usuario Registrado	
<b>Descripción:</b> Permite la eliminación de un atributo del modelo.	
<b>Precondiciones:</b> El usuario debe haber ingresado a crear o editar un modelo de calidad.	
<b>Flujo Normal:</b>	
26. El usuario selecciona el atributo que desea eliminar y escoge la opción de eliminar.	
27. El sistema muestra la confirmación de la eliminación del atributo.	
28. El usuario confirma la eliminación del atributo seleccionado.	
29. El sistema verifica los criterios para permitir la eliminación y que no esté siendo usado este atributo en un proyecto existente.	
30. El sistema realiza la eliminación del atributo seleccionado y re direcciona al usuario a la página de creación o modificación de un modelo.	
<b>Flujo Alternativo:</b>	
4.A.- El sistema encuentra que existe algún criterio por el cual no debe eliminarse el registro por lo cual presenta un mensaje al usuario con la información respectiva. Se regresa al paso 1.	
<b>Poscondiciones:</b> El usuario elimina un atributo del modelo de calidad registrado.	

TABLA 34 : Caso de Uso Eliminar un Atributo de Modelo

C30: Crear un Defecto

<b>Código:</b>	C30
<b>Nombre:</b>	Creación de un Defecto
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b> Usuario Registrado	
<b>Descripción:</b> Permite el registrar los defectos de un proyecto.	
<b>Prioridad:</b> 5	
<b>Precondiciones:</b> El usuario debe haber ingresado a listar los defectos de un proyecto.	
<b>Flujo Normal:</b>	
33. El usuario ingresa en la opción de creación de un defecto.	

34. El sistema presenta el formulario de ingreso del nuevo defecto.
35. El usuario llena los datos del formulario.
36. El sistema verifica que la información ingresada por el usuario cumpla con los criterios de validación y presenta un mensaje para confirmar el ingreso.
37. El usuario confirma el ingreso de los datos ingresados al sistema.
38. El sistema registra los datos ingresados por el usuario.

**Flujo Alternativo:**

4.A.- El sistema encuentra que los datos ingresados no cumplen con los criterios de validación por lo que le presenta al usuario un mensaje que pide el reingreso de los campos errados. Regresa al punto 3.

**Poscondiciones:**

El usuario registra un nuevo defecto a un proyecto.

TABLA 35 : Caso de Uso Crear un Defecto

## C31: Listar los Defectos

<b>Código:</b>	C31
<b>Nombre:</b>	Listar los Defectos.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite listar los defectos de un proyecto.
<b>Prioridad:</b>	4
<b>Precondiciones:</b>	El usuario debe haber ingresado en un proyecto e ingresar en la opción de registro de defectos.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>7. El usuario selecciona la opción de listar los defectos del proyecto.</li> <li>8. El sistema muestra a través de un listado los proyectos registradas por el usuario en formato PDF.</li> </ol>
<b>Flujo Alternativo:</b>	
<b>Poscondiciones:</b>	El usuario puede visualizar los defectos creados en el sistema.

TABLA 36 : Caso de Uso Listar los Defectos

## C32: Editar un Defecto

<b>Código:</b>	C32
<b>Nombre:</b>	Edición de un Defecto.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	

Permite la modificación de la información de un defecto registrado.
<b>Prioridad:</b> 5
<b>Precondiciones:</b> El usuario debe haber listado los defectos de un proyecto.
<b>Flujo Normal:</b> 31. El usuario selecciona el defecto que desea modificar. 32. El sistema presenta un formulario con la información del defecto seleccionado. 33. El usuario cambia los datos del proyecto. 34. El sistema verifica que la información ingresada por el usuario cumpla con los criterios de validación y presenta un mensaje para confirmar la modificación. 35. El usuario confirma la modificación de los datos ingresados. 36. El sistema modifica los datos del defecto seleccionado.
<b>Flujo Alternativo:</b> 4.A.- El sistema encuentra que los datos ingresados no cumplen con los criterios de validación por lo que le presenta al usuario un mensaje que pide el reingreso de los campos errados. Regresa al punto 3.
<b>Poscondiciones:</b> El usuario actualiza la información del defecto.

TABLA 37 : Caso de Uso Editar un Defecto

## C33: Eliminar un Defecto

<b>Código:</b>	C33
<b>Nombre:</b>	Eliminar un Defecto.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite la eliminación de un defecto.
<b>Prioridad:</b>	5
<b>Precondiciones:</b>	El usuario debe haber listado los defectos de un proyecto seleccionado.
<b>Flujo Normal:</b> 31. El usuario selecciona el defecto que desea eliminar y escoge la opción de eliminar. 32. El sistema muestra un mensaje con la confirmación de la eliminación del defecto. 33. El usuario confirma la eliminación del proyecto seleccionado. 34. El sistema verifica los criterios para permitir la eliminación. 35. El sistema realiza la eliminación de la empresa seleccionada y re direcciona al usuario a la página de listado de proyectos.	
<b>Flujo Alternativo:</b> 4.A.- El sistema encuentra que existe algún criterio por el cual no debe eliminarse el registro por lo cual presenta un mensaje al usuario con la	

información respectiva. Se regresa al paso 1.

**Poscondiciones:**

El usuario elimina un defecto registrado.

TABLA 38 : Caso de Uso Eliminar un Defecto

C34: Imprimir Defectos

<b>Código:</b>	C34
<b>Nombre:</b>	Imprimir Defectos.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite la impresión del listado de defectos creados por el usuario.
<b>Prioridad:</b>	4
<b>Precondiciones:</b>	El usuario debe haber listado los defectos creados.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>9. El usuario selecciona la opción imprimir los defectos creados.</li> <li>10. El sistema muestra a través de un listado los defectos registrados por el usuario.</li> </ol>
<b>Flujo Alternativo:</b>	
<b>Poscondiciones:</b>	El usuario lista los defectos registrados por el usuario.

TABLA 39 : Caso de Uso Imprimir Defectos

C35: Listar Colaboradores

<b>Código:</b>	C31
<b>Nombre:</b>	Listar los Colaboradores.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	9 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite listar los colaboradores registrados.
<b>Prioridad:</b>	4
<b>Precondiciones:</b>	El usuario debe haberse autenticado ante el sistema.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción de colaboradores.</li> <li>2. El sistema muestra a través de un listado los colaboradores de los diferentes proyectos existentes.</li> </ol>
<b>Flujo Alternativo:</b>	
<b>Poscondiciones:</b>	El usuario puede visualizar los defectos creados en el sistema.

TABLA 40 : Caso de Uso Listar Colaboradores

## C36: Solicitar un Colaborador

<b>Código:</b>	C36
<b>Nombre:</b>	Solicitar un Colaborador
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b> Usuario Registrado	
<b>Descripción:</b> La creación de una solicitud de colaboración a otro usuario del sistema.	
<b>Prioridad:</b> 5	
<b>Precondiciones:</b> El usuario debe haber ingresado a listar los colaboradores.	
<b>Flujo Normal:</b>	
<ol style="list-style-type: none"> <li>3. El usuario ingresa en la opción de solicitud de colaboración.</li> <li>4. El sistema presenta el formulario de creación de una solicitud de colaboración.</li> <li>5. El usuario llena los datos del formulario.</li> <li>6. El sistema verifica que la información ingresada por el usuario cumpla con los criterios de validación y presenta un mensaje para confirmar el ingreso.</li> <li>7. El usuario confirma el envío de la solicitud de colaboración.</li> <li>8. El sistema registra la solicitud de colaboración y envía un correo electrónico al usuario receptor.</li> </ol>	
<b>Flujo Alternativo:</b>	
4.A.- El sistema encuentra que los datos ingresados no cumplen con los criterios de validación por lo que le presenta al usuario un mensaje que pide el reintegro de los campos errados. Regresa al punto 3.	
<b>Poscondiciones:</b> El usuario registra un nuevo defecto a un proyecto.	

TABLA 41 : Caso de Uso Solicitar un Colaborador

## C37: Eliminar un Colaborador

<b>Código:</b>	C37
<b>Nombre:</b>	Eliminar un Colaborador.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b> Usuario Registrado	
<b>Descripción:</b> Permite la eliminación de un colaborador.	
<b>Prioridad:</b> 5	
<b>Precondiciones:</b> El usuario debe haber ingresado a listar los colaboradores.	
<b>Flujo Normal:</b>	
<ol style="list-style-type: none"> <li>1. El usuario selecciona el colaborador que desea eliminar y escoge la opción de eliminar.</li> </ol>	

2. El sistema muestra un mensaje con la confirmación de la eliminación del colaborador.
3. El usuario confirma la eliminación del colaborador seleccionado.
4. El sistema verifica los criterios para permitir la eliminación.
5. El sistema realiza la eliminación del colaborador seleccionado y re direcciona al usuario a la página de listado de colaboradores.

**Flujo Alternativo:**

4.A.- El sistema encuentra que existe algún criterio por el cual no debe eliminarse el registro por lo cual presenta un mensaje al usuario con la información respectiva. Se regresa al paso 1.

**Poscondiciones:**

El usuario elimina un colaborador registrado.

TABLA 42 : Caso de Uso Eliminar un Colaborador

C37: Asignar de Defecto a Colaborador

<b>Código:</b>	C37
<b>Nombre:</b>	Asignar defecto a Colaborador.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite que el usuario administrador del proyecto asigne un defecto a un colaborador.
<b>Prioridad:</b>	5
<b>Precondiciones:</b>	El usuario debe haber ingresado a lista los defectos de un proyecto.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona el defecto que desea asignar y escoge la opción de asignar defecto a colaborador.</li> <li>2. El sistema muestra un formulario con el listado de los usuarios que colaboran con el proyecto.</li> <li>3. El usuario selecciona el o los usuarios a los cuales asignará el defecto, y selecciona la opción asignar.</li> <li>4. El sistema muestra un mensaje con la confirmación de la asignación del defecto.</li> <li>5. El usuario acepta la confirmación de la asignación colaborador seleccionado.</li> <li>6. El sistema verifica los criterios para permitir la asignación.</li> <li>7. El sistema realiza la asignación de defectos a los colaboradores seleccionados.</li> </ol>
<b>Flujo Alternativo:</b>	4.A.- El sistema encuentra que existe algún criterio por el cual no debe asignar el defecto a un usuario por lo que presenta un mensaje al usuario con la información respectiva. Se regresa al paso 2.
<b>Poscondiciones:</b>	El usuario asigna un defecto a él o los colaboradores.

TABLA 43 : Caso de Uso Asignar de Defecto a Colaborador

C37: Analizar la información de Defectos

<b>Código:</b>	C37
<b>Nombre:</b>	Analizar la información de Defectos
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b> Usuario Registrado	
<b>Descripción:</b> Permite que el usuario administrador del proyecto realice análisis sobre la información de defectos registrada para un proyecto.	
<b>Prioridad:</b> 5	
<b>Precondiciones:</b> El usuario debe haber ingresado al listado de proyectos registrados.	
<b>Flujo Normal:</b>	
<ol style="list-style-type: none"> <li>1. El usuario selecciona el proyecto en el cual quisiera realizar algún análisis de su información de defectos y selecciona la opción analizar información de defectos.</li> <li>2. El sistema muestra un formulario con los diferentes análisis que se pueden aplicar a la información de defectos del proyecto seleccionado.</li> <li>3. El usuario selecciona el tipo de análisis que desea realizar y presiona la opción generar.</li> <li>4. El sistema realiza el análisis de defectos al proyecto seleccionado.</li> <li>5. El sistema presenta un reporte con los resultados obtenidos por este análisis realizado.</li> </ol>	
<b>Flujo Alternativo:</b>	
<b>Poscondiciones:</b> El usuario realiza un análisis de defectos a los datos registrados para un proyecto.	

TABLA 44 : Caso de Uso Analizar la información de Defectos

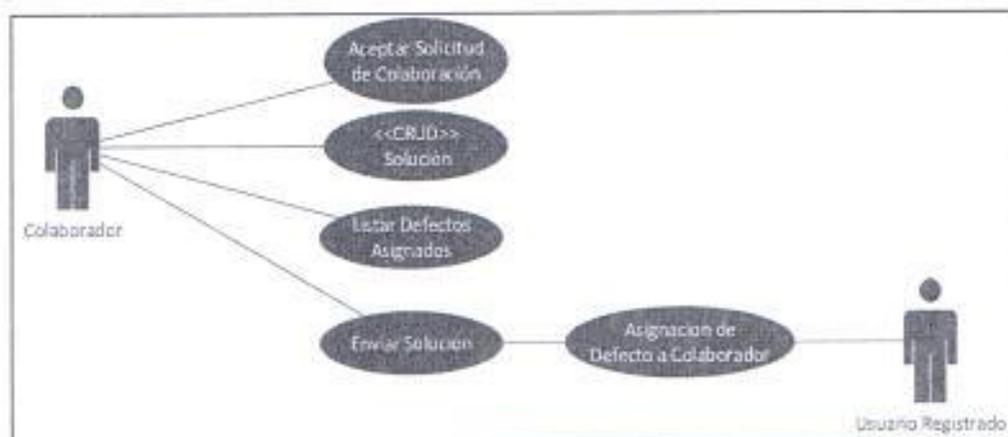


FIGURA 13 : Diagrama de Casos de Uso de un Colaborador

## C38: Aceptar Solicitud de Colaboración

<b>Código:</b>	C38
<b>Nombre:</b>	Aceptar Solicitud de Colaboración
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b> Usuario Registrado	
<b>Descripción:</b> Permite la aceptación de una solicitud de colaboración.	
<b>Prioridad:</b> 5	
<b>Precondiciones:</b> El usuario debe haber ingresado a listar los colaboradores.	
<b>Flujo Normal:</b>	
<ol style="list-style-type: none"> <li>1. El usuario selecciona la solicitud de colaboración recibida y presiona aceptar solicitud de colaboración.</li> <li>2. El sistema realiza el registro de aceptación de la solicitud recibida, y cambia de estado a la solicitud a aprobada.</li> </ol>	
<b>Flujo Alternativo:</b>	
<ol style="list-style-type: none"> <li>1.A.- El usuario selecciona la solicitud de colaboración pero presiona negar solicitud de colaboración.</li> <li>1.B.- El sistema realiza el registro de negación de la solicitud recibida, y elimina del listado la solicitud seleccionada.</li> </ol>	
<b>Poscondiciones:</b> El usuario acepta una solicitud de colaboración.	

TABLA 45 : Caso de Uso Aceptar Solicitud de Colaboración

## C39: Listar Defectos Asignados

<b>Código:</b>	C39
<b>Nombre:</b>	Listar Defectos Asignados.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	9 de noviembre del 2013
<b>Actores:</b> Usuario Registrado	
<b>Descripción:</b> Permite listar los defectos asignados a un colaborador.	
<b>Prioridad:</b> 4	
<b>Precondiciones:</b> El usuario debe haber ingresado al listado de proyectos asignados al colaborador.	
<b>Flujo Normal:</b>	
<ol style="list-style-type: none"> <li>1. El usuario selecciona uno de los proyectos asignados y selecciona la opción mostrar defectos asignados.</li> <li>2. El sistema muestra un listado de los defectos asignados a un usuario.</li> </ol>	
<b>Flujo Alternativo:</b>	
<b>Poscondiciones:</b> El usuario lista los defectos asignados en su rol de colaborador.	

TABLA 46 : Caso de Uso Listar Defectos Asignados

## C40: Crear una Solución

<b>Código:</b>	C40
<b>Nombre:</b>	Creación de una Solución
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite el registrar los de una solución para un defecto asignado.
<b>Prioridad:</b>	5
<b>Precondiciones:</b>	El usuario debe haber ingresado a listar los defectos de un proyecto en el que es colaborador.
<b>Flujo Normal:</b>	<p>39. El usuario ingresa en la opción de creación de una solución.</p> <p>40. El sistema presenta el formulario de ingreso del nueva solución.</p> <p>41. El usuario llena los datos del formulario.</p> <p>42. El sistema verifica que la información ingresada por el usuario cumpla con los criterios de validación y presenta un mensaje para confirmar el ingreso.</p> <p>43. El usuario confirma el ingreso de los datos ingresados al sistema.</p> <p>44. El sistema registra los datos ingresados por el usuario.</p>
<b>Flujo Alternativo:</b>	4.A.- El sistema encuentra que los datos ingresados no cumplen con los criterios de validación por lo que le presenta al usuario un mensaje que pide el reingreso de los campos errados. Regresa al punto 3.
<b>Poscondiciones:</b>	El usuario registra una nueva solución.

TABLA 47 : Caso de Uso Crear una Solución

## C41: Listar los Soluciones

<b>Código:</b>	C41
<b>Nombre:</b>	Listar los Soluciones.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite listar las soluciones de un proyecto.
<b>Prioridad:</b>	4
<b>Precondiciones:</b>	El usuario debe haber ingresado en un proyecto asignado.
<b>Flujo Normal:</b>	9. El usuario selecciona la opción de listar los soluciones.

10. El sistema muestra a través de un listado las soluciones registradas por el usuario en formato PDF.

**Flujo Alternativo:**

**Poscondiciones:**

El usuario puede visualizar las soluciones creados para un proyecto.

TABLA 48 : Caso de Uso Listar los Soluciones

C42: Editar una Solución

<b>Código:</b>	C42
<b>Nombre:</b>	Edición de una Solución
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite la modificación de la información de un defecto registrado.
<b>Prioridad:</b>	5
<b>Precondiciones:</b>	El usuario debe haber listado las soluciones de un proyecto asignado.
<b>Flujo Normal:</b>	<p>37. El usuario selecciona la solución que desea modificar.</p> <p>38. El sistema presenta un formulario con la información de la solución seleccionada.</p> <p>39. El usuario cambia los datos de la solución.</p> <p>40. El sistema verifica que la información ingresada por el usuario cumpla con los criterios de validación y presenta un mensaje para confirmar la modificación.</p> <p>41. El usuario confirma la modificación de los datos ingresados.</p> <p>42. El sistema modifica los datos de la solución seleccionada.</p>
<b>Flujo Alternativo:</b>	4.A.- El sistema encuentra que los datos ingresados no cumplen con los criterios de validación por lo que le presenta al usuario un mensaje que pide el reingreso de los campos errados. Regresa al punto 3.
<b>Poscondiciones:</b>	El usuario actualiza la información de la solución.

TABLA 49 : Caso de Uso Editar Solución

C43: Eliminar una Solución

<b>Código:</b>	C43
<b>Nombre:</b>	Eliminar una Solución.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite la eliminación de una solución registrada.
<b>Prioridad:</b>	5

<b>Precondiciones:</b> El usuario debe haber listado las soluciones de un proyecto asignado.
<b>Flujo Normal:</b> 36. El usuario selecciona la solución que desea eliminar y escoge la opción de eliminar. 37. El sistema muestra un mensaje con la confirmación de la eliminación de la solución. 38. El usuario confirma la eliminación del proyecto seleccionado. 39. El sistema verifica los criterios para permitir la eliminación. 40. El sistema realiza la eliminación de la empresa seleccionada y re direcciona al usuario a la página de listado de soluciones.
<b>Flujo Alternativo:</b> 4.A.- El sistema encuentra que existe algún criterio por el cual no debe eliminarse el registro por lo cual presenta un mensaje al usuario con la información respectiva. Se regresa al paso 1.
<b>Poscondiciones:</b> El usuario elimina una solución registrada.

TABLA 50 : Caso de Uso Eliminar una Solución

## C44: Imprimir Soluciones

<b>Código:</b>	C44
<b>Nombre:</b>	Imprimir Soluciones.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b> Usuario Registrado	
<b>Descripción:</b> Permite la impresión del listado de soluciones creados por el usuario.	
<b>Prioridad:</b> 4	
<b>Precondiciones:</b> El usuario debe haber listado las soluciones creadas para un proyecto asignado.	
<b>Flujo Normal:</b> 11. El usuario selecciona la opción imprimir los defectos creados. 12. El sistema muestra a través de un listado las soluciones creadas para un proyecto asignado.	
<b>Flujo Alternativo:</b>	
<b>Poscondiciones:</b> El usuario lista las soluciones registradas por el usuario.	

TABLA 51 : Caso de Uso Imprimir Soluciones

## C45: Enviar una Solución

<b>Código:</b>	C45
<b>Nombre:</b>	Enviar Solución.
<b>Autor:</b>	Daniel Silva
<b>Fecha:</b>	8 de noviembre del 2013
<b>Actores:</b>	Usuario Registrado
<b>Descripción:</b>	Permite el envío de la o las soluciones creadas para a un defecto.
<b>Prioridad:</b>	4
<b>Precondiciones:</b>	El usuario debe haber listado las soluciones creadas para un proyecto asignado.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>3. El usuario selecciona la o las soluciones que desea enviar y selecciona la opción enviar.</li> <li>4. El sistema muestra un mensaje de confirmación del envío de las soluciones seleccionadas.</li> <li>5. El usuario confirma el envío de la o las soluciones.</li> <li>6. El sistema verifica que las soluciones cumplen con los criterios de envío.</li> <li>7. El sistema realiza el registro de las soluciones y envía un correo electrónico al administrador del proyecto.</li> </ol>
<b>Flujo Alternativo:</b>	<p>4.A.- El sistema encuentra que la o las soluciones seleccionadas no cumplen con los criterios para el envío.</p> <p>5B.- El sistema presenta un mensaje que informa el problema al usuario y re direcciona nuevamente al usuario a la página de listado de soluciones.</p>
<b>Poscondiciones:</b>	El usuario realiza el envío de una solución a un defecto asignado.

TABLA 52 : Caso de Uso Enviar una Solución

## 4.4 Diseño de Interfaces

A continuación se presenta los diseños utilizados para el Sistema de Gestión de Defectos. Muchos de estos diagramas son las pantallas que aparecen en el sistema.

### 4.4.1 Interfaz de Ingreso al Sistema

La interfaz de la figura mostrada a continuación muestra el formulario de ingresar a la aplicación. Aquí se presenta las opciones de ingreso al sistema y de registro de un nuevo usuario.



The image shows a web form titled "Login". It has two input fields: "Usuario" containing the text "danyche" and "Password" containing masked characters "\*\*\*\*\*". Below the password field is a checkbox labeled "Remember me". At the bottom of the form are two buttons: "Login" and "Reset". At the very bottom of the page, there is a link that says "No estás Registrado? Regístrate Aquí".

FIGURA 14 : Diseño de Página de Inicio de Sesión.

### 4.4.2 Interfaz de Página Principal de la Aplicación

La interfaz de la figura representa la página inicial de la aplicación, aquí se puede distinguir secciones bien definidas como son la sección del panel de control el cual contiene todos los gráficos estadísticos de estado de la cuenta del usuario (Sección A). El menú que permite ingresar a las demás opciones de configuración

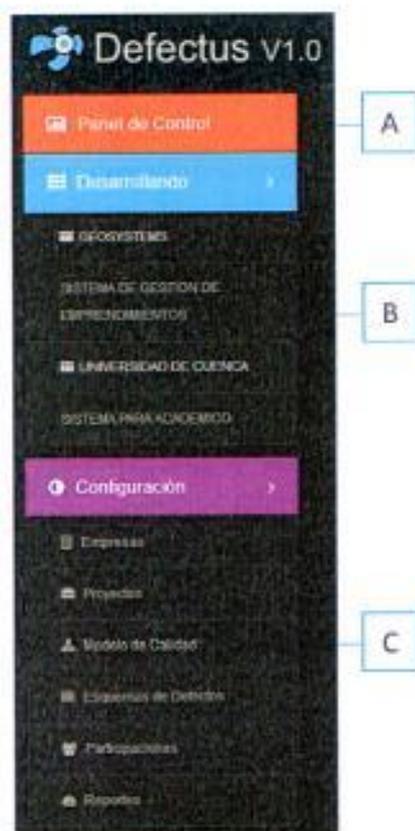
del sistema (Sección B) la barra de estado que permite ver el usuario logueado y la opción de salida del sistema (Sección C).



FIGURA 15 : Diseño de Panel de Control.

#### 4.4.3 Interfaz del Menú de Opciones de la Aplicación

Este menú muestra las opciones de acceso al sistema. Entre las opciones que presenta este menú se puede observar el acceso al panel de control (Opción A), el acceso a las opciones de desarrollo (Opción B) que muestra los proyectos en los que se está trabajando y las opciones de configuración (Opción C) donde es posible la creación de empresas, proyectos, esquemas, modelos, medidas de prevención, etc.



Interfaz 3

FIGURA 16 : Diseño de Menú de Opciones del Sistema.

#### 4.4.4 Interfaz Para Listados del Sistema

La interfaz de la figura muestra las opciones típicas de un mantenimiento del sistema. Este diseño es aplicable para el listado de empresas, proyectos, esquemas de calidad, modelos y medidas de prevención. Los mantenimientos muestran el listado de registros y permiten la creación, modificación, borrado, además presentan opciones de búsqueda e impresión.

# Lista de Empresas

Empresas de Software

Home / Empresas

<span>Nuevo</span> <span>Editar</span> <span>Borrar</span> <span>Imprimir</span> <input type="text" value="Buscar"/> <span>Buscar</span>			
RUC	Nombre	Ciudad	País
<input type="checkbox"/> 0105029645	GEOSYSTEMS	CUENCA	ECUADOR
<input type="checkbox"/> 040001564001	UNIVERSIDAD DE CUENCA	CUENCA	ECUADOR

FIGURA 17 : Diseño de Listado de un Mantenimiento del Sistema.

## 4.4.5 Interfaz de Creación y Edición de Registros

La interfaz de las figuras número 18, 19, 20 y 21 muestra los datos necesarios para la creación de empresas, proyectos, modelos de calidad y esquemas de defectos. Los campos que conforman un registro varían para cada opción del sistema. A continuación se muestran los diferentes tipos de formularios de creación que presenta el sistema.

### Editar Empresa

Mantenimiento

Home / Empresas / Editar

Datos de la Empresa	
Nombre	UNIVERSIDAD DE CUENCA
Nombre Corto	UCUENCA
Razón Social	UNIVERSIDAD DE CUENCA
RUC	040001564001
Teléfono	2881952
Página Web	WWW.UCUENCA.EDU.EC
Email	info@ucuenca.edu.ec
Dirección	12 DE ABRIL
Ciudad	CUENCA
País	ECUADOR
<input type="button" value="Guardar"/> <input type="button" value="Cancelar"/>	

FIGURA 18 : Diseño de Formulario de Edición de una Empresa.

## Editar Proyecto

SISTEMA PARA ACADÉMICO

Inicio / Proyecto / Editar

**Datos del Proyecto**

Nombre: SISTEMA PARA ACADÉMICO

Nombre corto: SPA

Descripción: SISTEMA LLEVA EL CONTROL Y REGISTRO DE LOS ESTUDIANTES DE UN SISTEMA.

Duración: 2  
AÑO

Tipo Proyecto: ACADÉMICO

Institución: UNIVERSIDAD DE CUENCA

Equipos: ESQUEMA DE CLASIFICACION DE DEFECTOS PARA ESQUEMAS CONCEPTUALES

**Datos del Técnico**

Lenguaje: JAVA

Herramienta: NETBEANS

Metodología: AXL

Fase de Desarrollo: ANALISIS Y DISEÑO

Tecnología: SF, RICHFACE, HIBERNATE, SUBVERSION

**Participantes**

Participantes:

Nombre	Cargo	
DANIEL ANDRÉS SILVA PALACIOS	ADMINISTRADOR	
SEBASTIÁN SILVA	EDITOR	
DELIA GABRIELA GRANDA JUCA	PROGRAMADORA	

Cancelar Cambiar

FIGURA 19 : Diseño de Formulario de Edición de un Proyecto.

## Esquema

Something Goes Here

Home | Escema

▲ ▼ ✕
**Datos de la Esquema**

**Nombre:** ESQUEMA DE CLASIFICACION DE DEFECTOS PARA ESQUEMAS CONCEI

**Descripción:** ESTA CLASIFICACION DE DEFECTOS REPRESENTA DEFINA LOS ATRIBUTOS

**Modelo de Calidad:** NORMA DE EVALUACION ISO9001

▲ ▼ ✕
**Atributos del Esquema**

**Agrega Atributo**

Nombre	Opciones										
<div style="background-color: #ccc; padding: 2px; margin-bottom: 5px;">Defecto ID</div>	<p><b>Descripción:</b> Identificador único que indica la calidad de la propiedad (1 - 6) más el número secuencial de el defecto con la propiedades de calidad. (El formato es: D6CXY)</p> <p><b>Componente:</b> <input type="text" value="Selecciona varias opciones con valores, nombre y texto"/></p> <p><b>Orden:</b> 1</p> <p><b>Agregar Opción</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Nombre</th> <th style="width: 60%;">Descripción</th> <th style="width: 30%;"></th> </tr> </thead> <tbody> <tr> <td>X</td> <td>Propiedad de Calidad (1-6)</td> <td style="text-align: right;">▲ ▼ ✕</td> </tr> <tr> <td>Y</td> <td>Número secuencial de defectos que identifica los defectos con la clasificación de defectos basados en calidad.</td> <td style="text-align: right;">▲ ▼ ✕</td> </tr> </tbody> </table>	Nombre	Descripción		X	Propiedad de Calidad (1-6)	▲ ▼ ✕	Y	Número secuencial de defectos que identifica los defectos con la clasificación de defectos basados en calidad.	▲ ▼ ✕	
Nombre	Descripción										
X	Propiedad de Calidad (1-6)	▲ ▼ ✕									
Y	Número secuencial de defectos que identifica los defectos con la clasificación de defectos basados en calidad.	▲ ▼ ✕									
<div style="background-color: #ccc; padding: 2px; margin-bottom: 5px;">Categoría</div>	<p><b>Descripción:</b> Que tipo de defecto basados en los síntomas. Grupos de defectos basados en las características comunes.</p> <p><b>Componente:</b> <input type="text" value="Texto"/></p> <p><b>Orden:</b> 15</p> <p><b>Agregar Opción</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Nombre</th> <th style="width: 60%;">Descripción</th> <th style="width: 30%;"></th> </tr> </thead> <tbody> <tr> <td>Categoría</td> <td>Nombre de la Categoría</td> <td style="text-align: right;">▲ ▼ ✕</td> </tr> </tbody> </table>	Nombre	Descripción		Categoría	Nombre de la Categoría	▲ ▼ ✕				
Nombre	Descripción										
Categoría	Nombre de la Categoría	▲ ▼ ✕									

1
2
▶
◀
100%

Guardar
Cancelar

FIGURA 20 : Diseño de Formulario de Edición de un Esquema.

## Editar Modelo de Calidad

CONTABILIDAD / CALIDAD

Inicio / Configuración / Editar

**Datos del Modelo de Calidad**

Nombre:

Descripción:

**Atributos del Modelo de Calidad**

Nombre	Descripción	Relaciona	Subcategoría
<input type="checkbox"/> Satisfacción	La satisfacción es la respuesta del usuario a la interacción con el software, e incluye las actitudes hacia el uso del mismo.	+	CALIDAD DE UI
<input type="checkbox"/> Cambiabilidad	La capacidad del software para que la implementación de una modificación se pueda realizar. Incluye también codificación.	+	CAPACIDAD DE

FIGURA 21 : Diseño de Formulario de Modelo de Calidad.

### 4.4.6 Interfaz de Reportes

Esta sección muestra las opciones del formulario de generación de los reportes del sistema. Todas las opciones del formulario deben ser seleccionadas para la generación de un reporte.

## Reportes del Sistema

ANÁLISIS DE CALIDAD

Inicio

**Filtros**

Empresa:

Proyecto:

Reporte de Análisis:

FIGURA 22 : Diseño de Formulario de Generación de Reportes.

## **4.5 Diseño del Esquema de Seguridad de la Aplicación**

### **4.5.1 Autenticación**

La autenticación del sistema se realiza a través de un formulario de autenticación, donde se ingresan las credenciales del usuario. Se solicita un usuario y la contraseña que fue seleccionada por el usuario al registrarse al sistema, el usuario puede ser el correo electrónico o el nombre de usuario. La clave de seguridad es la seleccionada al registrarse al sistema. El sistema posibilita el cambiar esta clave a través de la opción datos de perfil del usuario.

### **4.5.2 Autorización**

La autorización a los recursos de la aplicación se realiza a través de un filtro de seguridad que verifica la existencia de una sesión de usuario. En el caso que se haya creado una sesión a través del formulario de autenticación este filtro permite el acceso normal a los recursos de la aplicación, en el caso que no exista una sesión válida se restringe el acceso a los recursos y el usuario es redireccionado a la página de ingreso del sistema.

Las opciones de la aplicación y las funciones exclusivas asignadas a los usuarios son determinadas a través de una verificación de los permisos del usuario que son almacenados en la base de datos y son verificados cada vez que el usuario ingresa al sistema.

### 4.5.3 Seguridad en la Conexión

Se logra obtener una conexión segura a la aplicación a través de la implementación de certificados digitales en el servidor Glassfish. Se ha generado un certificado auto firmado con fines pruebas para asegurar la comunicación entre la maquina cliente y el servidor de aplicaciones.

## 4.6 Diseño de Reportes y Paneles de Control

### 4.6.1 Diseño de Panel de Control Principal

El panel de control muestra en un formato gráfico los datos y estadísticas de la cuenta de usuario. En esta sección se presentan los siguientes gráficas:

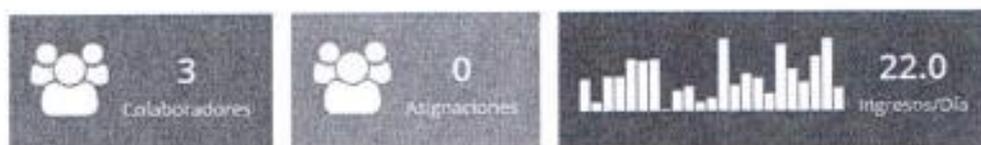


FIGURA 23 : Gráficas Estadísticas de la Cuenta de Usuario.

Estas gráficas muestra el número de colaboradores que actualmente trabajan en los proyectos que dirige el usuario, en el caso que el usuario trabaje para algún proyecto también presenta el número total de defectos asignados al usuario. Además muestra a través de un gráfico de barras los datos históricos de los ingresos diarios al sistema y su promedio.

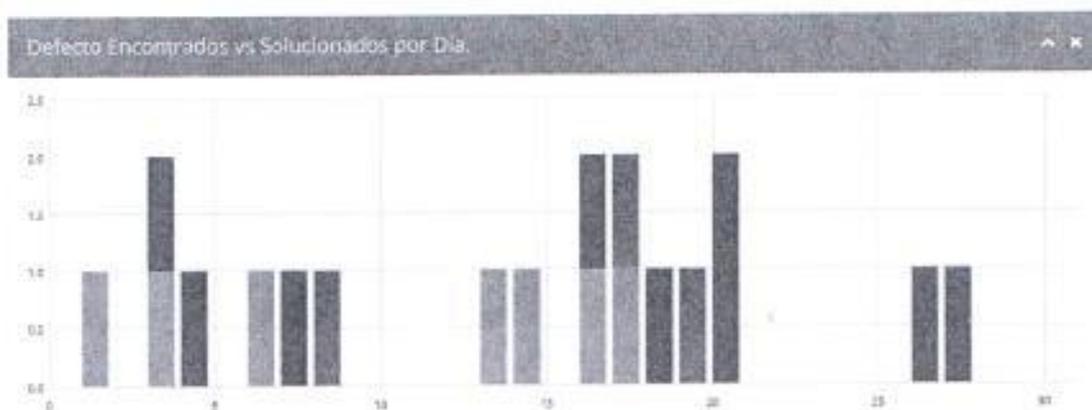


FIGURA 24 : Gráfica De Defectos Encontrados vs Solucionados por Día.

Este gráfico muestra los defectos encontrados por cada día del mes actual, además nos muestra con un tono más claro la cantidad de defecto fueron solucionados exitosamente.



FIGURA 25 : Gráfica de Proyectos y Datos de la Cuenta.

El gráfico de proyecto muestra de forma gráfica el porcentaje de defectos que han sido solucionados de cada proyecto, esta información permite llevar un control y dar seguimiento al porcentaje de defectos solucionados.



FIGURA 26 : Gráfica de Defectos Encontrados Versus Solucionados.

El gráfico acumulado de defectos encontrados vs solucionados por día, muestra un histórico acumulado de la cantidad de defectos que han sido encontrados y solucionados en los proyectos existentes. Con este gráfico el usuario puede hacer una evaluación de rendimiento y eficiencia que tiene la empresa al resolver defectos de software.

#### 4.6.2 Diseño de Reporte para Proyectos Individuales

Este reporte permite analizar datos relacionados a proyectos específicos. Ya que este proyecto de tesis centra su análisis en la implementación de un esquema de clasificación de defectos para en el diseño y modelado de esquemas conceptuales estos reportes son exclusivos con el uso de este esquema de clasificación. El diseño que se presenta a continuación es aplicable para varios análisis presentados en el sistema. A continuación se presenta un reporte que muestra la información de los defectos categorizado por los atributos de calidad.



FIGURA 27 : Análisis por Atributos de Calidad.

En esta gráfica se puede apreciar la ocurrencia acumulada de los defectos agrupados por los atributos de calidad. Se puede observar la ocurrencia histórica de los defectos, la cantidad total de defectos encontrados y el porcentaje que cada una de las categorías representadas. Estos tipos de reporte permiten realizar un análisis profundo de los defectos organizados por los atributos del esquema de calidad.

## 4.7 Mecanismos de Gestión

### 4.7.1 Análisis de Causalidad

El análisis causal permite entender las relaciones de causalidad que existen entre defectos de software encontrados en un proyecto. Estas relaciones pueden explicar las repercusiones que tiene un defecto en el funcionamiento de un

sistema. En la figura que se muestra a continuación se observa las relaciones existentes en una red de defectos causal típica de un proyecto informático.



FIGURA 28 : Diagrama de Causalidad Típica de un Proyecto Informático

Un defecto puede ser el causante de uno o varios defectos y estos a su vez podrían ser el origen de muchos más. La estructura resultante es un árbol con relaciones causales. La figura a continuación muestra las posibles relaciones entre defectos de software que se pueden encontrar en un diagrama causal de defectos.

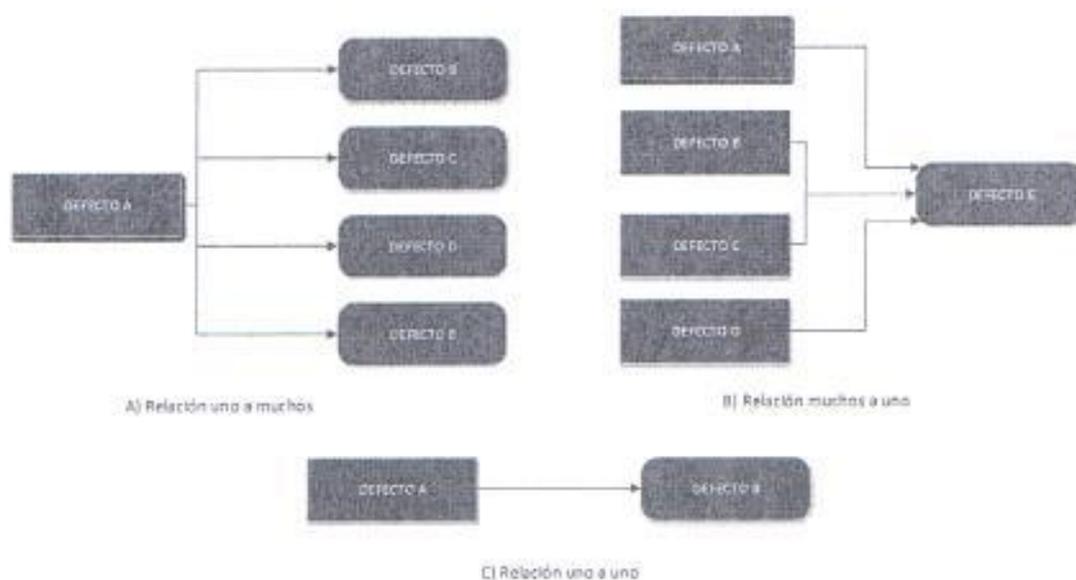


FIGURA 29 : Tipos de Relaciones entre Defectos.

Con un análisis eficiente de causalidad se puede obtener el orden óptimo de solución de defectos de software, esta generación se puede realizar a través de la opción "Generar Orden Óptimo" del sistema de defectos. A continuación se muestra un ejemplo típico de una red de defectos encontrada para un proyecto.

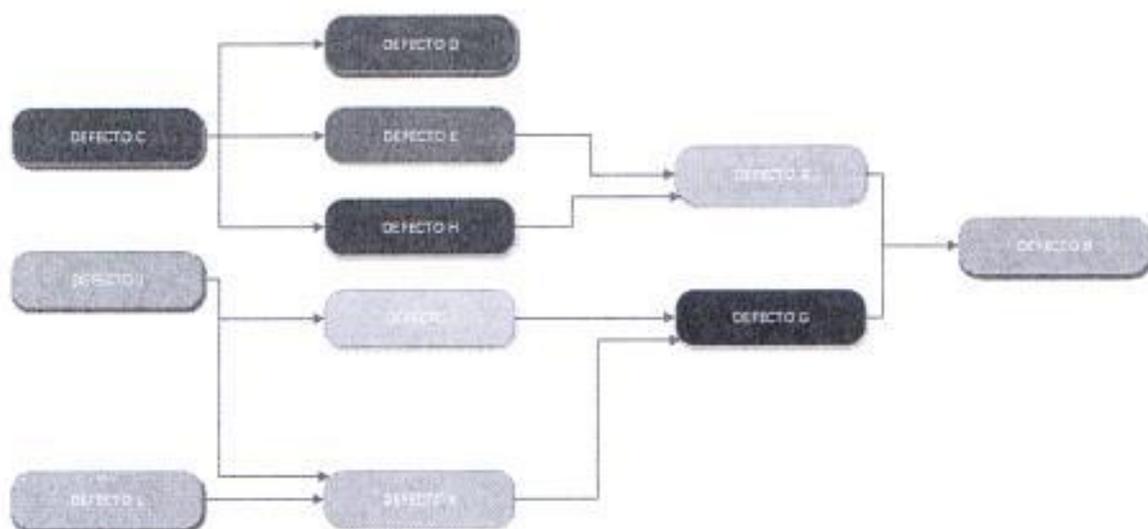


FIGURA 30 : Ejemplo de Red de Defectos.

En el caso que no se seleccione un defecto a solucionar el sistema presentará un orden óptimo sugerido para la solución de todos los defectos de la red. El criterio para la generación de orden óptimo es el número de relaciones originarias, donde mientras más relaciones originarias tenga el defecto mayor será su posición en el listado de soluciones sugerido. Para el caso anterior el orden óptimo para solucionar este defecto es:

DEFECTO	Nro. de Relaciones
	Originarias
C	0
J	0
L	0
D	1
E	1
H	1
I	1
K	2
A	2
G	2
B	2

TABLA 53 : Orden Óptimo para Solucionar Red de Defectos

Muchas veces los defectos tienen abundantes relaciones entre sí por lo cual es necesaria el aislamiento de los defectos que son causantes de un defecto en

particular. Por lo tanto se debería poder especificar un defecto como punto central del análisis y a partir de él encontrar su red de defectos que lo originan. Si continuamos usando el ejemplo anterior y nuestro objetivo es la solución del defecto G es posible aislarlo por lo cual la resultante sería el diagrama que se muestra a continuación.

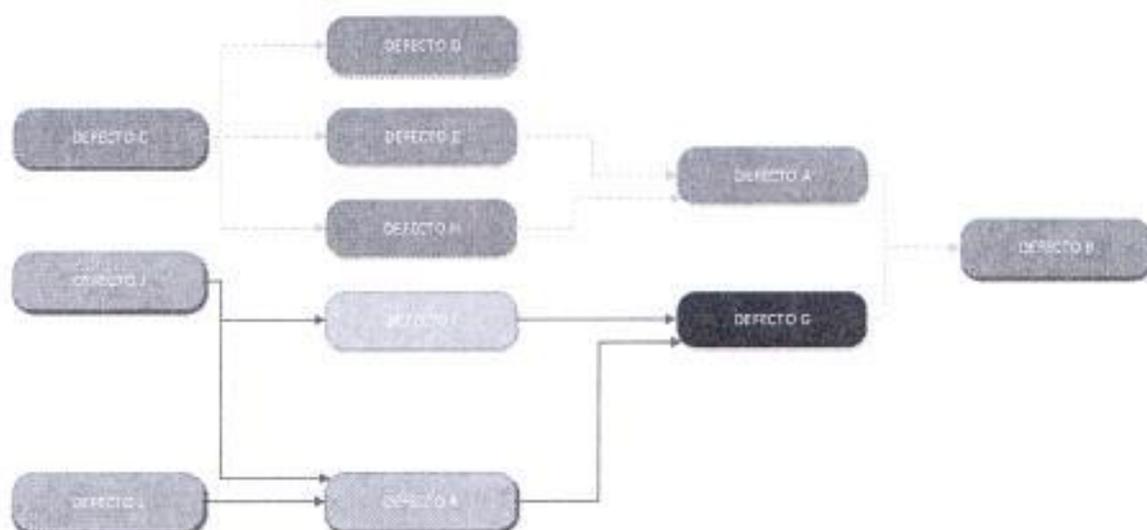


FIGURA 31 : Ejemplo de Red Aislada de Defectos.

El orden de solución sugerido por la red de la figura anterior se muestra a continuación:

DEFECTO	Nro. de Relaciones	
	Originarias	
J	0	
L	0	
I	1	
K	2	

<b>G</b>	<b>2</b>
----------	----------

TABLA 54 : Orden Óptimo para Solucionar Red Aislada

Este análisis de causalidad permite encontrar el orden óptimo de solución de defectos de software basado en el criterio de relaciones originarias. Esta opción es útil para un jefe de proyecto ya que permite obtener una secuencia de soluciones de defectos de software automáticamente, con lo que podría empezar la asignación de los defectos a los colaboradores de una forma ordenada y eficiente.

#### 4.7.2 Base de Acumulada de Conocimientos

El Sistema de Gestión de Defectos de Software permite el registro de defectos y sus características a través de un esquema de clasificación. Después de trabajar con varios proyectos informáticos el sistema creará una base de información valiosa que podría usarse en futuros proyectos informáticos ya que se almacena los defectos y sus procedimientos de solución. Se pretende que esta base de información sirva de ayuda para la solución de defectos futuros que presenten similares características.

Al registrar un defecto de software, el sistema sugerirá posibles soluciones ya que es muy probable que la información necesaria para resolver un defecto ya se encuentre en base de información. Claro está que no todos los defectos son iguales, sin embargo el sistema podría aportar con ideas de cómo se resolvió un problema parecido anteriormente. Una vez registrado un nuevo defecto en el sistema y al seleccionar la opción "Buscar Soluciones" se presentará una variedad de posibles soluciones a este problema.

Para presentar al usuario las posibles soluciones a un defecto específico el sistema parte de la premisa que un defecto anterior que presente un mayor porcentaje de atributos similares podría en mayor medida usarse como base para la solución del defecto analizado. De esta forma se busca entre experiencias previas que podrían ser aplicadas. Esta característica es de utilidad al laborar con grandes equipos de trabajo donde un desarrollador no necesariamente está en contacto constante con otros miembros del equipo que posiblemente ha tenido esa experiencia previa con un defecto similar. También es aplicable a equipos de trabajo donde el personal no es fijo y se cambia frecuentemente ya que es difícil que los antiguos y experimentados desarrolladores accedan a compartir sus conocimientos con nuevos integrantes del equipo. Esta opción se presentaría en la ventana de listado de soluciones de un defecto como se muestra a continuación:

## Lista de Soluciones

DEFECTO0DC3643

Home / Soluciones



FIGURA 32: Opción de Búsqueda de Soluciones de un Defecto.

Una vez realizada la búsqueda en la base de información de defectos se mostrará una lista de defectos que presenten similares características que las encontradas en el defecto seleccionado. Cada una incluirá un porcentaje de semejanza que permita cuantificar que tan similar son los defectos encontrados.

	PROYECTO	DEFECTO	Nro. Soluciones	% SEMEJANZA
<input type="checkbox"/>	Sistema Gestión Académico	DEFECTO A35	5	90 %
<input checked="" type="checkbox"/>	Sistema Gestión de Indicadores	DEFECTO C34	4	80 %
<input type="checkbox"/>	Sistema Gestión de Investigación	DEFECTO G32	3	60 %
<input type="checkbox"/>	Sistema Gestión Académico	DEFECTO A30	6	50 %

VER DETALLE DEFECTO    VER PROCESO    SELECCIONAR

FIGURA 33 : Defectos Similares Encontrados

Cada una de estas opciones representa una ruta de solución que el usuario puede seguir. El usuario será el encargado de aceptar o no las sugerencias del sistema por lo que tiene la posibilidad de ver la estructura del defecto sugerido de forma que pueda verificar la validez de la opción presentada. Otra de las opciones importantes es la opción "Ver Proceso", donde un usuario podría observar el diagrama de secuencia que muestra las actividades que se siguieron para llegar a la solución.

Es importante que en este diagrama de secuencia sea posible ver el flujo de actividades que se realizaron para solucionar el defecto ya que el usuario tendría que seguir exactamente la misma secuencia para obtener resultados similares. Es posible que esta secuencia de actividades no represente exactamente el problema sin embargo otorga al desarrollador una herramienta que le permita encontrar una solución válida.

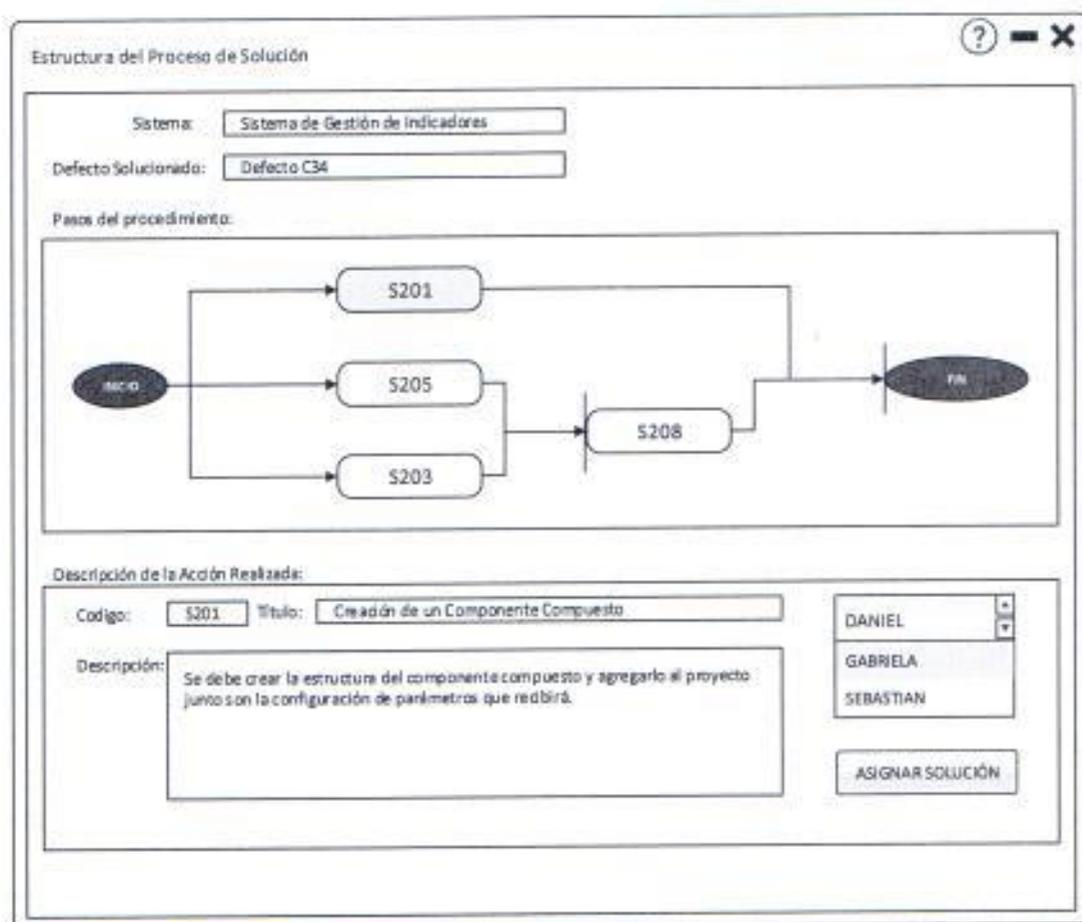


FIGURA 34 : Estructura de un Proceso de Solución de Defecto

En la figura anterior se puede observar un diagrama de secuencias que representa el proceso de solución de un defecto típico. Se puede observar que para ejecutar la secuencia existen actividades que necesitan ser completadas en un orden específico. Del mismo modo es posible asignar directamente el defecto a un colaborador junto con la o las actividades a ejecutar. Si el usuario logra la solución del defecto, esta nueva solución alimentará la base de conocimientos del sistema y permitirá utilizar en el futuro esta información para solucionar nuevos defectos.

### 4.7.3 Medidas de Prevención

El seguimiento de defectos y sus soluciones permiten aprender del pasado y tener una fuente de conocimiento que permita resolver errores en el futuro, pero si queremos prevenir la inyección de defectos es necesario crear acciones que prevengan su ocurrencia. Esto se puede realizar creando un grupo de reglas que permitan a los desarrolladores, diseñadores y arquitectos de sistemas informáticos evitar cometer errores en cualquiera de las fases del proceso de desarrollo de software. Estas reglas tratan de aclarar dudas y evitar malas prácticas al desarrollar sistemas informáticos.

Se ha realizado el diseño de los formularios que permiten al sistema la implementación de medidas que prevengan la ocurrencia de defectos. Las cuales pueden ser accedidas directamente a través del menú del sistema o al momento de registrar un defecto como solucionado. Para la creación de una medida preventiva es necesario seleccionar algunas características del proyecto las cuales permitirán identificar los tipos de proyectos en los que se recomienda deben ser aplicadas.

Las medidas de prevención son sugerencia que aparecerá a los desarrolladores a la hora de crear un nuevo proyecto de desarrollo informático en el sistema. Dependiendo si el proyecto creado presenta características similares se presentará una o varias medidas de prevención.

Además la especificación de estas medidas de prevención representa las mejores prácticas o guías importantes que impiden la ocurrencia de defectos al desarrollar proyectos informáticos.

Medidas Preventivas

Características del Proyecto:

lenguaje:	Jave
Herramientas:	Netbeans IDE
Metodología:	Agil
Fase del Desarrollo:	Desarrollo
Tecnologías:	Richfaces, Hibernate, Subversion, Inreport

Acción de Prevención:

Título:

Descripción:

FIGURA 35 : Diseño de Creación de una Medida de Prevención

Estas medidas de prevención proporcionan a los desarrolladores procedimientos importantes que se deben tomar en cuenta al trabajar bajo diferentes tecnologías. El conjunto total de medidas preventivas conforman las mejores prácticas y experiencias aprendidas por todos los desarrolladores de la empresa.

La creación de medidas de prevención es posible en cualquier momento del ciclo de vida del proyecto, sin embargo se podría realizar la creación de una medida justo después de resolver uno o varios defectos ya que la experiencia adquirida permite documentar y redactar recomendaciones en forma de una medida de prevención para tenerla presente en desarrollos futuros. A continuación se presenta la interfaz que presenta la lista de medidas de prevención:

Lista de Medidas de Prevención

Nuevo Edición Borrar Imprimir

Ingresar Texto Buscar

Filtros:

Lenguaje: Java

Herramientas: Netbeans IDE

Metodología: Agil

Fase del Desarrollo: Desarrollo

Tecnologías: Richfaces, Hibernate, Subversion, ireport

Ustado:

	Lenguaje	Herramienta	FASE	TITULO
<input type="checkbox"/>	JAVA	ECUPSE	DESARROLLO	Utilizar herramientas de generación de código
<input checked="" type="checkbox"/>	JAVA	NETBEANS	DESARROLLO	Componente para desplegar archivos en PDF.
<input type="checkbox"/>	Punto NET	VISUAL STUDIO	ANALISIS Y DISEÑO	Probar con código
<input type="checkbox"/>	PHP	ECUPSE	REQUERIMIENTOS	Participación activa de los clientes

⏪ ⏩ ⏴ ⏵

Crear Cancelar

FIGURA 36 : Listado de Medidas de Prevención

La creación de medidas de prevención evita la ocurrencia de defectos en cualquier etapa de desarrollo de un sistema informático. Por lo cual nos ayudan a incrementar la calidad de los productos y utilizar las mejores prácticas desarrolladas por la empresa y sus trabajadores.

## 4.8 Conclusiones

El diseño planteado para el Sistema de Gestión de Defectos toma ventaja de las últimas tecnologías en el desarrollo web, combina varias tecnologías para formular una aplicación de altas prestaciones y eficiente. La arquitectura de la aplicación es multicapa e incluye un manejador de persistencia que es el que controla la transaccionalidad de la aplicación y genera las consultas a la base de datos. Se desarrolló el diseño y modelamiento completo del sistema de gestión de defectos donde se incluyó diagramas de clase, entidad relación, casos de uso, diseño de interfaces, y diseño de reportes.

Se creó mecanismos de gestión a través de los cuales es posible encontrar los orígenes de los defectos utilizando análisis de causalidad, además se planteó la utilización de la base de conocimiento para la reutilización de la información histórica y se creó la posibilidad de establecer medidas que permitan prevenir la ocurrencia de defectos en proyectos futuros. Partiendo de estos diseños y mecanismos de gestión es posible el desarrollo de una aplicación web que cumpla con los objetivos planteados por este proyecto de tesis.

# CAPÍTULO 5

## Desarrollo y Pruebas

En este capítulo se realiza la planificación, el desarrollo y las pruebas con el objetivo de validar los componentes que conforman el sistema. Se presentarán los resultados obtenidos, las posibles aplicaciones del sistema y las recomendaciones para futuras versiones.

### 5.1 Planificación del Proyecto

#### 5.1.1 Recursos

Para la realización del proyecto fue necesario contratar los servicios de un ingeniero de sistemas y un programador. Además es necesario para ponerlo en producción de un administrador que realice un seguimiento de la aplicación y monitoreo del rendimiento.

#### 5.1.2 Calendario de Actividades

El plan de trabajo muestra la planificación de actividades realizadas durante la ejecución del proyecto.

AÑO	2013						2014								
MES	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8
Análisis de Requerimientos															
Diseño y Modelamiento															
Desarrollo															
Implementación															
Pruebas															

TABLA 55 : Calendario de Trabajo



Ingresos									
Nro Usuarios		100	200	300	400	500	600	700	800
Costos Suscripción		49	49	49	49	49	49	49	49
<b>Total Ingresos</b>	<b>0</b>	<b>4900</b>	<b>9800</b>	<b>14700</b>	<b>19600</b>	<b>24500</b>	<b>29400</b>	<b>34300</b>	<b>39200</b>
<b>Flujo de Efectivo</b>	<b>-9244</b>	<b>-6188</b>	<b>-1288</b>	<b>3612</b>	<b>8512</b>	<b>13412</b>	<b>18312</b>	<b>23212</b>	<b>28112</b>

<b>TASA DE DESCUENTO</b>	<b>10%</b>
<b>VAN:</b>	<b>\$ 36.283,97</b>
<b>TIR</b>	<b>37%</b>

TABLA 58 : Flujo Proyectado para 8 años

Se puede observar que el proyecto presenta un valor actual neto positivo por lo tanto el proyecto es rentable y se calculó un valor TIR del 37 % es la tasa de interés máxima aceptable para el dinero invertido en el proyecto. Además es necesario especificar que esta proyección de flujos utiliza para su cálculo los siguientes supuestos:

- Un costo de suscripción fijo de \$ 49 USD anuales por cuenta de usuario.
- Un sueldo fijo del administrador del sitio de \$ 500 USD.
- Un alquiler de los Servicios de Amazon por una instancia EC2 con un valor de 0.052 USD la hora de alquiler.
- Una inversión mensual de \$200 USD para las campañas de marketing.
- Un alquiler de \$200 USD para la creación de una oficina de atención al usuario y desarrollo de aplicaciones.

## 5.2 Desarrollo del Sistema de Gestión de Defectos

### 5.2.1 Descripción del Proceso de Desarrollo

El desarrollo de la aplicación se realizó a través de un proceso incremental, se partió de la definición de los requerimientos y diagramas diseñados en los

capítulos anteriores. Se creó las estructuras de datos y los mecanismos de inicio de sesión, luego se realizó la creación de las diferentes opciones del sistema. Finalmente se creó los reportes de análisis y se validó el acceso a la información para el usuario.

### **5.2.2 Estructura de la Aplicación Web**

La estructura empleada para el desarrollo de la aplicación fue generada bajo una plantilla Maven <sup>7</sup>para aplicaciones web con "Maven Archetype Plugin". Es conveniente la generación de la estructura inicial del proyecto a través de esta plantilla porque nos permite usar un estándar al construir aplicaciones web usando el lenguaje Java. Además el uso de Maven en nuestro proyecto permite la organización automática de librerías, dependencias y configuraciones necesarias para que funcione correctamente nuestra aplicación.

---

<sup>7</sup> Apache Maven. Es una herramienta de software para la gestión y construcción de proyectos Java.

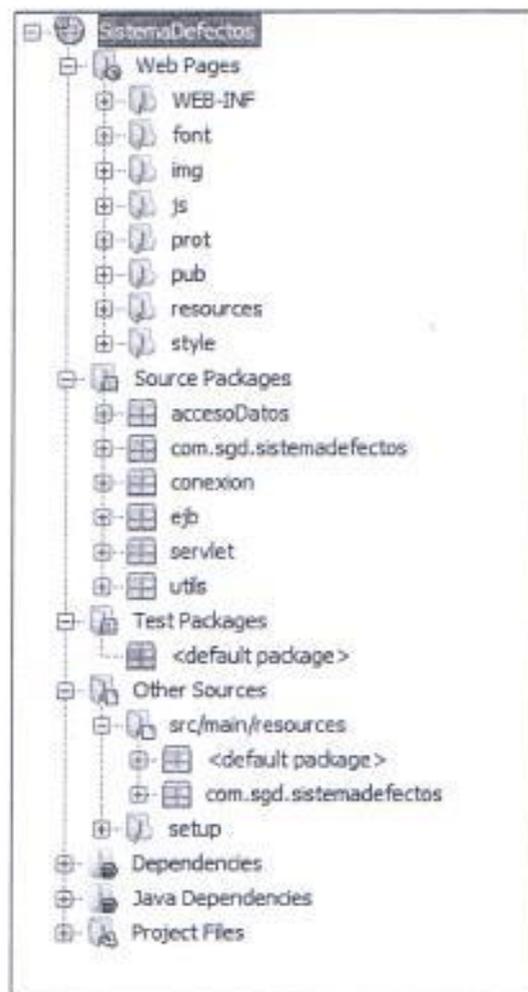


FIGURA 37 : Estructura de la Aplicación Basada en Plantilla Maven.

### 5.2.3 Tecnologías Utilizadas para el Desarrollo

Las tecnologías que se utilizaron en el desarrollo de esta aplicación fueron:

1. HTML5 .- Se utilizó esta tecnología para tomar ventaja sobre las última tecnología de desarrollo web por la funcionalidad agregada que puede ofrecer a la aplicación.
2. Java Enterprise Edition 7 .- Esta plataforma de desarrollo integra muchas tecnologías para el desarrollo web y provee de librerías

- específicas para reducir el tiempo de desarrollo, la complejidad y mejoran el desempeño de la aplicación.
3. Richfaces 4.3 .- Es una librería de componentes que permiten la rápida construcción de aplicaciones web usando componentes que integran en su estructura tecnología AJAX.
  4. Arborjs 0.92 .- Se utilizó esta librería Javascript para la generación de diagramas de causalidad ya que presenta una fortaleza en la presentación gráfica y estructuras jerárquicas.
  5. Hibernate 3.6 .- Se empleó Hibernate como manejador de persistencia ya que permite un control sobre las consultas realizadas a la base de datos y mantiene la persistencia de los objetos. Además permite realizar consultas a través de un lenguaje de consultas orientado a objetos.
  6. JasperReports 5 .- Se utilizó este motor para la generación de reportes ya que permite generar documentos en diferentes formatos con altas características de calidad en gráficas y componentes
  7. Estilos Bootstrap .- Se utilizó esta librería de estilos para la definición de los estilos de las páginas del sistema, no solamente porque tienen un gran atractivo visual sino que además permiten la creación de sitios web adaptativos, que podrían abrirse en varios dispositivos.

#### **5.2.4 Herramientas Utilizadas en el Desarrollo**

Las herramientas que se utilizaron para desarrollar esta aplicación fueron:

1. Netbeans 7.4 .- Este entorno de desarrollo integrado nos permitió la creación de la aplicación, realizar las pruebas y depuraciones del código de la aplicación.
2. Glassfish 4 .- Fue usado como el servidor para el despliegue de la aplicación.
3. IReport 5.6 .- Es la herramienta empleada para el diseño de reportes presentados en el sistema, a través de esta herramienta se crearon las plantillas para los reportes generados por la aplicación.

### **5.3 Despliegue del Sistema de Gestión de Defectos**

#### **5.3.1 Configuración de Entorno**

El proceso de despliegue de la aplicación se realizó utilizando los Servicios Web de Amazon donde se requirió la creación de una cuenta de usuario. El servicio de Computación en la Nube de Amazon permite la creación de máquinas virtuales con características específicas. Una de las ventajas más importantes al usar este servicio es que tiene una capa de acceso gratuito donde se permite a un desarrollador realizar pruebas y prototipos de aplicaciones web por lo que presenta las características ideales para implementar la aplicación.

La máquina virtual que se contrató tiene las siguientes características:

Tipo de Instancia	vCPU	Memoria (GIB)
t2.micro	1	1
<b>t2.small</b>	1	2
t2.medium	2	4

TABLA 59 : Micro Instancias de Bajo Costo de Nube Elástica de

Amazon

Se intentó realizar la implementación del sistema sobre una instancia T1 con una memoria de 0.613 GiB que es parte de la capa gratuita pero se vio un rendimiento limitado, por lo que se seleccionó la instancia "t2.small" ya que tiene un aumento a 2GIB RAM con lo cual se logró que la aplicación funcionara correctamente. La instancia T2 está creada bajo un procesador Intel Xeon con velocidad de reloj de hasta 3.3GHz, tiene una conexión a internet de 631.49 Mbps de carga y 306.84 Mbp de subida. El precio por la instancia es de \$ 0.036 USD por hora de uso, lo cual equivale a un aproximado de \$ 12.96 USD mensuales si es usado 12 horas diarias. Una vez con vez configurada la máquina virtual se procedió a la instalación del servidor de aplicaciones Glassfish versión 4 y la base de datos Oracle XE. Además se realizó la configuración de los usuarios y la estructura de la base de datos.

### 5.3.2 Optimización de la Aplicación

Se realizó varias actividades para que optimizar el funcionamiento de la aplicación, entre las que se puede mencionar:

- Cambios de la configuración del servidor de Aplicaciones Glassfish.
- Se realizó creación de un Pool de Conexiones el cual permite optimizar el uso de las conexiones a la base de datos.

## **5.4 Pruebas de la Aplicación**

### **5.4.1 Configuración de las Pruebas**

Las pruebas de carga se realizaron a través del programa NeoLoad en su versión 5.0.0 bajo una licencia gratuita de uso. Esta licencia de prueba permite realizar pruebas de carga para un máximo de 50 usuarios virtuales. Debido al bajo ancho de banda utilizado para las pruebas se realizó pruebas para 10 y 15 usuarios simultáneos. La computadora cliente posee una conexión a internet de bajada de 2.60 Mbps y subida de 0.94 Mbps.

Se realizó bajo un escenario común, donde un usuario realiza la siguiente secuencia de actividades:

1. Navegar a la página de inicio de sesión.
2. Ingresar al sistema.
3. Navegar a la página principal de la aplicación.
4. Abrir un defecto de un proyecto específico.
5. Realizar cambios en el defecto.
6. Generar un reporte con los datos estadísticos del proyecto.
7. Navega a la opción de reportes del sistema.
8. Generar un reporte por síntomas.
9. Salir del sistema.

La prueba de carga se realizó con un incremento de 2 usuarios virtuales cada 10 segundo por lo que existe un periodo de tiempo antes de llegar al máximo de usuarios usados en la prueba.

#### 5.4.2 Análisis de Resultados

Los resultados obtenidos a través de NeoLoad son presentados a través de gráficas y resultados estadísticos del análisis, a continuación se presentan los resultados obtenidos en las pruebas realizadas.

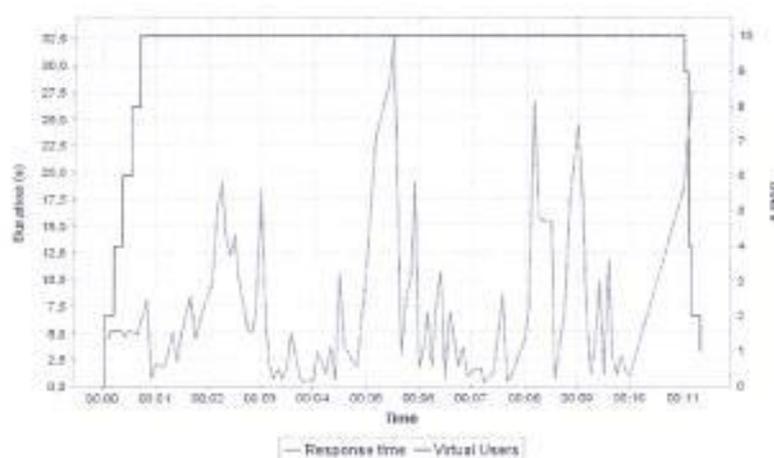


FIGURA 38 : Tiempo Promedio de Respuesta de la Página

Mínimo	Promedio	Máximo	Medía	Desviación Estándar
0.354	7.409	32.724	4.952	7.877

TABLA 60 : Resultados Estadísticos del Tiempo Promedio de Respuesta

El tiempo de respuesta nos indica que tal rápido se realizó la carga de las páginas del sistema. Se puede observar que el promedio de respuesta de las páginas es de 7 segundos, una alta variabilidad y una media de 4.9 segundos que es un valor aceptable bajo las condiciones de la prueba.

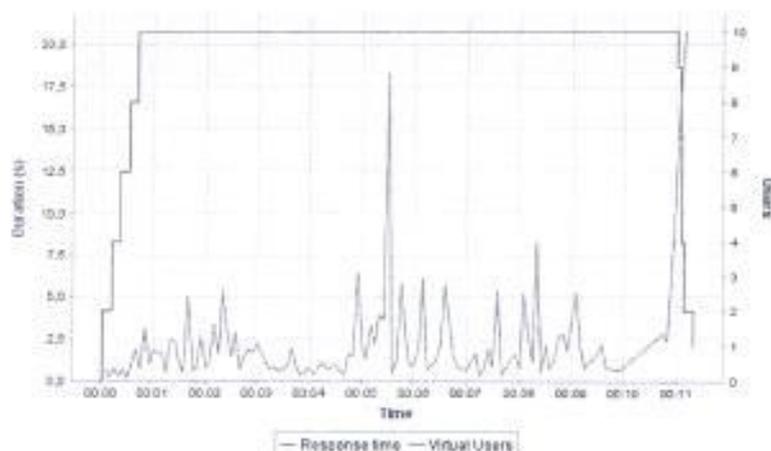


FIGURA 39 : El Tiempo Promedio Respuesta por Petición

Mínimo	Promedio	Máximo	Medio	Desviación Estándar
0.197	1.87	20.762	1.271	2.197

TABLA 61 : Resultados Estadísticos del Tiempo Promedio de Petición

El tiempo de respuesta por petición nos indica que tal rápido se dio la respuesta a la petición realizadas en la prueba. Se puede observar que el promedio de respuesta por petición es de 1.87 segundos, una media variabilidad y una media de 1.271 segundos que es un valor aceptable bajo las condiciones de la prueba.

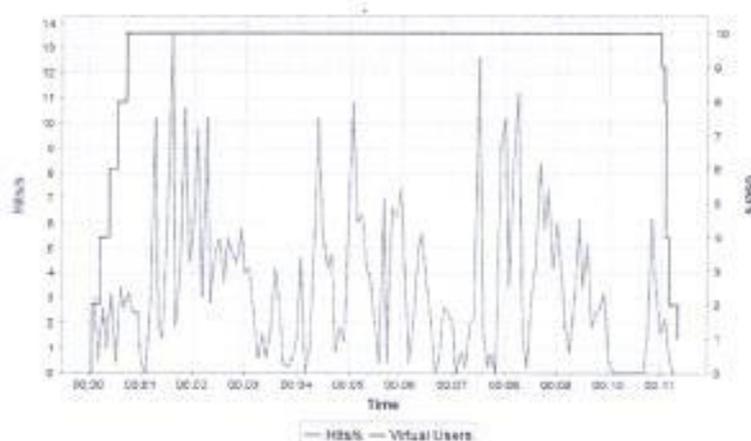


FIGURA 40 : Peticiones por Segundo

Mínimo	Promedio	Máximo	Media	Desviación Estándar
0	3.4	13.6	2.6	3.1

TABLA 62 : Resultados Estadísticos de las Peticiones por Segundo

El número de peticiones por segundo representa en que momentos de la prueba se dio la mayor cantidad de peticiones. Se puede observar que el promedio de peticiones es de 3.4 peticiones por segundos, con cierto grado de variabilidad y una media de 2.6 peticiones. Se puede observar que existe un incremento de peticiones mientras más usuarios simultáneos tratan de usar el sistema.

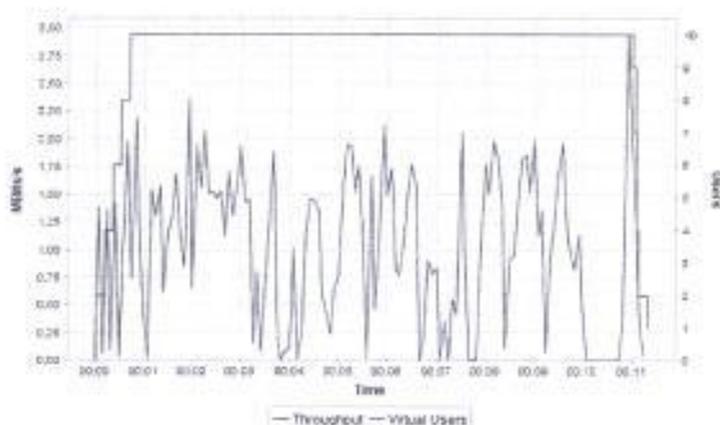


FIGURA 41 : Tráfico

Mínimo	Promedio	Máximo	Media	Desviación Estándar
0	1	2.94	1.06	0.71

TABLA 63 : Resultados Estadísticos del Tráfico

El tráfico representa el tamaño total de las respuestas de todas las peticiones realizadas durante la prueba. Se puede observar que el promedio de peticiones es de 1 Mbps/s, con una variabilidad baja y una media de 1.06 Mbps/s.

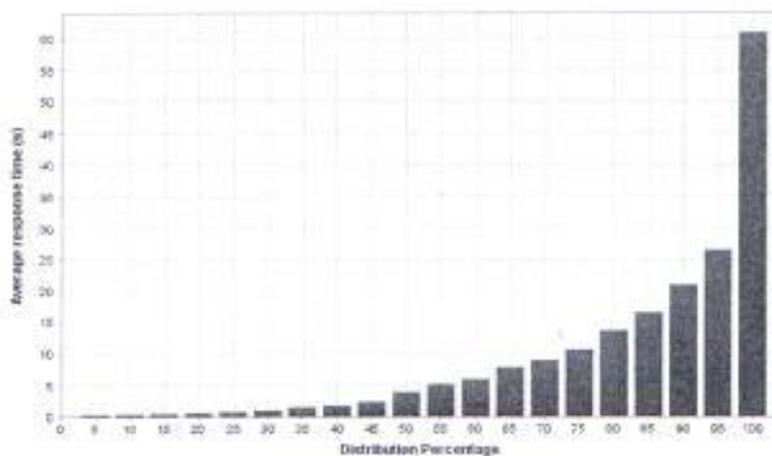


FIGURA 42 : Tiempo de Respuesta Versus Carga

Gracias a la gráfica anterior se puede inferir que un 80% de las páginas tiene un tiempo de respuesta promedio por debajo de los 14 segundos.

De los resultados obtenidos de las pruebas de la aplicación se puede observar un rendimiento aceptable de la aplicación. Se observó además que para un total de 10 usuarios simultáneos no se presentaron errores en la aplicación mientras que al aumentar los usuarios a 15 ya se empezó a visualizar errores por tiempo de espera excedido y un incremento en el tiempo de respuesta. Por lo tanto bajo las condiciones y limitaciones en ancho de banda que presenta la maquina cliente es posible únicamente realizar una conexión al sistema de hasta 10 usuarios simultáneos con un ancho de banda de 2.60 Mbps.

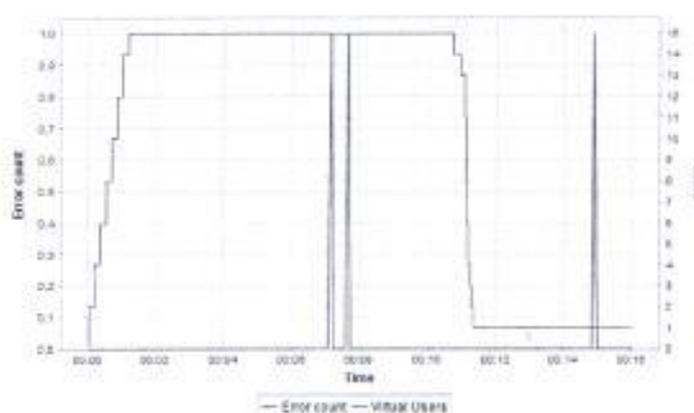


FIGURA 43 : Errores de la Aplicación para 15 Usuarios Simultáneos

## 5.5 Análisis de Mejoras

La aplicación podría mejorar su funcionamiento en futuras versiones para que sea una herramienta flexible y adaptable a cualquier tipo de esquemas de clasificación de defectos. Además se ha visto la necesidad de incluir otras características al desarrollar la aplicación que quedan fuera de los objetivos de esta tesis.

1. Tener la posibilidad de subir imágenes junto a la creación de los defectos, ya que muchas veces son necesarias pantallas que muestren la ocurrencia de un defecto de software.
2. Se podría codificar un mecanismo que genéricamente permita configurar reportes que se adapten a cualquier esquema de clasificación de defectos.
3. Es posible la implementación de una base de datos Ldap para el manejo eficiente de usuario y mejorar la seguridad de la aplicación.

Se sugiere incluir un servidor de autenticación para proteger la aplicación.

4. Es posible una optimización de la interfaz gráfica para que se adapte completamente a cualquier dispositivo móvil.

## 5.6 Conclusiones

Al finalizar el desarrollo de este Sistema de Gestión de Defectos de Software se puede observar que es un sistema web de última tecnología que trabaja con una modalidad de software como servicio que se encuentra en la nube de servicios informáticos de Amazon. Este sistema se ha desarrollado siguiendo los parámetros de diseño que responden a los requerimientos establecidos en el análisis de requerimientos realizado.

Este sistema informático utiliza Java y su paquete de desarrollo en su versión 7.0, además utiliza tecnologías como AJAX en sus componentes e implementa las mejores prácticas para el desarrollo de aplicaciones web. Se logró optimizar el código fuente y memoria utilizada para la ejecución de la aplicación. Esto se demostró en las pruebas de la aplicación ya que presentan características aceptables en cuanto a eficiencia y tiempos de respuesta.

Para las interfaces de la aplicación se utilizó estilos que presenten la información de forma ordenada y permitan gestionar la información de los defectos de varios proyectos a la vez. Además se utilizó una plantilla adaptativa para que el sistema pueda ser accedida mediante cualquier dispositivo.

## CONCLUSIONES Y RECOMENDACIONES

La gestión de defectos de software al desarrollar o actualizar sistemas información es una tarea compleja, los retos de los gerentes y jefes de proyectos hoy en día se ven aumentados con la aparición de nuevos lenguajes, sistemas operativos y cambios constantes en las necesidades de los usuarios. En un mundo globalizado la mejor estrategia es adaptarse, tomar ventaja y aplicar nuevas tecnologías en nuestros ambientes de trabajo.

Un sistema para la gestión de defectos de software permite a los jefes de proyectos informáticos analizar sus causas, entender sus consecuencias y tomar decisiones acertadas sobre cómo tratarlos de forma que tengan el menor impacto posible. Al desarrollar de este sistema se estableció las bases teóricas para el tratamiento de defectos de software, se describió los tipos de defectos, esquemas de clasificación y se revisó las metodologías de análisis existentes. Se buscó satisfacer las necesidades que tiene un jefe de proyectos que trabajan con sistemas informáticos que presentan defectos, ya sea que fueron desarrollados en el pasado o que se encuentran en desarrollo actualmente.

Este proyecto de tesis se centró en resolver el problema que tienen el equipo de desarrollo al utilizar una ingeniería basada en modelos, metodología que se encuentra en desarrollo y que promete resolver muchos de los problemas que existen al desarrollar aplicaciones optimizando el uso de recursos y minimizando el tiempos de entrega. Esta metodología se basa en el diseño

eficiente de modelos conceptuales para luego generar automáticamente código de aplicación. El sistema no solamente integró un mecanismo que permita registrar defectos sino también trabaja con un esquemas de clasificación de defectos para modelos conceptuales.

Al incluir un esquema de clasificación con el sistema de registro de defectos se logró almacenar información relacionada con atributos específicos y categorizar los defectos encontrados de forma que se puede realizar varios tipos de análisis. Estos análisis permiten al usuario observar los defectos desde varias perspectivas donde cada una agrega información relevante y permite la toma efectiva y mejorada de decisiones por parte de los administradores de proyectos. Se plantean mecanismos de gestión y documentación de experiencias al desarrollar proyectos informáticos. Se provee una opción para la búsqueda de soluciones posibles para un defecto, donde las soluciones encontradas presenten un diagrama de secuencias con las actividades necesarias para lograr la solución de un defecto. Además se ha incluido la creación de medidas de prevención que ayuden a evitar la ocurrencia de defectos en proyectos futuros.

El sistema planteado posee un entorno fácil de utilizar y simplificado, además permite la comunicación con el equipo de desarrolladores para trabajar en varios proyectos diferentes con distintas necesidades, los cuales son ambientes comunes en empresas desarrollo y departamentos de desarrollo informático.

Para la continuidad del presente estudio se recomienda la modificación del sistema presentado de forma que se generalice su aplicación para diferentes tipos de esquemas de clasificación. Además se recomienda el desarrollo de una

herramienta que permita a los administradores de proyectos la creación de reportes. Se aconseja la difusión del proyecto ya que se podría realizar investigaciones que tomen esta herramienta como punto de partida para nuevos estudios y proyectos de tesis. El estudio de los defectos de software es un terreno fértil donde cada proyecto nuevo contribuirá a que se realicen verdaderos avances en esta área.

## GLOSARIO

### **Model- Driven Engineering (MDE)**

En su traducción "Ingeniería Basada en Modelos". Permite la generación automática de aplicaciones partiendo de modelos conceptuales.

### **Software as Service (SAS)**

En su traducción "Software como Servicio". Es un modelo de distribución de software donde el soporte lógico y los datos que maneja se alojan en servidores de una compañía de tecnologías de información y comunicación, a los que se accede vía Internet desde un cliente.

### **Web 2.0**

Comprende aquellos sitios web que facilitan el compartir información, la interoperabilidad, el diseño centrado en el usuario y la colaboración en la World Wide Web. Un sitio Web 2.0 permite a los usuarios interactuar y colaborar entre sí como creadores de contenido generado por usuarios en una comunidad virtual, a diferencia de sitios web estáticos donde los usuarios se limitan a la observación pasiva de los contenidos que se han creado para ellos.

### **Instituto de Ingeniería Eléctrica y Electrónica (IEEE)**

Es una asociación mundial de técnicos e ingenieros dedicada a la estandarización y el desarrollo en áreas técnicas. Su trabajo es promover la creatividad, el desarrollo y la integración, compartir y aplicar los avances en las tecnologías de la información, electrónica y ciencias en general.

## **Hewlett-Packard (HP)**

Es una de las mayores empresas de tecnologías de la información del mundo, fabrica y comercializa hardware y software además de brindar servicios de asistencia relacionados con la informática.

## **Herramientas CASE (Ingeniería de Software Asistida por Computadora)**

Son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costos, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

## **JavaServer Faces (JSF)**

Es una tecnología y framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java para la web.

## **MAVEN**

Es una herramienta de software para la gestión y construcción de proyectos Java. Es similar en funcionalidad a Apache Ant, pero tiene un modelo de configuración de construcción más simple, basado en un formato XML.

## **Richfaces**

Es una biblioteca de código abierto basada en Java que permite crear aplicaciones web con Ajax.

## **AJAX**

Es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asincrónica con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.

## **NETBEANS**

Es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Soporta el desarrollo de Aplicaciones empresariales con Java, incluyendo herramientas de desarrollo visuales, herramientas de esquemas XML, servicios web, y modelado UML.

## **Subversion**

Es una herramienta de control de versiones gratuita basada en un repositorio cuyo funcionamiento se asemeja enormemente al de un sistema de ficheros.

## **Hibernate**

Es una herramienta de Mapeo objeto-relacional para la plataforma que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones.

## **Glassfish**

Es un servidor de aplicaciones de software libre desarrollado por Sun Microsystems, compañía adquirida por empresa Oracle Corporation, que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación.

## **JAVA EE (Edición Empresarial)**

Es una plataforma de programación parte de la Plataforma Java para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java. Permite utilizar arquitecturas de N capas distribuidas y se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

## **Oracle XE**

Posee todas las funcionalidades de una base de datos relacional, es gratuita y se utiliza para el desarrollo de prototipos y aplicaciones de pruebas.

### **Entorno de Desarrollo Integrado (IDE)**

Es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios.

### **Cloud Computing**

Es un paradigma que permite ofrecer servicios de computación a través de Internet. En este tipo de computación todo lo que puede ofrecer un sistema informático se ofrece como servicio, de modo que los usuarios puedan acceder a los servicios disponibles "en la nube de Internet" sin conocimientos (o, al menos sin ser expertos) en la gestión de los recursos que usan.

### **Unified Modeling Language (UML)**

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

### **Domain- Specific Modeling Languages (DSML)**

Un lenguaje específico del dominio es un lenguaje de programación o especificación de un lenguaje dedicado a modelar un problema en particular, representar un problema específico y proveer una técnica para modelar una situación particular.

## **International Business Machines**

Es una empresa multinacional estadounidense de tecnología y consultoría con sede en Armonk, Nueva York. IBM fabrica y comercializa hardware y software para computadoras, y ofrece servicios de infraestructura, alojamiento de Internet, y consultoría en una amplia gama de áreas relacionadas con la informática, desde computadoras centrales hasta nanotecnología.

## **App Store**

App Store es un servicio para el iPhone, el iPod Touch, el iPad y Mac OS X Snow Leopard o posterior, creado por Apple Inc, que permite a los usuarios buscar y descargar aplicaciones informáticas, desarrolladas con el iPhone SDK y publicadas por Apple. Estas aplicaciones están disponibles para ser compradas o bien gratuitas, dependiendo de cada una.

## **Amazon Web Services**

Es una colección de servicios de computación en la nube (también llamados servicios web) que en conjunto forman una plataforma de computación en la nube, ofrecidas a través de Internet por Amazon.com. Es usado en aplicaciones populares y es una de las ofertas internacionales más importantes de la computación en la nube.

## **Amazon Elastic Cloud (EC2)**

Es una parte central de la plataforma de cómputo en la nube de la empresa Amazon.com denominada Amazon Web Services (AWS). EC2 permite a los usuarios rentar computadores virtuales en los cuales poder correr sus propias

aplicaciones. Este tipo de servicio supone un cambio en el modelo informático al proporcionar capacidad informática con tamaño modificable en la nube, pagando por la capacidad utilizada. En lugar de comprar o alquilar un determinado procesador para utilizarlo varios meses o años, en EC2 se alquila la capacidad por horas.

### **Licencia Común de Desarrollo y Distribución**

Es una licencia de código abierto y libre, producida por Sun Microsystems, basada en la Mozilla Public License o MPL, versión 1.1. La licencia CDDL fue enviada para su aprobación al Open Source Initiative el 1 de diciembre de 2004, y fue aprobada como una licencia de código abierto a mediados de enero de 2005.

### **Licencia Pública General de GNU**

Es la licencia más ampliamente usada en el mundo del software y garantiza a los usuarios finales (personas, organizaciones, compañías) la libertad de usar, estudiar, compartir (copiar) y modificar el software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

### **NeoLoad**

NeoLoad es una herramienta de diseño de pruebas de carga para sistemas. Esta fue diseñada para probar aplicaciones web y móviles, de forma que se pueda simular realísticamente actividad de usuarios y monitorear el comportamiento en la infraestructura.

## REFERENCIAS

- Bridge, N. (1996). Using Defect Data to Improves Software Development. *Motorola Corporate Software Center*, 15.
- Colin Atkinson, T. K. (2003). Model-Driven Development: A Metamodeling Foundation. *IEEE Computer Society*, Pagina 1 - 6.
- France, R. (2007). Model- Driven Development of Complex Software: A Roadmap. *FOSE Future of Software Engineering*, Pages 37-54.
- Freimut, B. (2001). Developing and Using Defect Classification Schemas. *Fraunhofer IESE IESEReport No*, 40.
- Group, I. (2013). Standard Classification for Software Anomalies. *IEEE, New York*, 15.
- Jubair Basha, A. y. (2011). Model Based Software Development: Issues & Challenges. *Special Issue of International Journal of Computer Science*, 226-230.
- Shmidt, D. C. (2006). Model-Driven Engineering. *IEEE Computer Society*, Paginas 25 - 31.
- Wagner, S. (2008). Defect Classification and Defect Types Revisited Categories and Subject Descriptors. *Boltzmannstr*, 3, 39 - 40.