



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACION

Sistema de gestión de defectos para proyectos de una organización financiera.

Trabajo Especial de Grado presentado ante la ilustre
Universidad Central de Venezuela realizado por los bachilleres:

Álvaro David Marciales Llamaza,
Claudio Edmundo Torrez Huanca.

Para optar al título de
Licenciado en Computación

Tutora: Profa. Concettina Di Vasta

Caracas, 16 de Octubre de 2015

ACTA

Quienes suscriben, miembros del Jurado designado por el Consejo de Escuela de Computación de la Facultad de Ciencias de la Universidad Central de Venezuela, para examinar el Trabajo Especial de Grado titulado: “Desarrollo de un Sistema de Gestión de Defectos para los proyectos implementados en una Organización Financiera”, presentado por el bachiller Claudio Edmundo Torrez Huanca, portador de la cédula de identidad V.- 20.365.878 y el bachiller Álvaro David Marciales Llamaza portador de la cédula de identidad V.- 20.613.004 a los fines de optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

Dicho trabajo, leído por cada uno de los miembros del jurado, se fijó el día 26 de octubre de 2015 a las 5:30pm, para que sus autores lo defendieran en forma pública, se hizo en el aula PB III de la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual se respondieron las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo con la nota de _____ puntos.

En fe de lo cual se levanta la presente Acta, en Caracas a los 26 días del mes de octubre del año dos mil quince, dejando constancia de que actuó como coordinador del jurado la Profesora Concettina Di Vasta.

Prof. Concettina Di Vasta

(Tutor)

Prof. Aparicio Peña

(Jurado)

Ing. Vito Perricone

(Jurado)

DEDICATORIA

A Dios y a la Virgen,
Por estar siempre ahí, guiándome
En cada paso de mi vida y haberme dado la fuerza necesaria
Para ser siempre constante y así lograr todo lo que me proponga
Sigán ayudándome...

A mis padres,
Por todos los esfuerzos y sacrificios
Que hacen por ayudarme a salir adelante,
Todo lo que soy se lo debo a ustedes

A mis hermanos,
Por su constante apoyo y creer siempre en mí.

A toda mi familia,
Y los que son como familia
Por estar siempre allí

Claudio Edmundo Torrez Huanca

A mi Padre Celestial,
JESUCRISTO REY DE REYES Y SEÑOR DE SEÑORES
que gobierna sobre todo el universo
y a El sea toda la Gloria y la Honra
por los siglos de los siglos Amén.

Álvaro David Marciales Llamaza

AGRADECIMIENTOS

A Dios primeramente por haberme permitido pasar por este proceso de aprendizaje y formación profesional con amor y misericordia.

A mi compañero de tesis por haber formado parte de mi equipo de trabajo en este proyecto y por sus aportes de conocimientos y experiencias.

A mis hermanos que a pesar de este largo y arduo proceso de formación profesional siempre estuvieron presentes en mis batallas.

A la tutora Concettina Di Vasta por habernos ayudado y aconsejado en gran manera y además por su gran aporte de conocimientos con paciencia y mansedumbre.

A la Universidad por haberme permitido prepararme profesionalmente y académicamente para el campo laboral.

A mis compañeros de estudio que formaron parte de mi experiencia profesional y personal.

Álvaro David Marciales Llamaza.

A Dios y a la Virgen, por esa fe que han otorgado a toda nuestra familia.

A mis padres, Hilaria y Claudio, abnegados padres, gracias por darme la vida, preocuparse siempre por mí y ayudarme con lo que han podido, todo esto es gracias a ustedes.

A mis hermanos, Rommel y María del Rosario por su amor incondicional y ser los mejores ejemplo para una persona. Siempre estaremos juntos.

A nuestra tutora, Tina, gracias, muchas gracias por su colaboración, el esfuerzo y la paciencia que tuvo hacia Álvaro y a mí.

A mi compañero de tesis, Álvaro, y toda la gente buena de computación que de una u otra manera, pude conocer y compartir momentos juntos.

A mi padrino Ernesto y tía Rommy, por toda su colaboración y apoyo, de los que siempre han estado ahí preocupándose por mí. Sus preocupaciones también son mías, Martín y Luis también llegaron aquí.

A todos mis primos y sobrinos, para quienes siempre estaré disponible y procuraré darles un buen ejemplo.

A toda mi familia que se encuentra en Bolivia, donde a pesar de la altura siempre tienen los pies sobre la tierra.

A mi otra familia, el Sporting, Freddy, Alejandro, Jesús, Víctor, Keyber, Marco, Wilfredo y Ana y el equipo de Ciencias City con los que pase los mejores años y aventuras en la universidad, aún quedan varias por vivir.

A mi universidad, la casa que vence las sombras. Por permitir mi formación académica y profesional, ser testigo de mi crecimiento personal, ser mi segundo hogar y hacerme parte de ella.

A la gente de Pranical, por darme la oportunidad y otorgarme sabiduría para poder desarrollar esta tesis.

A cada persona con la que me pude haber cruzado, porque de alguna forma, todas esas coincidencias, me hicieron llegar aquí.

Gracias a todos,

Claudio Edmundo Torrez Huanca

RESUMEN

El objetivo del presente Trabajo Especial de Grado consiste en el desarrollo de un Sistema de Gestión de Defectos que permita el registro y seguimiento de los mismos, logrando así un control eficiente por parte de los empleados del área de tecnología de la información de la organización financiera. Este sistema brindará una gestión más eficiente sobre los defectos encontrados en los múltiples proyectos que se puedan estar certificando en los ambientes de Calidad, para llevar a cabo su respectiva solución de manera rápida y eficaz, ya que al no detectarse un defecto en ambiente de Calidad, él mismo podría generar incidencias en ambiente de Producción.

Durante mucho tiempo se ha utilizado un documento Excel para el almacenamiento y el envío de correo electrónicos como vía de notificación entre los usuarios que prueban los sistemas y el área de Tecnología de la Información de la organización para el reporte de estos defectos. Existen, sin embargo, soluciones más adecuadas que facilitan la resolución de problemas, acortando los tiempos de espera y mejorando el rendimiento. Luego de un análisis de varios sistemas para la gestión de defectos (MantisBT, Jira), se llevara a cabo el desarrollo de un sistema propio de la organización cuyas funcionalidades e implementación se describirán en este documento.

Para el desarrollo de este sistema, se aplicaron las metodologías y herramientas propias de la organización, en este caso una adaptación de la metodología de desarrollo de software RUP (Proceso Unificado de Racional) combinado con una serie de buenas prácticas y notaciones Java propias de la organización y la implementación de las buenas prácticas para las tecnologías de información del framework ITIL. Así como las herramientas para el desarrollo de software: Spring, JavaServer Faces, Hibernate y el manejador de Bases de Datos SQL Server 2008.

Palabras clave: defectos, incidencias, gestión, ITIL, soporte.

INDICE

ACTA	ii
DEDICATORIA	iii
AGRADECIMIENTOS.....	iv
RESUMEN.....	vi
INDICE	vii
INDICE DE ILUSTRACIONES	x
INDICE DE TABLAS	xii
INDICE DE DIAGRAMAS	xiii
INTRODUCCIÓN.....	1
CAPÍTULO 1: PROBLEMA DE INVESTIGACIÓN	3
1. SITUACIÓN ACTUAL.....	3
2. PLANTEAMIENTO DEL PROBLEMA.....	4
3. JUSTIFICACION	5
4. OBJETIVO GENERAL	6
5. OBJETIVOS ESPECÍFICOS	6
6. SOLUCION PROPUESTA.....	6
7. ALCANCE	10
CAPÍTULO 2: MARCO CONCEPTUAL.....	11
1. DEFECTOS	11
1.1. DEFINICIÓN	11
1.2. TIPOS DE DEFECTOS.....	11
1.3. REGISTRO DE DATOS DE DEFECTOS.....	12
1.4. FLUJO DE DEFECTOS	12
2. INCIDENCIAS	13
3. ITIL FRAMEWORK.....	13

3.1.	TECNOLOGIAS DE INFORMACION.....	13
3.2.	¿QUE ES ITIL?	14
3.3.	CICLO DE VIDA DE LOS SERVICIOS DE TI	14
3.4.	FASE DE ESTRATEGIA DEL SERVICIO	16
3.5.	FASE DE DISEÑO DEL SERVICIO	16
3.6.	FASE DE TRANSICION DEL SERVICIO	17
3.7.	FASE DE OPERACIÓN DEL SERVICIO.....	18
3.8.	FASE DE MEJORA CONTINUA DEL SERVICIO	19
3.9.	ITIL Y LA GESTIÓN DE INCIDENCIAS	20
4.	GESTIÓN DE INCIDENCIAS	21
4.1.	DEFINICIÓN	21
4.2.	OBJETIVOS.....	21
4.3.	CARACTERÍSTICAS	22
4.4.	ROLES ITIL PARA LA GESTION DE INCIDENCIAS	24
4.5.	PROCESO	26
4.6.	BENEFICIOS	28
4.7.	GESTION DE INCIDENCIAS COMO REFERENCIA PARA LA GESTION DE DEFECTOS	28
5.	APLICACIONES WEB	29
5.1.	DEFINICIÓN	29
5.2.	PATRÓN DE ARQUITECTURA MODELO VISTA CONTROLADOR (MVC)	29
5.3.	ARQUITECTURA CLIENTE-SERVIDOR.....	32
6.	TECNOLOGÍAS Y HERRAMIENTAS	33
6.1.	SOFTWARE DE DESARROLLO SPRING FRAMEWORK	34
6.2.	SOFTWARE DE DESARROLLO JSF (JAVASERVER FACES).....	37
6.3.	SOFTWARE DE DESARROLLO ORM: HIBERNATE.....	39
6.4.	SOFTWARE DE DESARROLLO MICROSOFT SQL SERVER 2008	40
6.5.	SOFTWARE DE DESARROLLO JAVADOC	42
6.6.	SISTEMA DE REFERENCIA: JIRA	43
6.7.	SISTEMA DE REFERENCIA: MANTISBT.....	44

7. METODOLOGÍAS ORIENTADAS AL DESARROLLO DE SOFTWARE	47
7.1. METODOLOGÍA RACIONAL UNIFIED PROCESS (RUP).....	47
7.1.1. DEFINICIÓN	47
7.1.2. CARACTERÍSTICAS	48
7.1.3. FASES RUP	48
7.1.4. DISCIPLINAS DE DESARROLLO	50
7.1.5. ROLES RUP	50
7.1.6. ARTEFACTOS	51
7.1.7. RUP Y LAS MEJORES PRACTICAS	52
8. DEFINICIÓN DE ESTÁNDARES Y BUENAS PRÁCTICAS	53
8.1. CONVENCIONES	53
8.2. ARCHIVO DE CÓDIGO FUENTE	54
8.3. ENCABEZADOS DE ARCHIVOS DE CÓDIGO FUENTE.....	54
CAPÍTULO 3: MARCO APLICATIVO	56
1. CONTEXTO DEL DESARROLLO	56
2. PROYECTO	56
2.1. FASES DE RUP.....	57
2.1.1. FASE DE INICIO	57
2.1.2. FASE DE ELABORACION Y DISEÑO	105
2.1.3. FASE DE CONSTRUCCION	145
2.1.4. FASE DE TRANSICION	154
2.2. APLICACIÓN DE ITIL AL SISTEMA.....	156
3. CONCLUSIONES.....	160
BIBLIOGRAFIA.....	161
ANEXOS.....	165

INDICE DE ILUSTRACIONES

Figura 1. Arquitectura del Sistema Propuesto.....	7
Figura 2. Autenticación de Usuario.....	8
Figura 3. Mi Vista.....	8
Figura 4. Vista de Defectos.....	9
Figura 5. Detalles de Defecto.....	9
Figura 6. Ciclo de Vida de los Servicios de TI. (TCPSI, 2014).	15
Figura 7. Proceso de Escalado.....	23
Figura 8. Roles durante el proceso de Gestión de Incidentes.....	25
Figura 9. Proceso de Gestión de Incidentes.	26
Figura 10. Flujo del Modelo Vista Controlador.	31
Figura 11. Módulos de trabajo de Spring Framework.....	34
Figura 12. Proceso RUP.	49
Figura 13. Formato de Encabezado.....	55
Figura 14. Flujo del Sistema.	59
Figura 15. Modelo del Contexto del Negocio.	60
Figura 16. Prototipo Login SGD.....	61
Figura 17. Prototipo Cabecera de Aplicación con Menú.....	61
Figura 18. Prototipo Menú opción MiVista.....	62
Figura 19. Prototipo MiVista.....	62
Figura 20. Prototipo Menú opción Control de Defectos.....	63
Figura 21. Prototipo Control y Gestión de Defectos.	63
Figura 22. Prototipo Reportar Defectos.....	64
Figura 23. Prototipo Operaciones CRUD.	64
Figura 24. Prototipo Editar Defecto.	65
Figura 25. Prototipo Tabla Seguimientos de Defectos.	66
Figura 26. Prototipo Tabla Histórico de Defectos.	66
Figura 27. Prototipo Menú opción Reportes.....	67
Figura 28. Prototipo Menú opción Reportes Desplegable.	67
Figura 29. Prototipo Menú opción Reporte General.....	67
Figura 30. Prototipo Menú opción Cerrar Sesión.	68
Figura 31. Inicio de Sesión.....	145
Figura 32. Principal. Administrador.	146
Figura 33. Módulo Mi Cuenta.....	146
Figura 34. Módulo Proyectos.	146
Figura 35. Crear Nuevo Proyecto.....	147
Figura 36. Módulo Usuarios.....	147
Figura 37. Crear Nuevo Usuario.	147
Figura 38. Reportar Defecto.....	148
Figura 39. Principal. Usuario Funcional.....	148
Figura 40. Detalle del Defecto.....	149

Figura 41. Principal. Líder de Proyecto.	149
Figura 42. Asignación. Detalle del Defecto.	150
Figura 43. Principal. Usuario Analista.	150
Figura 44. Resolver. Detalle del Defecto.	151
Figura 45. Principal. Usuario Funcional.	151
Figura 46. Validación. Detalle del Defecto.	152
Figura 47. Detalle del Defecto. Histórico de Estatus.	152
Figura 48. Menú de opciones para muestra de reportes.	153
Figura 49. Reporte y Gráfico de defectos por estado.	153
Figura 50. Reporte y Gráfico de defectos con mayor tiempo en estado “Abierto”	154
Figura 51. Login Aplicación.	166
Figura 52. Pantalla principal.	167
Figura 53. Formulario Crear Defecto.	167
Figura 54. Opciones Control de Defectos	168
Figura 55. Tabla de Defectos Resueltos y Pendientes por Validar	168
Figura 56. Tabla de Defectos Asignados y Pendientes por Resolver	168
Figura 57. Detalle del Defecto.	169
Figura 58. Opciones de Reportes	¡Error! Marcador no definido.
Figura 59. Resultados Reportes por Estado de Defecto	170
Figura 60. Administración del Sistema	170
Figura 61. Gestión de Proyectos.	171
Figura 62. Gestión de Usuarios	171
Figura 63. Gestión de Categorías	172
Figura 64. Gestión de Motivos de Rechazo	172
Figura 66. Información de la Cuenta	172
Figura 67. Filtro de Proyectos.	173
Figura 68. Cerrar Sesión.	173

INDICE DE TABLAS

Tabla 1. Ejemplo de categorización de defectos	12
Tabla 2. Ejemplo Sistema de Prioridades	22
Tabla 3. Palabras reservadas JavaDoc.....	43
Tabla 4. Prioridades del sistema.....	157

INDICE DE DIAGRAMAS

Diagrama 1. Diagrama de Componentes.....	58
Diagrama 2. Casos de Uso Nivel 0.....	68
Diagrama 3. Casos de Uso Nivel 1.....	70
Diagrama 4. Casos de Usos Nivel 2.....	71
Diagrama 5 .Colaboración Sistema de Autenticación.....	105
Diagrama 6. Colaboración Registro de Defectos.....	106
Diagrama 7. Colaboración Consulta de Defectos.....	106
Diagrama 8. Colaboración Asignación de Defectos.....	107
Diagrama 9. Colaboración Intervención de Defectos.....	107
Diagrama 10. Diagrama de Estados de Defectos.....	¡Error! Marcador no definido.
Diagrama 11. Diagrama de Flujo del ciclo de vida del Defecto.....	¡Error! Marcador no definido.
Diagrama 12. Secuencia Autenticación de Usuario.....	110
Diagrama 13. Secuencia Crear Categoría.....	111
Diagrama 14. Secuencia Modificar Categoría.....	112
Diagrama 15. Secuencia Eliminar Categoría.....	113
Diagrama 16. Secuencia Crear Motivos Rechazo.....	114
Diagrama 17. Secuencia Modificar Motivos Rechazo.....	115
Diagrama 18. Secuencia Eliminar Motivos Rechazo.....	116
Diagrama 19. Secuencia Crear Proyectos.....	117
Diagrama 20. Secuencia Modificar Proyectos.....	118
Diagrama 21. Secuencia Eliminar Proyectos.....	119
Diagrama 22. Secuencia Crear Usuario.....	120
Diagrama 23. Secuencia Modificar Usuario.....	121
Diagrama 24. Secuencia Eliminar Usuario.....	122
Diagrama 25. Secuencia Reportar Defecto.....	123
Diagrama 26. Secuencia Modificar Defecto.....	124
Diagrama 27. Secuencia Asignar Defecto.....	125
Diagrama 28. Secuencia Resolver Defecto.....	126
Diagrama 29. Secuencia Re- Abrir Defecto.....	127
Diagrama 30. Secuencia Validar Defecto.....	128
Diagrama 31. Prototipo Modelo Vistas del Sistema.....	129
Diagrama 32. Diagrama de Estados Privilegios Administrador.....	130
Diagrama 33. Diagrama de Estados Privilegios Líder de Proyecto.....	131
Diagrama 34. Diagrama de Estados Privilegios Analista.....	132
Diagrama 35. Diagrama de Estados Privilegios Usuario_Funcional.....	133
Diagrama 36. Diagrama de Estados Privilegios Visualizador.....	133
Diagrama 37. Diagrama Modelo E-R.....	134
Diagrama 38. Modelo Relacional de Datos.....	135

INTRODUCCIÓN

En los últimos años, las organizaciones financieras han estado experimentando una gran presión competitiva que los obliga a revisar los métodos y herramientas utilizadas, y analizan si es necesario, realizar inversiones en tecnología para desarrollar nuevos instrumentos que mejoren la prestación de sus servicios. La globalización, que ha introducido a los mercados nuevos competidores, junto con la presión de nuevas normativas por parte del gobierno nacional, son factores que ejercen presión para bajar los costos y brindar, a la vez, mejores servicios.

Las organizaciones financieras al igual que otras empresas deben ir a la par de los avances tecnológicos, los cuales facilitan la comunicación y el ahorro de tiempo en sus operaciones. En este sentido, es sabido que el negocio bancario, por manejar grandes bases de datos, de clientes, cuentas, cotizaciones, transferencias, fondos y otros está ligado estrechamente con la información, por lo cual se hace propicio para liderar innovaciones constantes; pero también es cierto que las instituciones financieras son una de las instituciones más conservadoras, por lo que ellos deben ofrecer, a la vez que productos novedosos, seguridad a toda prueba en un negocio en el que la confianza lo es todo.

Con la aparición y posterior avance de las tecnologías de información y comunicación (TICs) se ha influido significativamente la operatividad de las empresas de todos los sectores de la economía mundial. Las organizaciones financieras no son ajenas al impacto que tienen las nuevas tecnologías, el ámbito bancario cambió de rostro y es uno de los que más han evolucionado sobre la manera de operar en un sistema financiero globalizado, la realización de diversas operaciones electrónicas son ejemplo de los avances que el ámbito bancario ha dado gracias a las TICs.

Esta constante evolución también compromete que todos los procesos que sustentan el negocio deban evolucionar; por esta razón la organización busca optimizar sus procesos internos para estar en la capacidad de proveer rápidas y eficientes respuestas a los requerimientos de productos y/o servicios que se presentan, además de promover el uso de marcos de referencia que describen un conjunto de mejores prácticas y recomendaciones para la administración de servicios de TI, por ejemplo la Information Technology Infrastructure Library (ITIL).

En la organización financiera, el área de Tecnología de la Información, encargada del desarrollo de software no cuenta con un sistema para la gestión de defectos que permita el registro y seguimiento del progreso de un defecto que un usuario identifique hasta que el mismo sea resuelto. Un defecto puede ser cualquier evento que pueda causar fallas en el uso de un sistema informático cuando éste se genera en fase de desarrollo del sistema en ambiente de calidad (Information Technology Standard, 2014). Éste puede ser reportado por el usuario, guardando toda la información relevante acerca del fallo y siguiendo un flujo de trabajo determinado que permita al usuario estar informado sobre el estado del defecto, la progresión del camino hacia su resolución, los responsables de resolver el problema y las acciones a utilizar para solventar el defecto.

En síntesis, es la finalidad del presente trabajo desarrollar una aplicación web que permita el registro, control y seguimiento riguroso de defectos que ocurren en un proyecto durante su fase de pruebas en ambiente Calidad hasta lograr su resolución de manera veraz y eficaz a través de reportes de resolución de defectos que genera el sistema gestor de defectos, cuya función es generar estadísticas referente a los defectos resueltos por cada usuario con su rol correspondiente, para así dar apoyo al área de Tecnología de la Información de la organización financiera y finalizar de manera exitosa la implantación de un proyecto en ambiente Producción.

El documento se encuentra dividido en tres capítulos:

Capítulo I. Problema de Investigación: En este capítulo se explica la situación actual, presenta el planteamiento del problema con la solución propuesta, los objetivos de investigación, la justificación y delimitación del Trabajo Especial de Grado (TEG).

Capítulo II. Marco Conceptual: En este capítulo se definen conceptos que fundamenten la investigación y el desarrollo del Sistema de Gestión de Defectos. También se estudian las tecnologías y herramientas a utilizar para el desarrollo del proyecto.

Capítulo III. Marco Aplicativo: En este capítulo se expone la metodología a seguir para el análisis, diseño, construcción e implementación del Sistema de Gestión de Defectos, además de una serie de estándares y buenas prácticas fomentadas por el equipo de desarrollo del área de tecnología, que deben ser tomados en cuenta al momento de desarrollar o realizar mantenimiento en aplicaciones de la organización financiera.

Para finalizar se presentan las conclusiones y referencias bibliográficas que se utilizaron en el desarrollo del Trabajo Especial de Grado.

CAPÍTULO 1: PROBLEMA DE INVESTIGACIÓN

En este capítulo se detalla la situación actual de la organización financiera con respecto a la forma como maneja los defectos reportadas durante el desarrollo de sus proyectos. Esta situación actual da origen al problema planteado el cual se detallara a continuación, y se propone también una solución al mismo, el alcance, la justificación y los objetivos tanto generales como específicos.

1. SITUACIÓN ACTUAL

La visión de la entidad financiera está orientada a "ser una organización financiera líder, útil y responsable con la sociedad, cercana al cliente y enfocada en el negocio bancario, que garantice la plena satisfacción de sus clientes, colaboradores y accionistas".

La Organización Financiera de Caracas, orientada hacia la mejora en productos y servicios, realizó en el año 2012 la actualización de la infraestructura tecnológica de sus sistemas principales, es decir la plataforma interna de la organización, el core bancario. Se seleccionó una plataforma de última generación como IBM Power 7; dicha arquitectura permite ser más competitivos y ágiles en la prestación de servicios, acortando aún más los tiempos de respuesta. La ventana de tiempo de interrupción en caso de mantenimientos se reduce, lo cual significa que la disponibilidad de servicios mejora sustancialmente.

Este proyecto de innovación tecnológica que llevó a cabo la organización financiera facilita que se asuma una mayor demanda de clientes, así como permitir desarrollar más y mejores servicios. A la vez se le da continuidad al negocio, cumpliendo de manera cada vez más eficiente con las regulaciones establecidas por el órgano regulador que es la Superintendencia de Bancos y Otras Instituciones Financieras (Sudeban), satisfaciendo las nuevas exigencias gubernamentales.

El área de Tecnología de la Información como uno de los componentes principales en la ejecución de los proyectos, tiene a su disposición el hardware necesario, así como también el software y diversas aplicaciones para una eficiente gestión.

En la Organización Financiera de Caracas existe un sistema para el soporte a usuarios llamada: ServiDesk (Producto implantado por IT Business Solutions <http://www.itbscorp.com>), dicho sistema permite la creación de solicitudes, tales como soporte técnico, redes, hardware, entre otras; pero esta solución no está orientada a la gestión de proyectos, la cual es una gestión necesaria para el área de tecnología de información de la organización.

Además esta solución no permite el seguimiento y control de defectos típicos (registro, asignación, monitoreo, resolución de defectos, entre otras funcionalidades) al momento de certificar y hacer pruebas de los proyectos que el área de tecnología de la información realiza para mejorar la calidad de los servicios provistos por la organización financiera.

2. PLANTEAMIENTO DEL PROBLEMA

La misión del área de Tecnología de la Información es proveer y mantener soluciones integrales sobre la plataforma tecnológica de la organización financiera que respondan a las necesidades del negocio de manera ágil, innovadora, eficaz y basada en buenas prácticas de gestión de TI, con el mejor talento humano.

El sistema Service Desk provee un punto de contacto entre los usuarios de la organización y el equipo de soporte, permitiendo registrar y controlar sus actividades, desde el momento en que se recibe la llamada en el departamento de soporte o se levanta un ticket vía WEB, hasta el cierre del evento.

Aunque el sistema Service Desk gestiona las solicitudes, el mismo no permite el seguimiento y control de defectos de proyectos, por lo que no se puede almacenar el registro y seguimiento de un defecto.

En la actualidad, un defecto puede originarse en la ejecución de las pruebas, en ambiente de desarrollo o calidad, tanto del usuario funcional como las que puede efectuar un desarrollador, las mismas son registradas en una hoja Excel para su seguimiento y posteriormente se envía una notificación vía correo electrónico para su resolución dirigido al usuario responsable de resolver dicho defecto. Estos defectos tienen un estatus y una persona responsable; y no se registran los detalles ni las acciones utilizadas para solventar el fallo.

En una oportunidad, se utilizó una herramienta de software libre: MANTIS (<http://www.mantisbt.org/>), para realizar la gestión de los defectos encontrados en los proyectos, pero dada las normas del área de Tecnología de la Información, no se usa software libre, por lo tanto dicha herramienta sólo servirá como referencia para una propuesta de solución.

3. JUSTIFICACION

Hoy en día, los sistemas informáticos son cada vez más complejos y las empresas dependen mucho más de servicios tecnológicos por lo que un mal funcionamiento o la interrupción de servicios de éstos pueden llegar a tener importantes consecuencias para el logro de los objetivos de las empresas.

Por lo tanto, es crucial para las empresas que brindan estos servicios ser capaces de solucionar esos defectos en sus servicios en etapa de desarrollo y de hacerlo en el menor tiempo posible. Esto es debido a que dichos defectos afectan la productividad del proyecto, pues deben resolverse rápidamente para no afectar el tiempo y costo en la vida del proyecto. La calidad que brinden a la hora de solucionar defectos será un hecho determinante para la mejora de los tiempos de salida de sus productos a un ambiente de Producción.

Con todo lo mencionado anteriormente, se desprende la necesidad de que cualquier negocio que provea o desarrolle un servicio tecnológico, tanto para terceros como para uso propio, debe poseer un sistema que se encargue de gestionar defectos al menos durante la etapa de desarrollo. Además, este sistema debe disponer de una asignación automática de los problemas reportados a los desarrolladores disponibles en ese momento, valorando la carga de trabajo de cada uno de ellos, con el objetivo de reducir el tiempo dedicado a la gestión y planificación de los defectos según recursos.

Teniendo en cuenta todo lo anterior, se requiere implementar un sistema de gestión de defectos para llevar el seguimiento de todos los defectos de los proyectos de la organización financiera en los ambientes de desarrollo y calidad, y que sea capaz de adaptar al conjunto de defectos correspondiente a cada proyecto de la organización financiera en etapa de desarrollo o pruebas al flujo de control de estados de defectos que posee el sistema.

4. OBJETIVO GENERAL

Desarrollar una aplicación web para la gestión de defectos, permitiendo el registro y seguimiento de las mismas; logrando así un control veraz y eficaz en la resolución de cada defecto reportado en un proyecto a través de las estadísticas generadas por los reportes del sistema y apoyando la misión del área de Tecnología de la Información.

5. OBJETIVOS ESPECÍFICOS

Entre los objetivos específicos se encuentran:

- Analizar el proceso de Gestión de defectos, sus características y las recomendaciones en esta materia por la ITIL.
- Evaluar los procesos actuales e identificar todos los campos que deban registrarse en un defecto, tanto los obligatorios como los opcionales.
- Definir el flujo de trabajo y los roles en la asignación de los defectos, los estados que puedan tener y acciones posibles a ejecutar en los mismos
- Implementar el proceso de invocación al Servicio de Autenticación de la organización financiera, con el cual se permita el ingreso al sistema y se obtendrá el menú de opciones que puede utilizar un usuario.
- Desarrollar el formulario de Registro de Defectos y el módulo de Reportes que muestren estadísticas y gráficos de los defectos, en cuanto a sus estados, resolución y características como sus relaciones con los usuarios y responsables.
- Revisión de estándares y buenas prácticas durante el desarrollo de la aplicación, así como la elaboración de la documentación técnica de la aplicación.

6. SOLUCION PROPUESTA

Como solución, se plantea el desarrollo de un sistema para el seguimiento y control de defectos, un Sistema de Gestión de Defectos para el área de Tecnología de la Información, la arquitectura del sistema se encuentra ilustrada en la siguiente Figura 1.

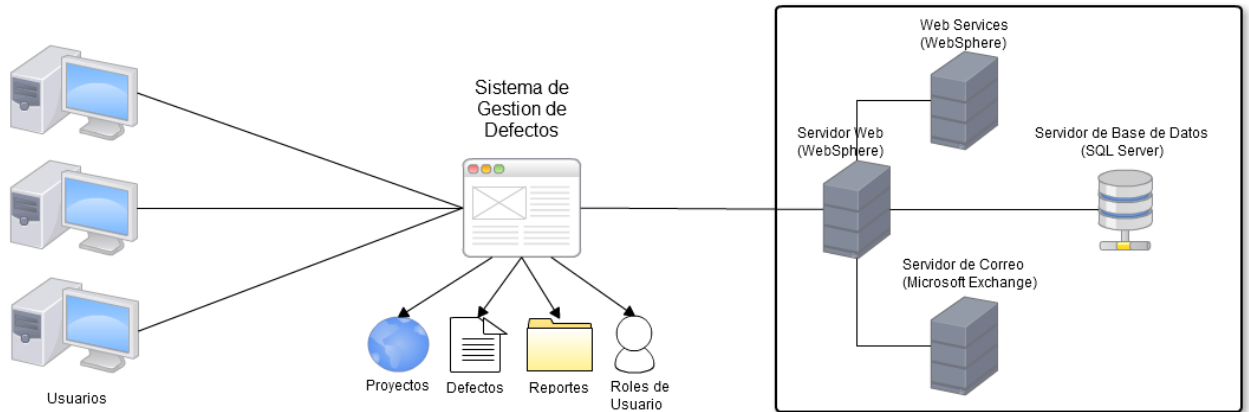


Figura 1. Arquitectura del Sistema Propuesto.

Como se puede observar en la figura 1 varios clientes se pueden conectar al sistema de gestión de defectos. Además el sistema permite la gestión de defectos relacionados a sus respectivos proyectos con ciertos privilegios de usuario a través de roles, genera estadísticas a través de los reportes y posee instancias a servicios web que son consumidos por el sistema como es el caso de autenticación de usuarios (WS BEASA) y las notificaciones por correo (WS BECA). Adicionalmente consta de una base de datos en el sistema manejador de base de datos SQL Server 2008 R2.

Esta aplicación web permite llevar un control eficaz de cada proyecto y mantener el historial de cada defecto en el tiempo; asimismo, dicha aplicación web facilita la verificación, en cualquier momento y lugar, el estado de los defectos, por ejemplo "Asignado" cuando el defecto fue asignado a un desarrollador, "No asignado" para el caso contrario, "Resuelto" cuando el usuario responsable solventa el defecto reportado, "Re-Abierto" cuando el usuario encargado de verificar y validar la resolución del defecto no lo aprueba, en caso de ser aprobado dicha resolución es definida en estado "Cerrado".

El lenguaje principal de desarrollo web para éste proyecto es JavaServer Faces (JSF) que posee un patrón MVC flexible debido a su fácil integración con otros frameworks existentes del mismo entorno Java tales como Spring Core que es un sub-módulo del framework Spring, Hibernate para establecer el ambiente de base de datos, librería PrimeFaces para el Front-End, entre otras tecnologías.

Para ingresar al sitio, los usuarios deben introducir su número de carnet, contraseña de red en el formulario de autenticación (Figura 2) y al pulsar el botón “Ingresar” se autenticarán versus el Directorio Activo, conocido también como Protocolo Ligero/Simplificado de Acceso a Directorios (LDAP), es decir, la gestión de contraseñas y permisología es controlada por otra aplicación, todo lo anterior por estándares de la organización financiera.

Figura 2. Autenticación de Usuario

Luego del ingreso exitoso se despliega la pantalla principal con varias opciones en el menú superior, la opción por defecto es “Mi Vista”, esta opción muestra resumidamente los defectos que han sido modificados o actualizados de estado en el último tiempo para un proyecto; se clasifican en las siguientes secciones: No Asignadas, Reportadas por Mí, Resueltas y Modificadas Recientemente (Figura 3).

Asunto_defecto	Descripcion_defecto	Fecha_estimada	Nombre_categoria	Estado	Detalle
Error en titulo	Error en el titulo, muestra datos privados del usuario logueado en lugar del perfil personal	2015-03-31 23:24:17.417	Seguridad	ABIERTO	

En total son 1 defecto(s) sin asignar por mí.

Figura 3. Mi Vista

La segunda opción del menú superior es “Vista de Defectos”, aquí se despliega un listado de todos los defectos que han sido reportados en el proyecto elegido; se pueden ordenar por: Estatus, Severidad, Asignado a, Categoría y Fecha de Creación, esta página contiene un conjunto de visualización de filtros en la parte superior (Figura 4).

Asunto_defecto	Descripcion_defecto	Fecha_estimada	Nombre_categoria	Estado	Detalle
Error en titulo	Error en el título, muestra datos privados del usuario logueado en lugar del perfil personal	2015-03-31 23:24:17.417	Seguridad	ABIERTO	
Error de login	Error al ingresar username en el campo login	2015-04-04 20:01:33.67	Seguridad	CERRADO	

En total son 2 defecto(s).

Figura 4. Vista de Defectos.

Cuando se haga click en el número de la incidencia se muestra un detalle de la misma, se presentan los campos: Id de la Incidencia, Proyecto, Categoría, Estatus, Asignado a, etc. (Figura 5).

Figura 5. Detalles de Defecto.

Esta solución permite la creación de un proyecto y la carga de defectos que sean reportados por los usuarios, se puede asignar la resolución a un responsable (definiendo un flujo de trabajo, donde las notificaciones se harán vía correo electrónico); dichos defectos tienen características tales como: Estatus, Fecha Creación, Usuario Responsable, etc. También es posible la generación de reportes en base a toda la información recolectada (los tiempos de respuesta, defectos por usuario, defectos por desarrollador, entre otros).

7. ALCANCE

Los sistemas de Gestión de Defectos abarcan muchos detalles, entre las modalidades y funcionamiento que contempla la solución a implementar están:

- Especificar los datos necesarios para el registro de un defecto.
- Definir un flujo de trabajo del defecto.
- Establecimiento de roles de usuario y estatus para los defectos.
- Script SQL para la creación de la base de datos.
- Módulo para la administración de usuarios y defectos.
- Módulo para la consulta de un defecto por parte de un cliente.
- Generación de reportes de defectos y alertas vía correo electrónico.
- El proyecto cubrirá las etapas de diseño lógico y físico hasta su Implementación en fase de transición.
- Indicar conformidad de la evaluación por parte del evaluador.
- Documentación y Manual de usuario.

CAPÍTULO 2: MARCO CONCEPTUAL

1. DEFECTOS

1.1. DEFINICIÓN

Cabe destacar que para definir el término defectos es importante aclarar en qué fase del ciclo de vida de un sistema de información éstos se generan. Tomando como referencia la metodología de desarrollo RUP, dicha metodología posee las siguientes fases: Inicio, Elaboración y Diseño, Construcción y Transición. Los defectos se ubican en la etapa de transición (en Certificación de usuarios en Calidad y antes del pase a Producción, es decir transición a operaciones) a diferencia de las incidencias que se ubican en el ambiente de Producción.

Aclarado esto, Defecto es la imperfección en un componente o sistema que puede causar que el componente o sistema falle en desempeñar las funciones requeridas, por ejemplo una sentencia o una definición de datos incorrectos. Si se localiza un defecto durante una ejecución puede causar un fallo en el componente o sistema. (GlobeTesting.,2011)

En definitiva un defecto, reduce la capacidad de los programas para cumplir completa y efectivamente las necesidades de los usuarios. Es un objeto que se puede identificar, describir y contabilizar. Para mejorar la calidad del programa, es esencial que los desarrolladores del Área de Tecnologías de la Información de la organización aprendan a gestionar todos los defectos que presentan en sus proyectos. Para ello, es importante encontrar, clasificar y corregir los defectos de software ya que los costes para una posterior corrección serian elevados, es importante minimizarlos. Además se debe aprender de los defectos introducidos, identificar los errores que los causan y aprender a no repetir el fallo en el futuro.

1.2. TIPOS DE DEFECTOS

Para poder analizar los defectos, estos se deben categorizar con la finalidad de estudiar las categorías más problemáticas y procurar su prevención y eliminación. Según Godoy, Gómez y Rubio (2004), un ejemplo de esta categorización se muestra en la siguiente tabla:

Nombre del tipo de defecto	Descripción
Documentación	Comentarios, mensajes
Sintaxis	Ortografía, puntuación, erratas, formato de las instrucciones.
Construir Paquetes	Gestión del cambio, librerías, control de versión
Asignación	Declaración, nombres duplicados, ámbito, límites
Interfaz	Llamadas a procedimientos y referencias E/S, formatos de usuario.
Chequeo	Mensajes de error, chequeos inadecuados.
Datos	Estructura, contenido
Función	Lógica, punteros, bucles, recursión, defectos de función
Sistema	Configuración, pruebas y otros problemas que soporta el sistema.

Tabla 1. Ejemplo de categorización de defectos.

1.3. REGISTRO DE DATOS DE DEFECTOS

Los datos de cada defecto encontrado y corregido se deben anotar de forma separada y completa. De acuerdo a Godoy, Gómez y Rubio (2004), los campos sugeridos para el registro de defectos son:

Fecha: La que se encontró el defecto.

Número: Número de orden secuencial de error encontrado.

Tipo: Según lista de defectos.

Fase localización: Fase en la que se introduce el defecto (diseño, codificación, compilación, entre otros).

Fase Completado: Fase en la que se encontró y corrigió el defecto.

Tiempo de corrección: Medición o estimación del tiempo necesario para encontrar y corregir el defecto.

Descripción: Breve descripción de forma clara del defecto.

1.4. FLUJO DE DEFECTOS

Según Godoy, Gómez y Rubio (2004), un flujo de defectos es indispensable para el buen funcionamiento de un proyecto. Con ello se logra que el defecto siga los flujos adecuados y se pueda organizar de manera más sencilla la información relacionada a los defectos.

Un flujo debe tener al menos estos 4 estados básicos: “Abierto”, “En Desarrollo”, “Terminado” y “Cerrado”.

- Abierto: El defecto está abierto y listo para validar.
- En Desarrollo: El desarrollador realiza la solución del bug.
- Terminado Desarrollo: El tester procede a validarlo.
- Cerrado: El tester ha dado su OK. El defecto se cierra y queda como historial.

2. INCIDENCIAS

Van Haren Publishing (2008) define ‘Incidencia’ como: “Cualquier evento que no forma parte de la operación estándar de un servicio y que provoca o puede provocar, la interrupción o reducción en la calidad del servicio”.

En este contexto “Incidente” incluye no solo errores de hardware y software, sino también Peticiones de Servicio porque ambas se gestionan de manera similar las cuales se generan cuando el servicio se encuentra en el ambiente de producción.

Durante las pruebas del sistema, cuando un tester ejecuta los casos de prueba puede notar que los resultados de esas pruebas no coinciden con los resultados esperados. La variación en los resultados esperados y los reales se conoce como defectos. Diferentes organizaciones tienen diferentes nombres para describir esta variación, comúnmente a los defectos también se les conocen como errores, problemas, o incidencias.

3. ITIL FRAMEWORK

A continuación se explican conceptos relacionados con Tecnología de la Información (TI), ITIL, Ciclo de Vida de los servicios TI, entre otros conceptos, para centrarnos más aún en el contexto de los Servicios de TI y su vital importancia referente a la gestión de defectos que se ubica dentro de la fase de Operación de Servicios en el cuál se basa el desarrollo del sistema de gestión de defectos.

3.1. TECNOLOGIAS DE INFORMACION

3.1.1. DEFINICION

Las tecnologías de la información se definen como el conjunto de herramientas, procesos y metodologías (tales como la codificación/programación, comunicaciones de datos, conversión de datos,

almacenamiento y recuperación de información, análisis y diseño de sistemas, control de sistemas) encargadas de almacenar, recoger, procesar y transmitir la información. (WebFinance, 2015)

3.1.2. IMPORTANCIA DE LAS TI EN LAS EMPRESAS

Para (DeGerencia, 2014), la importancia de las TI en las empresas, el departamento o equipo de trabajo que ejerce las funciones de TI dentro de una organización, se encarga de estudiar, diseñar, desarrollar, implementar y administrar los sistemas de información utilizados para el manejo de datos e información de toda la organización. Estos sistemas, a su vez, comprenden aplicaciones o software, y equipos o hardware.

Llevar a cabo las tareas de la organización apoyándose en la Tecnología de información, generalmente redundante en un procesamiento más rápido y confiable de sus datos. La información resultante tiene mayor movilidad y accesibilidad, y cuenta con mayor integridad, que cuando se procesa en forma manual. Igualmente, las computadoras relevan a los empleados de numerosas actividades repetitivas y aburridas, permitiéndoles aprovechar mejor su tiempo en actividades que agregan más valor.

3.2. ¿QUE ES ITIL?

Biblioteca de Infraestructura de Tecnología de la Información (ITIL - Information Technology Infrastructure Library).

TRAKER SUITE (2013) define ITIL de la siguiente manera:

“La Biblioteca de Infraestructura de Tecnología de la Información (ITIL) es un framework de enfoques de mejores prácticas que pretenden facilitar la entrega de tecnología de información de alta calidad (TI). ITIL resume un extenso conjunto de procedimientos de gestión que están destinadas a apoyar a las empresas en la consecución de la calidad y la relación calidad-precio en las operaciones de TI. Estos procedimientos son de proveedores independientes y se han desarrollado para proporcionar orientación sobre la anchura de la infraestructura de TI, desarrollo y operaciones.”

3.3. CICLO DE VIDA DE LOS SERVICIOS DE TI

Según (Osiatis, 2011) ITIL estructura la gestión de los servicios TI sobre el concepto de Ciclo de Vida de los Servicios.

Este enfoque tiene como objetivo ofrecer una visión global de la vida de un servicio desde su diseño, donde se encuentra la gestión de defectos, hasta su eventual abandono sin por ello ignorar los detalles de todos los procesos y funciones involucrados en la eficiente prestación del mismo.

Cabe destacar que la gestión de incidencias está incluida dentro de fase de Servicio de Operación del ciclo.

En la figura 6 se puede observar el Ciclo de Vida del Servicio, consta de cinco fases que se retroalimentan entre ellas de una manera cíclica:



Figura 6. Ciclo de Vida de los Servicios de TI. (TCPSI, 2014).

- **Estrategia del Servicio:** cuyo propósito es definir qué servicios se prestarán, a qué clientes y en qué mercados.
- **Diseño del Servicio:** responsable de desarrollar nuevos servicios o modificar los ya existentes, asegurando que cumplen los requisitos de los clientes y se adecuan a la estrategia predefinida.
- **Transición del Servicio:** encargada de la puesta en operación de los servicios previamente diseñados.
- **Operación del Servicio:** responsables de todas las tareas operativas y de mantenimiento del servicio, incluida la atención al cliente.
- **Mejora Continua del Servicio:** a partir de los datos y experiencia acumulados propone mecanismos de mejora del servicio.

3.4. FASE DE ESTRATEGIA DEL SERVICIO

La fase de Estrategia del Servicio es el eje que permite que las fases de Diseño, Transición y Operación del servicio se ajusten a las políticas y visión estratégica del negocio.

Para conseguir este objetivo es imprescindible determinar en primera instancia qué servicios deben ser prestados y por qué han de ser prestados desde la perspectiva del cliente y el mercado.

Una correcta Estrategia del Servicio debe:

- Servir de guía a la hora de establecer y priorizar objetivos y oportunidades.
- Conocer el mercado y los servicios de la competencia.
- Armonizar la oferta con la demanda de servicios.
- Proponer servicios diferenciados que aporten valor añadido al cliente.
- Gestionar los recursos y capacidades necesarios para prestar los servicios ofrecidos teniendo en cuenta los costes y riesgos asociados.
- Alinear los servicios ofrecidos con la estrategia de negocio.
- Elaborar planes que permitan un crecimiento sostenible.
- Crear casos de negocio para justificar inversiones estratégicas.

Los procesos asociados directamente a la fase de Estrategia son:

- **Gestión Financiera:** responsable de garantizar la prestación de servicios con unos costes controlados y una correcta relación calidad-precio.
- **Gestión del Portafolio de Servicios:** responsable de la inversión en servicios nuevos y actualizados que ofrezcan el máximo valor al cliente minimizando a su vez los riesgos y costes asociados.
- **Gestión de la Demanda:** responsable de la armonización de la oferta de los servicios ofrecidos con las demandas del mercado.

3.5. FASE DE DISEÑO DEL SERVICIO

La principal misión de la fase de Diseño del Servicio es la de diseñar nuevos servicios o modificar los ya existentes para su incorporación al catálogo de servicios y su paso al entorno de producción.

El Diseño del Servicio debe seguir las directrices establecidas en la fase de Estrategia y debe a su vez colaborar con ella para que los servicios diseñados:

- Se adapten a las necesidades del mercado.
- Sean eficientes en costes y rentables.
- Cumplan los estándares de calidad adoptados.
- Aporten valor a clientes y usuarios.

Las funciones y procesos asociados directamente a la **fase de Diseño** son:

- **Gestión del Catálogo de Servicios:** responsable de crear y mantener un catálogo de servicios de la organización TI que incluya toda la información relevante: gestores, estatus, proveedores, etcétera.
- **Gestión de Niveles de Servicio:** responsable de acordar y garantizar los niveles de calidad de los servicios TI prestados.
- **Gestión de la Capacidad:** responsable de garantizar que la organización TI dispone de la capacidad suficiente para prestar los servicios acordados.
- **Gestión de la Disponibilidad:** responsable de garantizar que se cumplen los niveles de disponibilidad acordados en los SLA.
- **Gestión de la Continuidad de los Servicios TI:** responsable de establecer planes de contingencia que aseguren la continuidad del servicio en un tiempo predeterminado con el menor impacto posible en los servicios de carácter crítico.
- **Gestión de la Seguridad de la Información:** responsable de establecer las políticas de integridad, confidencialidad y disponibilidad de la información.
- **Gestión de Proveedores:** responsable de la relación con los proveedores y el cumplimiento de los UCs.

3.6. FASE DE TRANSICION DEL SERVICIO

La misión de la fase de Transición del Servicio es hacer que los productos y servicios definidos en la fase de Diseño del Servicio se integren en el entorno de producción y sean accesibles a los clientes y usuarios autorizados.

Sus principales objetivos se resumen en:

- Supervisar y dar soporte a todo el proceso de cambio del nuevo (o modificado) servicio.
- Garantizar que los nuevos servicios cumplen los requisitos y estándares de calidad estipulados en las fases de Estrategia y la de Diseño.
- Minimizar los riesgos intrínsecos asociados al cambio reduciendo el posible impacto sobre los servicios ya existentes.

- Mejorar la satisfacción del cliente respecto a los servicios prestados.
- Comunicar el cambio a todos los agentes implicados.

Las principales funciones y procesos asociados directamente a la Fase de Transición del Servicio son:

- **Planificación y soporte a la Transición:** responsable de planificar y coordinar todo el proceso de transición asociado a la creación o modificación de los servicios TI.
- **Gestión de Cambios:** responsable de supervisar y aprobar la introducción o modificación de los servicios prestados garantizando que todo el proceso ha sido convenientemente planificado, evaluado, probado, implementado y documentado.
- **Gestión de la Configuración y Activos del Servicio:** responsable del registro y gestión de los elementos de configuración (CIs) y activos del servicio. Este proceso da soporte a prácticamente todos los aspectos de la Gestión del Servicio
- **Gestión de Entregas y Despliegues:** Responsable de desarrollar, probar e implementar las nuevas versiones de los servicios según las directrices marcadas en la fase de Diseño del Servicio.
- **Validación y pruebas:** responsable de garantizar que los servicios cumplen los requisitos preestablecidos antes de su paso al entorno de producción.
- **Evaluación:** responsable de evaluar la calidad general de los servicios, su rentabilidad, su utilización, la percepción de sus usuarios, etcétera
- **Gestión del Conocimiento:** gestiona toda la información relevante a la prestación de los servicios asegurando que esté disponible para los agentes implicados en su concepción, diseño, desarrollo, implementación y operación.

3.7. FASE DE OPERACIÓN DEL SERVICIO

La fase de operación del Servicio es aquella donde se realizan todas las actividades necesarias para la prestación y el soporte de los servicios. Asimismo, es la fase que principalmente nutre de información a la fase de Mejora Continua del Servicio.

Los procesos asociados a la Operación del Servicio son los siguientes:

- **Gestión de Eventos:** monitoriza todos los eventos que ocurren en la infraestructura del servicio y alerta de situaciones que pudieran llegar a suponer una incidencia.
- **Gestión de Incidencias:** restaura servicios interrumpidos o degradados lo antes posible, de forma que se minimice el impacto sobre el negocio.
- **Gestión de Peticiones de Servicio:** proporciona información relativa al catálogo de servicios a los usuarios y un canal para que dichos usuarios puedan solicitar servicios predefinidos y pre aprobados.
- **Gestión de Problemas:** diagnostica la causa raíz de las incidencias ocurridas, proporciona soluciones a la gestión de incidencias para restaurar los servicios lo antes posible y proactivamente ejecuta acciones para detectar y prevenir futuras incidencias.
- **Gestión de Accesos:** concede a los Usuarios autorizados el derecho a utilizar un Servicio e impide el acceso a Usuarios no autorizados

Las funciones asociadas a la fase de Operación del Servicio son las siguientes:

- **Service Desk (Centro de Servicio):** proporciona un punto de comunicación entre los usuarios y el servicio y un punto de coordinación entre grupos y procesos del servicio.
- **Gestión de Operaciones TI:** ejecuta las operaciones diarias necesarias para que la infraestructura del servicio funcione.
- **Gestión Técnica:** proporciona habilidades técnicas detalladas y los recursos necesarios para soportar la operación de la infraestructura del servicio.
- **Gestión de Aplicaciones:** es la responsable de la gestión del ciclo de vida de las aplicaciones TI.

3.8. FASE DE MEJORA CONTINUA DEL SERVICIO

Los principales objetivos de la fase de Mejora Continua del servicio se resumen en recomendar mejoras para todos los procesos y actividades involucrados en la gestión y prestación de los servicios TI, así como monitorizar y analizar los parámetros de seguimiento de Niveles de Servicio y contrastarlos con los SLAs en vigor y el dar soporte a la fase de estrategia y diseño para la definición de nuevos servicios y procesos/ actividades asociados a los mismos.

Los resultados de esta fase del ciclo de vida han de verse reflejados en Planes de Mejora del Servicio que incorporen toda la información necesaria para:

- Mejorar la calidad de los servicios prestados.
- Incorporar nuevos servicios que se adapten mejor a los requisitos de los clientes y el mercado.
- Mejorar y hacer más eficientes los procesos internos de la organización TI.

Los principales procesos asociados directamente a la **fase de Mejora del Servicio** son:

- **Proceso de Mejora:** este es un proceso que consta de 7 pasos que describen como se deben medir la calidad y rendimiento de los procesos para generar los informes adecuados que permitan la creación de un Plan de Mejora del Servicio (SIP).
- **Informes de Servicios TI:** es el responsable de la generación de los informes que permitan evaluar los servicios ofrecidos y los resultados de las mejoras propuestas.

3.9. ITIL Y LA GESTIÓN DE INCIDENCIAS

La gestión de incidencias es un proceso ITIL enmarcado en la fase de “Operación del Servicio” que se encarga de gestionar las incidencias del servicio. Las incidencias pueden incluir fallos o defectos reportados por los usuarios, el equipo del servicio o por alguna herramienta de monitorización de eventos.

La gestión de incidencias tiene como objetivo resolver, de la manera más rápida y eficaz posible, cualquier incidente que cause una interrupción en el servicio. Esta no debe confundirse con la Gestión de Problemas, pues a diferencia de esta última, no se preocupa de encontrar y analizar las causas subyacentes a un determinado incidente sino exclusivamente a restaurar el servicio.

Por otro lado, también es importante diferenciar la Gestión de Incidencias de la Gestión de Peticiones, que se ocupa de las diversas solicitudes que los usuarios plantean para mejorar el servicio, no cuando éste falla.

Asimismo, la función de **Centro de Servicio** (Service Desk) es una parte muy importante en el proceso de Gestión de Incidentes, ya que debe ser el único punto de contacto para los usuarios en el día a día del servicio y las tareas que lleva a cabo el Centro de Servicio son muy importantes para la resolución de las incidencias, tales como:

- Registrar, categorizar y priorizar las peticiones abiertas por los clientes.

- Proporcionar una primera línea de soporte, realizando un primer diagnóstico y resolviendo peticiones.
- Asignar las peticiones que no pueda resolver.
- Monitorizar la resolución de las peticiones, escalando aquellas para las que exista riesgo de incumplir el acuerdo de nivel de servicio.
- Mantener informados a los clientes del estado de sus peticiones.
- Cerrar las peticiones resueltas, previa validación con los usuarios.
- Medir el nivel de satisfacción de los usuarios.

4. GESTIÓN DE INCIDENCIAS

4.1. DEFINICIÓN

Es el proceso que se enfoca en cubrir todo tipo de incidencias encontradas en el ambiente de producción, ya sean fallos o defectos, consultas planteadas por usuarios o por el personal técnico, resolver el incidente y restaurar la provisión del servicio rápidamente. Los incidentes son registrados, y la calidad de los registros determina la efectividad para otros procesos (Iglesias B., 2010).

4.2. OBJETIVOS

El objetivo principal de la Gestión de Incidentes según Iglesias B. (2010) es restablecer el servicio, como lo define el Acuerdo de Nivel de Servicio - Service Level Agreement (SLA), lo más pronto que sea posible, con el menor impacto posible sobre la actividad del negocio de la organización y del usuario.

Entre otros objetivos de la Gestión de Incidentes se encuentran:

- Detectar cualquiera alteración en los servicios TI.
- Registrar y clasificar estas alteraciones para medir y evaluar el proceso.
- Asignar el personal encargado de restaurar el servicio según se define en el SLA correspondiente.
- Restablecimiento del servicio acordado lo antes posible y con el mínimo impacto al negocio.
- Minimizar el impacto negativo de un incidente en el negocio, garantizando el más alto nivel de calidad y disponibilidad del servicio.
- Identificar mejoras del servicio proactivamente.

- Minimizar el riesgo de incidentes perdidos.
- Recolección de información sobre la gestión.

OSIATIS S.A. (2009) concluye que una incorrecta **Gestión de Incidencias** puede acarrear efectos adversos tales como:

- Reducción de los niveles de servicios.
- Se dilapidan valiosos recursos: demasiada gente o gente del nivel inadecuado trabajando concurrentemente en la resolución del incidente.
- Se pierde valiosa información sobre las causas y efectos de los incidentes para futuras reestructuraciones y evoluciones.
- Se crean clientes y usuarios insatisfechos por la mala y/o lenta gestión de sus incidentes.

4.3. CARACTERÍSTICAS

4.3.1. Impacto, Urgencia y Prioridad

Cuando se atienden muchos incidentes al mismo tiempo, se deben establecer prioridades para la resolución de las mismas.

Para hacer una evaluación subjetiva se deben discutir con el usuario, los siguientes criterios:

- Impacto del Incidente: determina la importancia del incidente dependiendo de cómo éste afecta a los procesos de negocio y/o del número de usuarios afectados.
- Urgencia del Incidente: la demora aceptable para el usuario o el proceso de negocio, para resolver un incidente.

En la Tabla 2 se muestra un ejemplo de un sistema de prioridades.

		I M P A C T O		
		Alto	Medio	Bajo
U R G E N C I A	Prioridad tiempo de resolución			
	Alto	Critico < 1 hora	Alto < 8 horas	Medio < 24 horas
	Medio	Alto < 8 horas	Medio < 24 horas	Bajo < 48 horas
Bajo	Medio < 24 horas	Bajo < 48 horas	Planificado planeado	

Tabla 2. Ejemplo Sistema de Prioridades

La prioridad se determina sobre la base de la urgencia y el impacto. Para cada prioridad se define un número de personas y cierta cantidad de recursos. Para incidentes con la misma prioridad, el esfuerzo requerido para resolverlo puede determinar el orden. Ej.: Un incidente con poco impacto llevará menos tiempo de resolverlo ya que requiere poco esfuerzo y se puede resolver antes que un incidente con un impacto mayor el cual requiere más esfuerzo.

4.3.2. Escalado

OSIATIS S.A. (2009) testifica que es frecuente que en la Base de Lecciones Aprendidas no se encuentre la resolución en primera instancia de un incidente y para ello deba recurrir a un especialista o a algún superior que pueda tomar decisiones que se escapen de su responsabilidad. A este proceso se le denomina escalado.

Básicamente hay dos tipos diferentes de escalado:

- **Escalado funcional:** Se requiere el apoyo de un especialista de más alto nivel para resolver el problema.
- **Escalado jerárquico:** Debemos acudir a un responsable de mayor autoridad para tomar decisiones que se escapen de las atribuciones asignadas a ese nivel, como, por ejemplo, asignar más recursos para la resolución de un incidente específico.

El proceso de escalado puede resumirse de la siguiente manera (Figura 7):

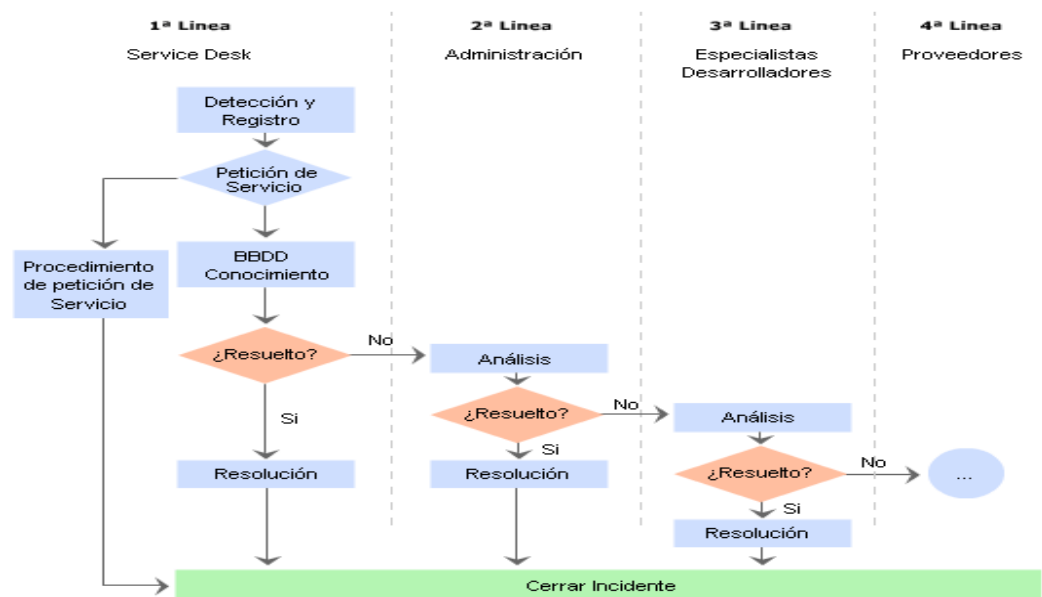


Figura 7. Proceso de Escalado

4.4. ROLES ITIL PARA LA GESTION DE INCIDENCIAS

Un rol es un conjunto de actividades y responsabilidades asignada a una persona o un grupo. En lo que concierne a Roles ITIL para el proceso de gestión de incidencias, IT Process Maps (2013) los clasifica de la siguiente manera:

- **Gestor de Incidencias**
 - El Gestor de Incidentes es responsable de la implementación efectiva del proceso de Gestión de Incidentes y prepara los informes correspondientes.
 - Ofrece representación durante la primera fase de escalado de incidencias, cuando no se pueden solucionar en el marco de los niveles de servicio acordados.
- **Equipo de Incidencias Graves**
 - Se trata de un equipo de gestores de TI y técnicos expertos establecido dinámicamente, bajo el mando de un Gestor de Incidentes, y formulado para concentrarse en la solución de un Incidente grave.
- **Soporte de Primera Línea**
 - La responsabilidad del Soporte de Primera Línea es registrar y clasificar los Incidentes reportados y llevar a cabo esfuerzos inmediatos para restaurar lo antes posible un servicio de TI que ha fallado.
 - Si no se encuentra una solución adecuada a estos fines, el Soporte de Primera Línea refiere el incidente a grupos de apoyo técnico especializado (Soporte de Segunda Línea).
 - El Soporte de Primera Línea también mantiene informados a los usuarios acerca del estatus de los Incidentes cada cierto tiempo.
- **Soporte de Segunda Línea**
 - El Soporte de Segunda Línea se hace cargo de los Incidentes que no pueden ser resueltos con los recursos del Soporte de Primera Línea.

- De ser necesario, requerirá apoyo externo de manufactureros de programados y de hardware.
- La meta es restaurar un servicio de TI fallido en el menor tiempo posible.
- Si no se encuentra solución, el Incidente debe ser referido a Gestión de Problemas.
- **Soporte de Tercera Línea**
 - Se encargara de una serie de grupos técnicos internos y/o terceros proveedores/mantenedores. Puede incluir:
 - Soporte de base de datos.
 - Soporte de red.
 - Soporte de servidor.
 - Ingenieros de Mantenimiento de Hardware
 - Proveedores.
 - Dependiendo en donde una organización obtiene el soporte de sus servicios, los grupos anteriores pueden ser internos o externos.

La Figura 8 muestra la participación de los roles en el proceso de gestión de incidentes.

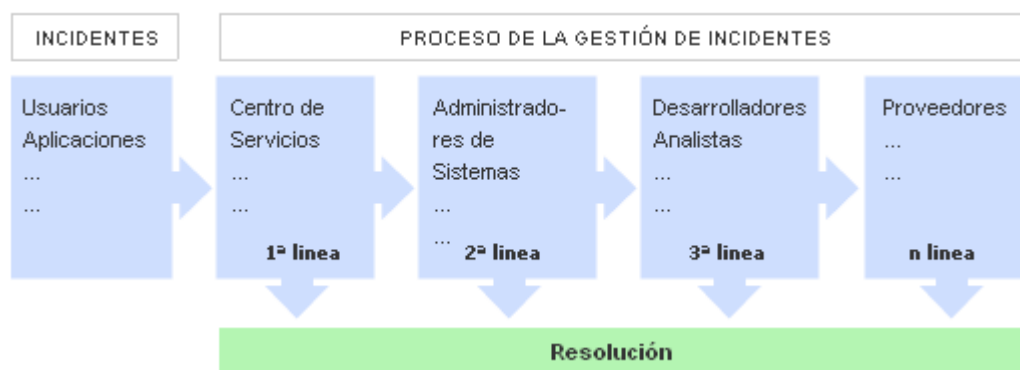


Figura 8. Roles durante el proceso de Gestión de Incidentes

4.5. PROCESO

La figura 9 ilustra los pasos que incluye el proceso de Gestión de Incidentes.

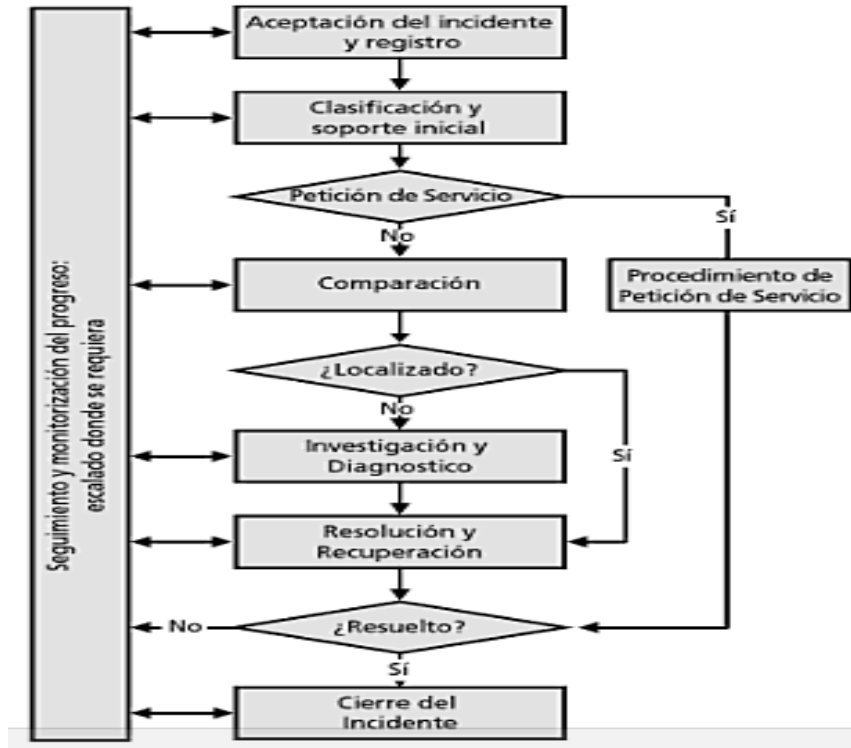


Figura 9. Proceso de Gestión de Incidentes.

- **Admisión y registro:** El incidente es detectado y se crea un registro del incidente.
- **Clasificación y soporte inicial:** El incidente se codifica por tipo, estado, impacto, urgencia, etc. Además, se puede sugerir al usuario una solución, aunque sea temporal.
- Si la llamada es por una **Petición de Servicio** se inicia el proceso que corresponde.
- **Comparación:** se investiga si el incidente es conocido y si se relaciona con un incidente existente, un problema o error conocido y si existe una solución o trabajo en curso relacionado.
- **Investigación y Diagnostico:** si no hay una solución conocida, se investiga el incidente.
- **Resolución y Recuperación:** una vez se encontró la solución, se resuelve el tema.

- **Cierre:** se pregunta al usuario si está satisfecho con la solución y se cierra el incidente.
- **Monitorización y Seguimiento del progreso:** se monitoriza el ciclo entero del incidente, y si se considera que no se puede resolver el problema a tiempo o con el actual nivel de experiencia, se continúa con el escalado.

4.5.1. CONTROL DEL PROCESO

De acuerdo a OSIATIS S.A. (2009), una correcta Gestión de Incidentes requiere de una infraestructura que facilite su correcta implementación. Entre ellos cabe destacar:

- Un correcto sistema automatizado de registro de incidentes y relación con los clientes.
- Una Base de Lecciones Aprendidas que permita comparar nuevos incidentes con incidentes ya registrados y resueltos. Una Base de Lecciones Aprendidas actualizada permite:
 - Evitar escalados innecesarios.
 - Convertir el “know how” de los técnicos en un activo duradero de la empresa.
 - Poner directamente a disposición del cliente parte o la totalidad de estos datos (a la manera de FAQs -> [Preguntas Frecuentes, Frequently Asked Questions]) en una Extranet. Lo que puede permitir que a veces el usuario no necesite siquiera notificar la incidencia.
- Una CMDB que permita conocer todas las configuraciones actuales y el impacto que estas puedan tener en la resolución del incidente.

Para el correcto seguimiento de todo el proceso es indispensable la utilización de métricas que permitan evaluar de la forma más objetiva posible el funcionamiento del servicio. Algunos de los aspectos clave a considerar son:

- Número de incidentes clasificados temporalmente y por prioridades.

- Tiempos de resolución clasificados en función del impacto y la urgencia de los incidentes.
- Nivel de cumplimiento del SLA.
- Costes asociados.
- Uso de los recursos disponibles en el Base de Lecciones Aprendidas.
- Porcentaje de incidentes, clasificados por prioridades, resueltos en primera instancia por la Base de Lecciones Aprendidas.
- Grado de satisfacción del cliente.

4.6. BENEFICIOS

Para el negocio en conjunto:

- Mayor resolución dentro del tiempo estipulado de los incidentes reduciendo el impacto en el negocio.
- Mayor productividad para el usuario.
- Monitorización de incidentes como proceso independiente, centrado en el cliente.
- Disponibilidad de información de producción centrada en la gestión del negocio de acuerdo con el SLA.

Para la organización de TI:

- Mejora de la monitorización, permitiendo una adecuada medición del rendimiento contra el SLA.
- Mejor y más eficaz uso del personal.
- No perder o registrar de manera incorrecta los incidentes y las peticiones de servicio.
- Mejora de la satisfacción del usuario y del cliente.

4.7. GESTION DE INCIDENCIAS COMO REFERENCIA PARA LA GESTION DE DEFECTOS

Debido a que ITIL ofrece buenas prácticas de cómo tratar la gestión de Incidencias durante la fase de Servicio de Operación para la gestión de defectos se puede tomar como referencia la misma analogía adaptándola a un ambiente de Calidad.

5. APLICACIONES WEB

5.1. DEFINICIÓN

Se denomina aplicación web a aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación de software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

Dentro de marco de desarrollo de aplicaciones web existen patrones de diseño creacionales que son utilizados para instanciar objetos, y así separar la implementación del cliente de la de los objetos que se utilizan y de ésta manera separar la lógica de creación de objetos y encapsularla. A continuación se explica uno de los patrones comúnmente usados en el ámbito de desarrollo web el cual tiene por nombre Modelo Vista Controlador.

5.2. PATRÓN DE ARQUITECTURA MODELO VISTA CONTROLADOR (MVC)

5.2.1. DEFINICIÓN

Es un patrón de arquitectura de desarrollo de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos.

5.2.2. DEFINICIÓN GENERAL DE LOS COMPONENTES EN APLICACIONES WEB

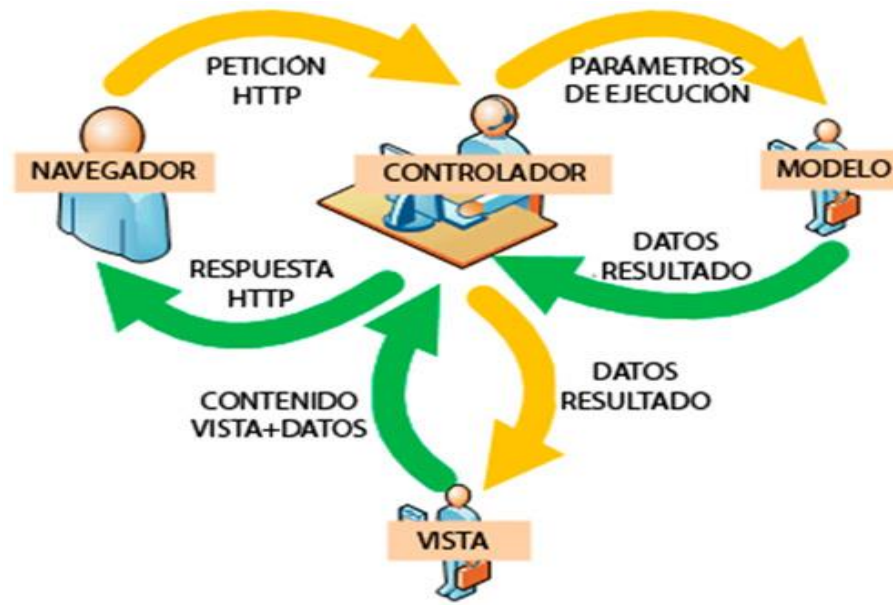
- **Modelo:** Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio).
- **Vista:** Presenta el modelo (información y lógica de negocio) en un formato adecuado para interactuar.
- **Controlador:** Administra la lógica de negocio; invoca peticiones al modelo y probablemente a la vista; recibe los eventos de entrada desde la vista, usualmente acciones del usuario.

5.2.3. RESPONSABILIDADES DE CADA COMPONENTE

- **Modelo:**
 - Acceder a la capa de almacenamiento de datos.
 - Llevar un registro de las vistas y controladores del sistema.
 - Si se está en un modelo activo, se notificará a las vistas los cambios que en los datos pueda producir un agente externo.
- **Vista:**
 - Recibir datos del modelo y lo muestra al usuario.
 - Tener un registro de su controlador asociado.
 - Dar el servicio de actualización, para que sea invocado por el controlador.
- **Controlador:**
 - Recibir los eventos de entrada (ej: un click, un cambio en un campo de texto).
 - Contener reglas de gestión de eventos (estas acciones pueden suponer peticiones al modelo o a las vistas).

5.2.4. FLUJO MVC

La figura 10, muestra como generalmente se cumple el flujo del modelo vista-controlador.



Modelo Vista Controlador

Figura 10. Flujo del Modelo Vista Controlador.

- a. El usuario interactúa con la interfaz de usuario de alguna forma.
- b. El usuario pulsa un botón, enlace, etc.
- c. El controlador recibe la notificación de la acción solicitada por el usuario (el controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos).
- d. El controlador accede al modelo, posiblemente modificándolo de forma correspondiente a la acción solicitada por el usuario.
- e. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario (la vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo).
- f. El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, se podría proveer una cierta in-dirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio (un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista).
- g. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

5.2.5. VENTAJAS

- Facilita de desarrollo y acortamiento del tiempo de respuesta.
- Facilita la especialización de perfiles de programación.
- Aumenta en gran medida el nivel de reusabilidad de código.
- Hay una clara separación entre los componentes de una aplicación web (Modelo, Vista y Controlador), lo que nos permite implementarlos por separado.
- Se pueden añadir y actualizar Controladores y Vistas conforme cambia el Modelo.

5.2.6. DESVENTAJAS

- La distribución de componentes obliga a crear y mantener un mayor número de ficheros.
- MVC es un patrón de diseño orientado a objetos por lo que su implementación es sumamente costosa y difícil en lenguajes que no siguen este paradigma.
- MVC requiere la existencia de una arquitectura inicial sobre la que se deben construir clases e interfaces para modificar y comunicar los módulos de una aplicación.
- Esta arquitectura inicial debe incluir, por lo menos, un mecanismo de eventos para poder proporcionar las notificaciones que genera el modelo de aplicación; una clase Modelo, otra clase Vista y una clase Controlador genéricas que realicen todas las tareas de comunicación, notificación y actualización que serán luego transparentes para el desarrollo de la aplicación.

5.3. ARQUITECTURA CLIENTE-SERVIDOR

5.3.1. DEFINICIÓN

Es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes llamados clientes.

5.3.2. CARACTERÍSTICAS

- El Cliente y el Servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades o tareas independientes.
- Las funciones de Cliente y Servidor pueden estar en plataformas separadas, o en la misma plataforma.
- Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los Clientes o de los Servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.
- La interrelación entre el hardware y el software están basados en una infraestructura poderosa, de tal forma que el acceso a los recursos de la red no muestra la complejidad de los diferentes tipos de formatos de datos y de los protocolos.
- Su representación típica es un centro de trabajo (PC), en donde el usuario dispone de sus propias aplicaciones de oficina y sus propias bases de datos, sin dependencia directa del sistema central de información de la organización.

6. TECNOLOGÍAS Y HERRAMIENTAS

A continuación se definen las tecnologías y herramientas que fueron utilizadas para el desarrollo del sistema de gestión de defectos, cabe destacar que las herramientas, metodologías y buenas prácticas a describir son las que usa la organización financiera para el desarrollo de sus proyectos.

Además, se describen los sistemas para la gestión de incidencias existentes en el mercado, en el caso de MantisBT y Jira, que ayudó a la hora de decidir cuáles características son necesarias para un mejor funcionamiento del sistema.

6.1. SOFTWARE DE DESARROLLO SPRING FRAMEWORK

6.1.1. DEFINICION

Spring Framework es una plataforma Java que proporciona un amplio soporte de infraestructura para el desarrollo de aplicaciones Java [GENBETADEV, 2011]. Spring se encarga de la infraestructura para que el desarrollador pueda centrarse en su aplicación.

6.1.2. MÓDULOS DE SPRING FRAMEWORK

Spring trabaja con una variedad de módulos los cuales se pueden observar en la figura 11.

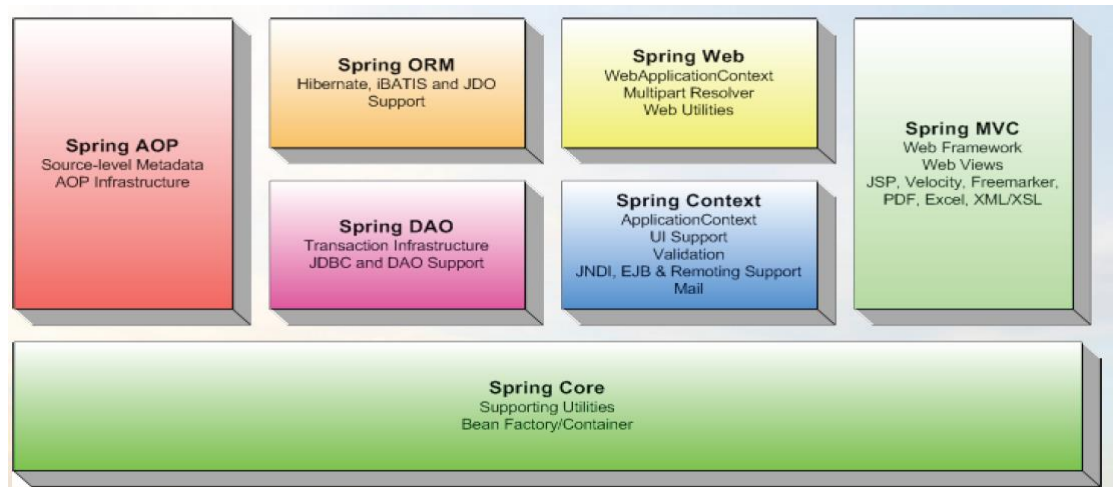


Figura 11. Módulos de trabajo de Spring Framework

Cada uno de los módulos o componentes que consisten en Spring Framework pueden valerse por sí mismos o ser ejecutados conjuntamente con uno o más de los otros módulos o componentes. La descripción de cada componente es la siguiente:

- **The Core Container:**

El contenedor de núcleo proporciona la funcionalidad fundamental Spring Framework. En este componente primario es el módulo de BeanFactory, una implementación del patrón de la Factory. El BeanFactory aplica la Inversión de Control (IOC) patrón para separar la configuración de una aplicación y especificación de dependencia del código de la aplicación real.

Este módulo comprende 4 sub-módulos los cuales se explican brevemente a continuación:

- **Módulo Core:** El módulo Core proporciona las partes fundamentales de la estructura, incluyendo las características de IoC (Inversión de Control) e inyección de dependencias.
 - **Inyección de dependencias:** La inyección de dependencia (DI) es un proceso mediante el cual los objetos definen sus dependencias, es decir, los otros objetos con los que trabajan, sólo a través de argumentos del constructor, argumentos a un método Factory, o propiedades que se establecen en la instancia del objeto después de que se construye o se devuelven a partir de un método Factory. El contenedor inyecta entonces esas dependencias cuando crea el Bean. Este proceso es fundamentalmente la inversa, de ahí el nombre de Inversión de Control (IoC), del mismo Bean controla la creación de instancias o ubicación de sus propias dependencias mediante el uso de la construcción directa de clases, o el patrón Service Locator.
 - **Inversión de Control:** Es una técnica que invierte el flujo tradicional. Lo tradicional es que el código que implementa el usuario llame a las librerías; la inversión de control ocurre cuando son las librerías las que llaman al código del usuario. En Spring, la inversión de control consiste en ceder el control a una entidad externa a la aplicación, llamada "**Contenedor**", que se encargará de gestionar las instancias (así como sus creaciones y destrucciones).
- **Módulo Beans:** El módulo bean proporciona un BeanFactory que es una sofisticada implementación del patrón de Factory.
- **Módulo Context:** El módulo de contexto se basa en la sólida base proporcionada por los módulos del Core y Beans y es un medio para

acceder a cualquier objeto definido y configurado. La interfaz de ApplicationContext es el punto central del módulo de Context.

- **Módulo Expression Language:** El módulo de lenguaje de expresión proporciona un poderoso lenguaje de expresión para consultar y manipular un gráfico de objetos en tiempo de ejecución.

- **Spring Context Module :**

El contexto de Spring es un archivo de configuración que proporciona información de contexto para el framework Spring. Además el contexto de Spring incluye servicios de la empresa, tales como eventos de correo electrónico, JNDI, EJB, internalización, validación, programación y aplicaciones de ciclo de vida. También se incluye soporte para la integración con los marcos de plantillas como la velocidad.

- **Spring AOP Module:**

El módulo Spring AOP permite que un componente de software que se decora con un comportamiento adicional, a través de su función de gestión de la configuración. Como resultado, se puede fácilmente habilitar AOP para cualquier objeto administrado por el framework Spring. El módulo Spring AOP proporciona servicios de gestión de transacciones para los objetos en cualquier aplicación basada en Spring. Con el Spring AOP se puede incorporar la gestión de transacciones declarativa en sus aplicaciones sin depender de los componentes EJB (Enterprise JavaBean).

- **Spring DAO Module:**

El módulo Spring DAO proporciona una capa de abstracción JDBC que reduce la necesidad de hacer la codificación tediosa JDBC y el análisis de los códigos de error específicos de bases de datos de proveedores. Además, el paquete JDBC proporciona una manera de hacer programática, así como la gestión de transacciones declarativa, no sólo para las clases que implementan interfaces especiales, sino para todos los POJOs del usuario (Plain Old Java Objects o Antiguo objeto plano de Java).

- **Spring ORM Module:**

Proporciona integración con las herramientas de mapeo objeto (Object Relational) como Hibernate, JDO e iBATIS. Soporta cada uno de los marcos ORM, así como JDBC.

- **Spring Web Module:**

Proporciona las características básicas para la integración Web y construye un contexto de este módulo aplicación, proporcionando contextos para las aplicaciones basadas en la Web. Como resultado, permite Spring soporta la integración con Struts. Este módulo también facilita las tareas de manejo de solicitudes de múltiples partes y de solicitud de parámetros de la unión de objetos de dominio.

- **Spring MVC Framework Module:**

Proporciona una arquitectura MVC (la implementación predeterminada del Proyecto Modelo - Vista - Control) conectable. Los usuarios tienen la opción de usar el framework de desarrollo web o seguir utilizando su framework Web existente. Spring separa las funciones del controlador, el modelo de objetos, el despachador y objeto de controlador que hace que sea más fácil de personalizar. Spring no requiere el uso de código JSP para la capa de vista. El usuario tiene la flexibilidad para utilizar JSP, plantillas XSLT, u otros, para proporcionar la vista.

6.2. SOFTWARE DE DESARROLLO JSF (JAVASERVER FACES)

6.2.1. DEFINICIÓN

Es una tecnología y framework para aplicaciones Java en plataforma web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa JavaServer Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías como XUL (acrónimo de XML-based User-interface Language, lenguaje basado en XML para la interfaz de usuario) [WIKIPEDIA, 2014].

6.2.2. CARACTERISTICAS

JSF incluye:

- Un conjunto de APIs para representar componentes de una interfaz de usuario y administrar su estado, manejar eventos, validar entrada, definir un esquema de navegación de las páginas y dar soporte para internacionalización y accesibilidad.
- Un conjunto por defecto de componentes para la interfaz de usuario.
- Dos bibliotecas de etiquetas personalizadas para JavaServer Pages que permiten expresar una interfaz JavaServer Faces dentro de una página JSP.
- Un modelo de eventos en el lado del servidor.
- Administración de estados.
- Beans administrados.

6.2.3. OBJETIVOS

Estos objetivos de diseño representan el foco de desarrollo de JSF:

- Definir un conjunto simple de clases base de Java para componentes de la interfaz de usuario, estado de los componentes y eventos de entrada. Estas clases tratarán los aspectos del ciclo de vida de la interfaz de usuario, controlando el estado de un componente durante el ciclo de vida de su página.
- Proporcionar un conjunto de componentes para la interfaz de usuario, incluyendo los elementos estándares de HTML para representar un formulario. Estos componentes se obtendrán de un conjunto básico de clases base que se pueden utilizar para definir componentes nuevos.
- Proporcionar un modelo de JavaBeans para enviar eventos desde los controles de la interfaz de usuario del lado del cliente a la aplicación del servidor.
- Definir APIs para la validación de entrada, incluyendo soporte para la validación en el lado del cliente.
- Especificar un modelo para la internacionalización y localización de la interfaz de usuario.

- Automatizar la generación de salidas apropiadas para el objetivo del cliente, teniendo en cuenta todos los datos de configuración disponibles del cliente, como versión del navegador.

6.2.4. COMPONENTE JSF: PRIMEFACES

○ **DEFINICION**

PrimeFaces es una librería de componentes para JavaServer Faces (JSF) de código abierto que cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las aplicaciones web. [WIKIPEDIA, 2014]

PrimeFaces está bajo la licencia de Apache License V2. Una de las ventajas de utilizar PrimeFaces, es que permite la integración con otros componentes.

○ **CARACTERISTICAS**

Algunas de las características que ofrece PrimeFaces son:

- Un gran conjunto de componentes (Editor de HTML, autocompletar, cartas, gráficas o paneles, entre otros).
- Soporte para Ajax, basándose en el estándar JSF 2.0 Ajax API
- Sin dependencias, ni configuraciones, además de ser muy ligero
- Soporte para interfaces de usuario sobre dispositivos móviles.
- Múltiples temas de apariencia prediseñados.

6.3. SOFTWARE DE DESARROLLO ORM: HIBERNATE

6.3.1. DEFINICION

Hibernate es una herramienta de Mapeo objeto-relacional (ORM) para la plataforma Java 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.

6.3.2. CARACTERÍSTICAS

Hibernate busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional). Para lograr esto permite:

- Detallar el modelo de datos y las relaciones existentes entre ellos.
- Permite a la aplicación manipular los datos en la base de datos operando sobre objetos, con todas las características de la POO.
- Convierte los datos entre los tipos utilizados por Java y los definidos por SQL. Hibernate
- Genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución.
- Hibernate está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente.
- Ofrece un lenguaje de consulta de datos llamado HQL (Hibernate Query Language), al mismo tiempo que una API para construir las consultas programáticamente (conocida como "criteria").

6.4. SOFTWARE DE DESARROLLO MICROSOFT SQL SERVER 2008

6.4.1. DEFINICIÓN

Es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL. [SLIDESHARE, 2010].

6.4.2. CARACTERÍSTICAS

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.

- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.

3.4.3 PROGRAMACIÓN

- T-SQL (Transact-SQL): Es el principal medio de interacción con el Servidor. Permite realizar las operaciones claves en SQL Server, incluyendo la creación y modificación de esquemas de la base de datos, la introducción y edición de los datos en la base de datos, así como la administración del servidor como tal. Esto se realiza mediante el envío de sentencias de T-SQL y declaraciones que son procesadas por el servidor y los resultados (o errores) regresan a la aplicación cliente.
- Cliente Nativo SQL: Cliente Nativo de SQL es la biblioteca de acceso a datos para los clientes de Microsoft SQL Server versión 2005 en adelante. Implementa nativamente soporte para las características de SQL Server, incluyendo la ejecución de la secuencia de datos tabular, soporte para bases de datos en espejo de SQL Server, soporte completo para todos los tipos de datos compatibles con SQL Server, conjuntos de operaciones asíncronas, las notificaciones de consulta, soporte para cifrado, así como recibir varios conjuntos de resultados en una sola sesión de base de datos.

3.4.4 VENTAJAS

- Reduce costes mediante la consolidación de servidores y también el coste de almacenamiento aplicando compresión de datos.
- Con la virtualización ahorra dinero en hardware y software.
- Mejora la continuidad de negocio con sus funcionalidades.
- Centraliza las tareas de análisis y elaboración de informes con data warehouse.
- Mejora la eficiencia de la empresa con sus panorámicas de datos completas.

- Optimiza los procesos de negocio con un mejor rendimiento y escalabilidad.
- Optimiza el ROI gracias a su compatibilidad e integración con la plataforma.
- Reduce los costes de seguridad y cumplimiento con sus medidas de seguridad corporativa.

3.4.5 DESVENTAJAS

- Usa Address Windowing extensión (AWE) para hacer el direccionamiento de 64-bit esto le impide usar la administración dinámica de memoria y sólo le permite alojar a lo más 64Gb de memoria compartida.
- Sólo permite 16 instancias distintas concurrentes aproximadamente en una máquina.
- No maneja compresión de datos por tanto ocupa mucho espacio en disco.
- Está atado a la plataforma de sistema operativo sobre la cual puede instalarse.

6.5. SOFTWARE DE DESARROLLO JAVADOC

6.5.1. DEFINICIÓN

Javadoc es una utilidad de Oracle para la generación de documentación de APIs en formato HTML a partir de código fuente Java. Javadoc es el estándar para documentar clases de Java. La mayoría de los IDEs utilizan Javadoc para generar de forma automática documentación de clases [WIKIPEDIA, 2013].

Para generar APIs con Javadoc han de usarse etiquetas (tags) de HTML o ciertas palabras reservadas precedidas por el carácter "@".

Estas etiquetas se escriben al principio de cada clase, miembro o método, dependiendo de qué objeto se desee describir, mediante un comentario iniciado con "/*" y acabado con "*/".

A continuación como se puede observar en la tabla 3, muestran algunas de las palabras reservadas:

Tag	Descripción	Uso	Versión
-----	-------------	-----	---------

@author	Nombre del desarrollador.	nombre_autor	1.0
@deprecated	Indica que el método o clase es antigua y que no se recomienda su uso porque posiblemente desaparecerá en versiones posteriores.	descripción	1.0
@param	Definición de un parámetro de un método, es requerido para todos los parámetros del método.	nombre_parametro descripción	1.0
@return	Informa de lo que devuelve el método, no se puede usar en constructores o métodos "void".	descripción	1.0
@see	Asocia con otro método o clase.	referencia (#método()); clase#método(); paquete.clase; paquete.clase#método()).	1.0
@throws	Excepción lanzada por el método	nombre_clase descripción	1.2
@version	Versión del método o clase.	versión	1.0

Tabla 3. Palabras reservadas Javadoc.

6.6. SISTEMA DE REFERENCIA: JIRA

6.6.1. DEFINICIÓN

JIRA es una aplicación basada en web para el seguimiento de errores, de incidentes y para la gestión operativa de proyectos. Jira también se utiliza en áreas no técnicas para la administración de tareas. La herramienta fue desarrollada por la empresa australiana Atlassian [WIKIPEDIA].

6.6.2. FUNCIONALIDAD

Jira permite informar sobre incidencias y gestionar errores en los proyectos. Por lo que JIRA es un sistema para informar de sucesos de las acciones llevadas a cabo a diario, entre las funcionalidades que lleva a cabo se encuentran:

- Categorizar según la importancia de la incidencia.

- Asignar incidencias a personas para que se hagan responsables y se resuelvan.
- Control de timing. Permite a cada miembro imputar el tiempo que ha dedicado a cada actividad y luego realizar informes sobre el mismo.
- El control de informes permite monitorizar el progreso en el proyecto.
- Control de errores, soporte, Gestión de proyectos y de procesos
- Compartir documentos. Imágenes, vídeos, documentos de texto, entre otros

6.6.3. VENTAJAS

Jira ofrece una gran cantidad de ventajas derivadas de ser una herramienta TIC [Juan Llado, 2011]:

- Gran cantidad de información de todo tipo: gráficas, audios, vídeos, documentos de texto, entre otros.
- Información fácilmente accesible.
- Procesamiento de la información. JIRA permite crear informes para toda la información que almacena.
- Gestión de grupos. Con la gestión de usuarios por roles que propone JIRA, se tiene controlado cada tipo de usuario que interviene en un proyecto.
- Es una arquitectura cliente servidor, necesita un ordenador con conexión a Internet o la Intranet para disponer de toda la información.
- Consultas en tiempo real.

6.6.4. DESVENTAJAS

- No es software gratuito. Sus licencias van desde 10\$ en adelante dependiendo de la cantidad de desarrolladores a usar el sistema.
- Posee una curva de aprendizaje muy elevada.

6.7. SISTEMA DE REFERENCIA: MANTISBT

6.7.1. DEFINICIÓN

Bilib (2011) establece que MantisBT (Mantis Bug Tracker) “es una aplicación software libre multiplataforma que permite gestionar las incidencias de una empresa, sistemas o proyectos. Es un sistema fácil de usar y adaptable a

muchos escenarios, tanto para tickets de soporte, reportes de incidencias técnicas, como bugs para proyectos de software”.

6.7.2. FUNCIONALIDAD

Por otra parte MantisBT posee varios tipos de funcionalidades, las cuales son categorizadas por Bilib (2011) de la siguiente manera:

- **Reporte de incidencias:** Mantis es una aplicación que permite a distintos usuarios reportar tickets de cualquier tipo, estando orientada la aplicación a muchos escenarios, como incidencias técnicas, peticiones de soporte o bugs de un sistema. El usuario puede describir con un breve título la incidencia, y también añadir un comentario descriptivo, además de un campo adicional para dar más detalles técnicos (como puede ser detalles de un equipo o de un software). Estos comentarios llegan al encargado o encargados de las incidencias, que pueden asignarla al responsable, y añadir comentarios como respuesta o pedir más datos para seguir un hilo de actividad hasta que quede resuelta.
- **Sistema de permisos de usuario:** Mantis incorpora un sistema de roles y permisos para identificar a los distintos usuarios que acceden al sistema. Los niveles de usuarios, comenzando por el que solo puede observar incidencias, y acabando por el que puede configurar todo el sistema, son: espectador, informador, actualizador, desarrollador, manager y administrador. Cada rol define las acciones que ese tipo de usuario puede realizar, por ejemplo el informador puede reportar incidencias y añadir nuevas notas, pero no puede asignar o modificar incidencias. Solo el administrador del sistema puede configurar los permisos de cada rol.
- **Completa descripción y estado de incidencias:** A la hora de reportar una incidencia, el usuario tiene muchas opciones y campos a rellenar con el fin de hacer más fácil el trabajo del encargado de resolver el ticket. Aunque por supuesto, lo más básico es poner un título y la descripción básica, y mediante notas el encargado de resolverlas puede pedir más datos. Algunos de los datos a rellenar son la reproducibilidad o réplica del error, la urgencia, la severidad, la

prioridad, el sistema operativo y su versión o los pasos específicos para reproducirlo.

- **Notificaciones de usuario:** Mantis permite notificar a los usuarios de novedades por correo electrónico. De esta manera, el responsable asignado de resolver la incidencia se da cuenta por correo electrónico, y cualquier nuevo dato de la incidencia es enviada por correo tanto al responsable como al reportador de la incidencia.
- **Personalización del sistema:** Mantis ofrece al administrador de la aplicación mucha flexibilidad a la hora de la configuración. Permite definir nuevos estados para las incidencias, así como la creación de nuevos campos, y personalizar los permisos de los roles de usuario. También permite administrar etiquetas, y organizar la información de la aplicación por proyectos, pudiendo ser unos privados o públicos, y con acceso a ciertos usuarios. Dentro de los proyectos se pueden generar sub-proyectos y categorías.
- **Otras características:** Mantis implementa otras características como seguir la actividad por RSS, utilizar time tracking para medir el tiempo que se tarda en resolver las incidencias, generar informes estadísticos, o publicar anuncios globales.

6.7.3. VENTAJAS

Según Carlos Gallo (2014) se establecen las siguientes ventajas y desventajas en MantisBT:

- Sistema completo de seguimiento de errores que además incluye un sistema de usuarios, de tal forma que varios usuarios puedan interactuar y pueda haber un seguimiento sobre múltiples proyectos.
- Instalación y uso simple.
- Soporta proyectos, sub-proyectos y categorías.
- Varios niveles de acceso.
- Permite extraer informes y estadísticas.
- Herramienta de búsqueda y filtrado.
- Sistema de reportes incluido.
- Notificación por email.
- Incidencias personalizables y asignables.
- Sistema económico ya que no supone coste alguno para la empresa.

6.7.4. DESVENTAJAS

- Interfaz de usuario anticuada.
- Desarrollado en software libre, el soporte no está garantizado.

7. METODOLOGÍAS ORIENTADAS AL DESARROLLO DE SOFTWARE

Las metodologías y estándares utilizadas para el desarrollo de un software nos proporcionan las guías necesarias para poder conocer todo el camino a recorrer desde antes de empezar la elaboración y diseño del producto, con lo cual se asegura la calidad del producto final, así como también el cumplimiento en la entrega del mismo en un tiempo estipulado.

Las metodologías orientadas al control de procesos: Usualmente llamadas “metodologías pesadas” o “tradicionales”, donde se establecen rigurosamente las actividades a desarrollar, herramientas a utilizar y notaciones que se usarán para el desarrollo del software.

Estas se centran en la definición detallada de los procesos y tareas a realizar, herramientas a utilizar, y requiere una extensa documentación, ya que pretende prever todo de antemano. Este tipo de metodologías son más eficaces y necesarias cuanto mayor es el proyecto que se pretende realizar respecto a tiempo y recursos que son necesarios emplear, donde una gran organización es requerida.

En el área de tecnología de la información de la organización financiera se utiliza la metodología RUP (Rational Unified Process) para el desarrollo de sus sistemas.

7.1. METODOLOGÍA RATIONAL UNIFIED PROCESS (RUP)

7.1.1. DEFINICIÓN

La metodología RUP, cuyas siglas en inglés significan Rational Unified Process o también conocido como Proceso Unificado de Racional, es un proceso de ingeniería de software, bien definido y estructurado, que proporciona un enfoque para la asignación de tareas y responsabilidades dentro de una organización de desarrollo, este producto provee un marco de proceso adaptable a las

necesidades y características de cada proyecto específico.

RUP mejora la productividad del equipo ya que permite que cada miembro del grupo sin importar su responsabilidad específica pueda acceder a la misma base de datos. Esto hace que todos compartan el mismo lenguaje, la misma visión y el mismo proceso acerca de cómo desarrollar un software.

7.1.2. CARACTERÍSTICAS

Está dirigido por los Casos de Uso: Esto quiere decir que los Casos de Uso son la base para el desenvolvimiento y desarrollo de las disciplinas con los artefactos, roles y actividades necesarias.

Es un proceso centrado en la arquitectura: Define la arquitectura de un sistema, es decir, la organización o estructura de sus partes más relevantes, y una arquitectura ejecutable, implementación parcial del sistema, construida como un prototipo evolutivo.

Es iterativo e incremental: RUP plantea la implementación del proyecto en Iteraciones, con lo cual se pueden definir objetivos por cumplir en cada iteración y así poder completar todo el proyecto iteración por iteración, este método tiene ventajas, ya que se pueden obtener pequeños avances del proyecto que son entregables al cliente el cual lo puede probar mientras se está desarrollando otra iteración del proyecto.

7.1.3. FASES RUP

La metodología RUP promueve el desarrollo iterativo y organiza el desarrollo de software en cuatro fases, cada una compuesta por una o más iteraciones ejecutables del software durante su etapa de desarrollo.

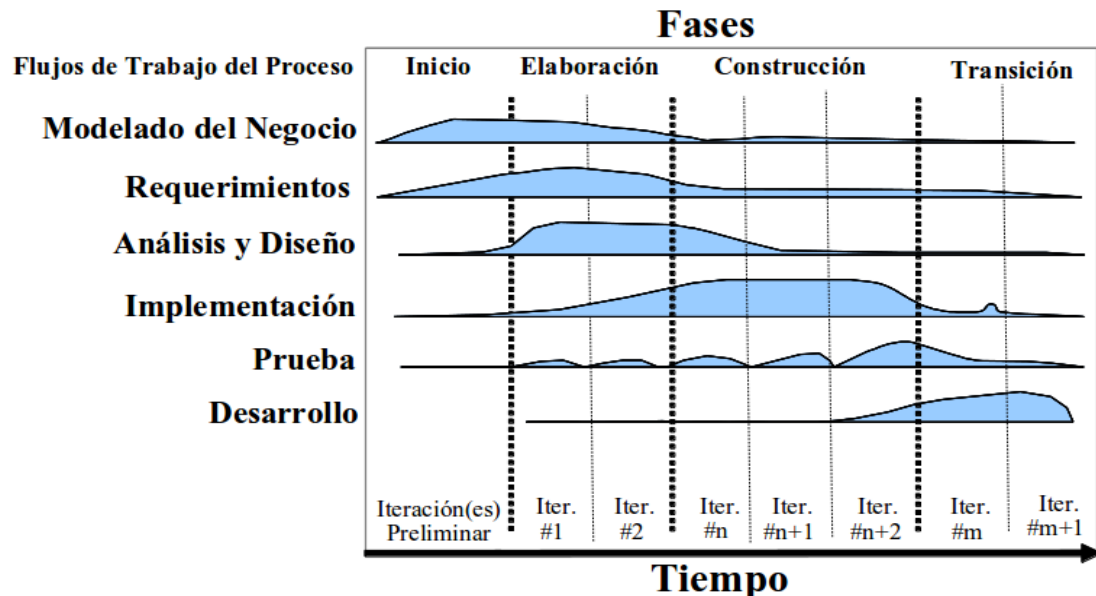


Figura 12. Proceso RUP.

Como se muestra en la Figura 12, RUP se repite a lo largo de una serie de ciclos. Cada ciclo consta de cuatro fases. En cada extremo de una fase se realiza una evaluación (actividad: Revisión del ciclo de vida de la finalización de fase) para determinar si los objetivos de la fase se han cumplido. Una evaluación satisfactoria permite que el proyecto se mueva a la próxima fase. Estas fases son:

- **Inicio (o Concepción):** El objetivo principal de esta fase es determinar la visión y alcance del proyecto. En esta fase, se desarrolla una descripción del producto final y se definen la funcionalidad y capacidades del producto, así como las estimaciones con respecto al costo de desarrollo y el tiempo.
- **Elaboración:** El objetivo es determinar la arquitectura óptima para el proyecto y el estudio en profundidad de su funcionalidad y el dominio del problema. En esta fase se planifica el proyecto considerando los recursos disponibles y se especifican en detalle la mayoría de los casos de uso del producto.
- **Construcción:** Durante esta fase el producto se desarrollara incrementalmente a través de iteraciones, donde cada iteración involucra tareas de análisis, diseño e implementación. Además, se empieza a documentar tanto el sistema construido, como el manejo del mismo. Esta fase proporcionará un modelo completo de diseño y casos de usos al igual que un producto construido junto con su documentación.
- **Transición:** En esta fase se logra la transición final del producto de software al

usuario. Se libera el producto y se entrega al usuario para un uso real. Los manuales de uso del producto se completan y refinan con la información obtenida de las pruebas para lograr obtener mejoras del producto.

Cabe destacar que los defectos se ubican después de la fase de Construcción, es decir, en la fase de Transición de la metodología RUP (Certificación de usuarios en Calidad y antes del pase a Producción).

7.1.4. DISCIPLINAS DE DESARROLLO

- **Modelado del Negocio:** Entendiendo las necesidades del negocio.
- **Análisis de Requerimientos:** Traslado de las necesidades del negocio a un sistema automatizado.
- **Análisis y Diseño:** Traslado de los requerimientos dentro de la arquitectura de software.
- **Implementación:** Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- **Pruebas:** Asegurándose que el comportamiento requerido es el correcto y que todo el solicitado está presente.
- **Despliegue:** Hacer todo lo necesario para la salida del proyecto.

7.1.5. ROLES RUP

Son los personajes encargados de la realización de las actividades definidas dentro de los flujos de trabajo de cada una de las disciplinas del RUP, estos actores se dividen en varias categorías:

- **Analistas**
 - Analista del Proceso del Negocio
 - Diseñador del Negocio

- Revisor del Modelo del Negocio
- Revisor de Requerimientos
- Analista del Sistema
- Especificador de Casos de Uso
- Diseñador de Interfaz del Usuario
- **Desarrolladores**
 - Arquitecto
 - Revisor de la Arquitectura
 - Revisor del Código y Revisor del Diseño
 - Diseñador de la Base de Datos
 - Implementador y un Integrador
- **Probadores Profesionales**
 - Diseñador de Pruebas
 - Probador
 - Encargados
 - Encargado de Control del Cambio
 - Encargado de la Configuración.
 - Ingeniero de Procesos
 - Encargado de Proyecto
 - Revisor de Proyecto
- **Otros**
 - Cualquier trabajador
 - Artista Gráfico
 - Administrador del Sistema
 - Técnico

7.1.6. ARTEFACTOS

Los artefactos son el resultado parcial o final que es producido y usado por los actores durante el proyecto. Son las entradas y salidas de las actividades, realizadas por los actores, los cuales utilizan y van produciendo estos artefactos para tener guías. Un artefacto puede ser un documento, un modelo o un elemento de modelo.

Se tiene un conjunto de artefactos definidos en cada una de las disciplinas y utilizadas dentro de ellas por lo actores para la realización de las mismas, a

continuación se definen cada una de estas categorías o grupos de artefactos dentro de las disciplinas del RUP:

a) Modelado del negocio

Los artefactos del modelado del negocio capturan y presentan el contexto del negocio del sistema. Los artefactos del modelado del negocio sirven como entrada y como referencia para los requisitos del sistema.

b) Requerimientos

Los artefactos de requerimientos del sistema capturan y presentan la información usada en definir las capacidades requeridas del sistema.

c) Análisis y diseño del sistema

Los artefactos para el análisis y del diseño capturan y presentan la información relacionada con la solución a los problemas se presentaron en los requisitos fijados.

7.1.7. RUP Y LAS MEJORES PRACTICAS

RUP describe la manera de implementar efectivamente "mejores prácticas" comercialmente probadas para el desarrollo de software.

RUP ofrece a cada miembro del equipo las guías, plantillas y herramientas necesarias para aprovechar al máximo las siguientes mejores prácticas:

- **Desarrollar software iterativamente:** RUP sugiere el desarrollo iterativo que aborda los riesgos más altos en cada paso del ciclo de vida, reduciendo en forma significativa los riesgos del proyecto.
- **Administrar requerimientos:** RUP describe cómo obtener, organizar y documentar la funcionalidad requerida y sus restricciones; rastrear y documentar especificaciones y decisiones, capturar y comunicar fácilmente requerimientos de negocio.
- **Utilizar arquitecturas basadas en componentes:** RUP soporta el desarrollo de software basado en componentes, módulos o subsistemas que cumplen una función clara. (De Carlo et al., 2004).

- Modelar el software visualmente: La base para el modelado visual es el Unified Modeling Language (UML), creado por Rational software y que se ha convertido en un estándar del mercado. (De Carlo et al., 2004).
- Verificar la calidad del software: RUP describe cómo obtener, organizar y documentar la funcionalidad requerida y sus restricciones; rastrear y documentar especificaciones y decisiones, capturar y comunicar fácilmente requerimientos de negocio.
- Controlar los cambios al software: El proceso describe cómo controlar, seguir y monitorear cambios para permitir el desarrollo iterativo.

8. DEFINICIÓN DE ESTÁNDARES Y BUENAS PRÁCTICAS

Los estándares y buenas prácticas son de gran importancia en cualquier proyecto de desarrollo sobre todo cuando varios desarrolladores están trabajando sobre el mismo proyecto. Con la existencia de estándares de codificación se garantiza que el código tenga menos errores, de alta calidad y fácil mantenimiento.

Los siguientes estándares y buenas prácticas son de uso obligatorio y fueron indicados por el equipo de desarrollo de la organización financiera.

Los objetivos a alcanzar con la realización de estas prácticas son:

- Código sencillo y fácil de utilizar.
- Documentación del código y la lógica del mismo, así como convenciones para las constantes y variables, como parte de las buenas prácticas.
- Definir formas y estilos de codificación así como también establecer un formato para estándares de documentación y comentarios.

8.1. CONVENCIONES

InfixCaps o CamelCase: Es un estilo de escritura que se aplica a frases o palabras compuestas. Existen dos tipos de CamelCase:

- UpperCamelCase: es cuando la primera letra de cada una de las palabras es mayúscula. Ejemplo: EjemploDeUpperCamelCase.
- lowerCamelCase: la primera letra es minúscula y la primera letra de cada una de las palabras siguientes es mayúscula. Ejemplo: ejemploDeLowerCamelCase

Para este proyecto se hará uso UpperCamelCase para definición de clases e interfaces y lowerCamelCase cuando se defina métodos, funciones, variables etc.

8.2. ARCHIVO DE CÓDIGO FUENTE

Nombre de Archivos Fuentes: Debido a que Java tiene la capacidad de soportar nombres largos, es recomendable que al momento de asignar un nombre a un archivo se coloque en mayúscula la inicial de cada palabra usando el estilo de escritura conocido como UpperCamelCase, tomando en cuenta que los caracteres permitidos en el nombre serán alfanuméricos y pueden contener el símbolo underscore “_”. Ejemplo: Action_ProductoCliente.java

8.3. ENCABEZADOS DE ARCHIVOS DE CÓDIGO FUENTE

El encabezado en cada archivo creado o editado consiste en reservar un área comentada al principio del archivo para colocar información acerca del contenido del archivo, el autor, fecha de creación, una breve explicación acerca de la funcionalidad desarrollada y una bitácora de los cambios realizados en el mismo. Este se divide en dos secciones, una sección que contendrá información general y una segunda que contendrá la bitácora de modificaciones.

Un ejemplo de lo que puede contener una sección de código general es la siguiente:

- **Institución:** nombre de la organización financiera.
- **Sistema:** nombre del sistema al cual está asociado el programa.
- **Siglas:** siglas del sistema al que está asociado el programa.
- **Nombre:** nombre exacto del proyecto.
- **Fecha de creación:** fecha en la cual fue creado el programa.
- **Versión:** Número de versión del documento.
- **Descripción:** Una breve descripción del objetivo del programa.
- **Autor:** Nombre del autor o desarrolladores originales del programa.

Un ejemplo de la sección bitácora de modificaciones puede ser el siguiente:

- **Autor:** Nombre del autor o desarrolladores originales del programa.
- **Fecha Inicio:** Fecha en la que se inició el desarrollo o la modificación del programa.

- **Fecha Fin:** Fecha en la que se finalizó el desarrollo o la modificación del programa.
- **Descripción:** Una breve descripción del objetivo del programa.

Un ejemplo del formato de encabezado se puede observar en la figura 13.

```

/*****
Organizacion Financiera de Caracas
Sistema:          Sistema de Gestion de Defectos.
Siglas:           SGD
Nombre:           ConsultarDefectos.java
Fecha creación:   19/01/2015
Versión:          1.1

Descripción:
  Procesamiento de solicitud de informacion de defectos reportados

Autor:
  Alvaro Marciales
  Claudio Torrez
---o---

Bitácora de Modificaciones:
Autor          Fecha Inicio   Fecha Fin   Descripción
Claudio Torrez 21/03/2015   29/04/2015 Creacion inicial de consulta del defecto
Alvaro Marciales 10/05/2015  18/05/2015 Agregados campos adicionales

---o---

NOTA: Recuerda no eliminar código ya desarrollado, debes comentar la porción de
código e incluir la nueva, documentando la fecha de la sustitución.
*****/

```

Figura 13. Formato de Encabezado.

CAPÍTULO 3: MARCO APLICATIVO

1. CONTEXTO DEL DESARROLLO

Este sistema ha sido desarrollado en el contexto de Trabajo Especial de Grado presentado ante la ilustre Universidad Central de Venezuela, Facultad de Ciencias, Escuela de Computación, bajo la tutoría de la profesora Concettina Di Vasta, para optar al Título de Licenciado en Computación, por parte de los bachilleres Claudio Torrez y Álvaro Marciales como equipo desarrollador.

2. PROYECTO

El presente TEG se basa en el desarrollo de un sistema que facilite a la organización financiera el seguimiento y control de los defectos de sus proyectos en el área de tecnología de información.

Para conseguir este objetivo, se toman en cuenta la gestión de proyectos en fase de desarrollo y las recomendaciones indicadas por ITIL (Information Technologies Infrastructure Library) acerca de las incidencias debido a que contienen procesos similares en comportamiento con respecto a los defectos pero en ambientes distintos (Calidad y Producción).

El sistema desarrollado es de tipo cliente/servidor e implementado usando tecnologías web de la organización financiera tales como: HTML, CSS, JavaScript, JavaServer Faces, el framework de desarrollo y arquitectura Spring y el sistema manejador de bases de datos Microsoft SQL Server 2008 integradas con otras tecnologías como el framework de persistencia de datos Hibernate, PrimeFaces como librería Front-End y PrettyFaces URL Rewrite para efectos de seguridad de URL's del sistema.

Para la realización de este sistema se utilizó una adaptación de la metodología pesada RUP, similar a la especificada en la Figura 12 del capítulo anterior, definido por la organización financiera para la realización de sus proyectos. A continuación se presenta las diferentes disciplinas o fases que cubre el proyecto siguiendo la metodología RUP.

2.1. FASES DE RUP

2.1.1. FASE DE INICIO

Esta fase se realizó en un período estimado de 4 a 6 semanas, donde se listan los requerimientos, el análisis de los recursos y la situación actual del sistema.

Para la captura de los requerimientos se llevó a cabo observaciones del sistema actual y reuniones con los empleados del área de tecnología, con la finalidad de identificar las funcionalidades que debe tener el sistema que se implementó.

Las observaciones realizadas consistieron en identificar el flujo actual de la gestión de los defectos durante el desarrollo de los proyectos del área de tecnología.

➤ Descripción de los recursos tecnológicos de la organización financiera:

La organización financiera cuenta para sus proyectos con frameworks de desarrollo tales como JavaServer Faces, el framework de desarrollo y arquitectura Spring, librerías de desarrollo PrimeFaces e Hibernate como framework de persistencia de datos y el sistema manejador de bases de datos Microsoft SQL Server 2008, entre otras tecnologías.

En el diagrama 1 se muestra los componentes tecnológicos que están integrados al sistema de gestión de defectos.

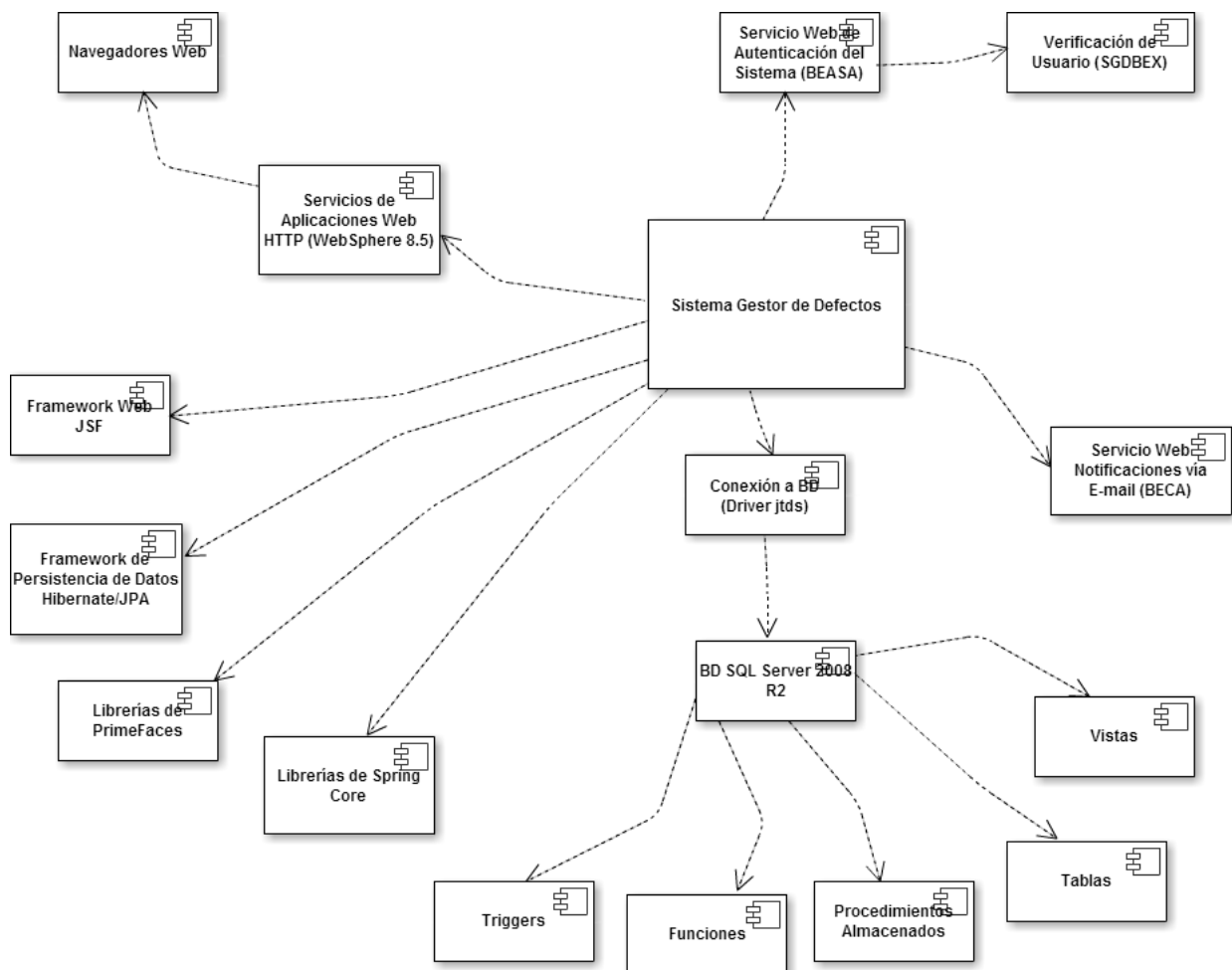


Diagrama 1. Diagrama de Componentes.

➤ **Capacidades del proyecto**

El proyecto está diseñado para que cada miembro del equipo de tecnología de la organización planifique y haga seguimientos de los proyectos y sus defectos, e incluir a los Gerentes de la organización para permitirles visualizar el rendimiento en cada proyecto.

Entre los requerimientos necesarios para el sistema se encuentran:

- Definir actores y funcionalidades.
- Definir flujo de defectos.
- Definir estructura de la aplicación.
- Definir reportes de la aplicación.
- Diseñar base de datos.

- Diseñar interfaces de aplicación.
- Desarrollar base de datos.
- Desarrollar interfaces y funcionalidades.
- Invocar servicios para el inicio de sesión y envío de correos del banco.
- Implementar manejo de las bitácoras en la aplicación.

➤ **Flujo actual de la gestión de defectos:**

El recorrido de una solicitud para el soporte de defectos en la empresa se refleja en la figura 14, en la cual un usuario funcional, que se encarga de validar los proyectos de la entidad financiera en el ambiente de calidad, detecta un error y procede a reportar un defecto en el sistema.

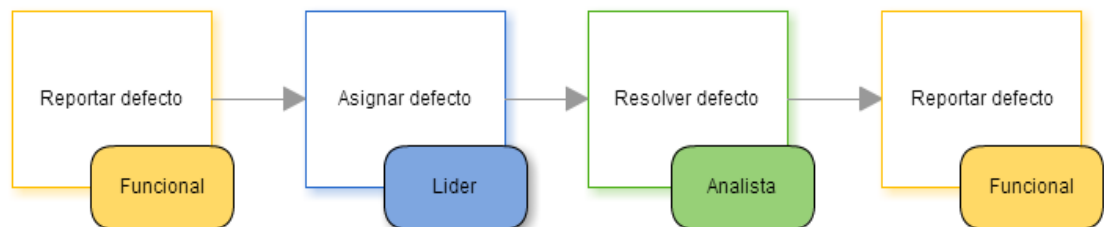


Figura 14. Flujo del Sistema.

Éste defecto es informado al líder del proyecto mediante un correo y se le añade la descripción del defecto y su reproducibilidad a una hoja Excel, donde se encuentran todos los defectos reportados en dicho proyecto.

El líder de proyecto deberá encargarse de la asignación de ese defecto reportado a un desarrollador para lograr su solución.

El desarrollador recibirá la notificación y procederá a lograr la solución de ese defecto, y en caso de encontrarla, actualizara la hoja Excel y notificara al usuario funcional que el defecto ha sido resuelto.

Posteriormente el funcional validara en el sistema si el defecto fue solucionado y actualizara su estatus en el documento Excel.

➤ **Usuarios**

En base a los resultados de las observaciones y siguiendo el lineamiento de la metodología se elaboró el siguiente modelo del negocio como se muestra en la figura 15:

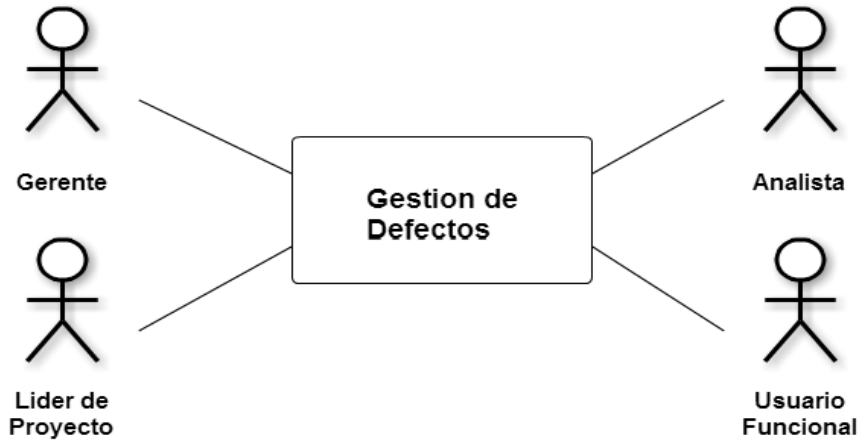


Figura 15. Modelo del Contexto del Negocio.

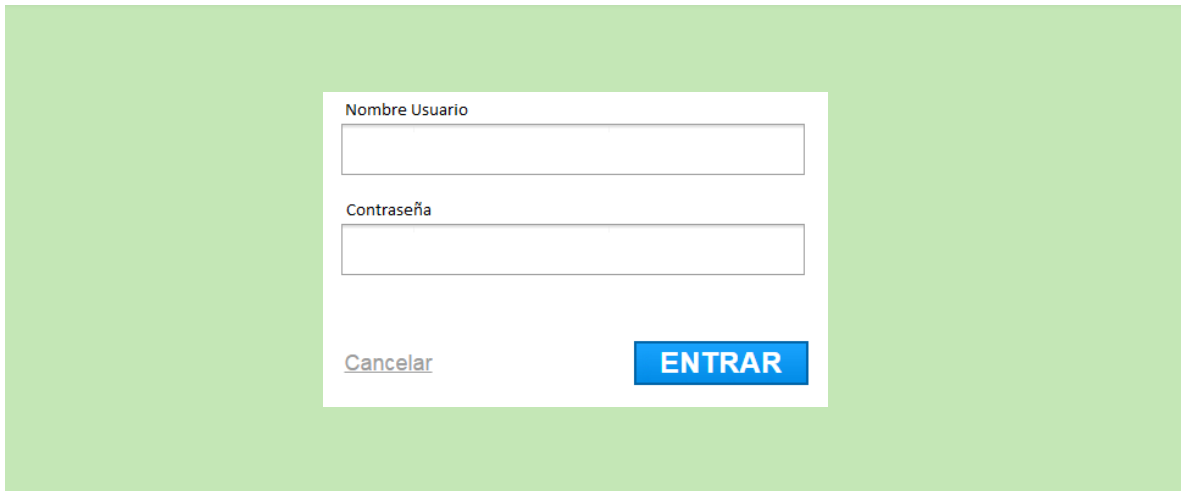
Los tipos de usuario que interactúan con el sistema son:

1. **Usuario Funcional:** Quien realiza las pruebas del sistema y reporta defectos en el sistema.
2. **Líder de Proyecto:** Encargado de verificar, evaluar, asignar analista responsable y hacerle el seguimiento a los reportes de defectos.
3. **Analista:** Encargado de resolver las solicitudes.
4. **Gerente:** Quien comprueba el desempeño en el desarrollo del proyecto y visualiza los reportes cuantitativos del sistema.
5. **Administrador:** Encargado de la administración general del sistema, entre ellos la gestión de usuarios, categorías y proyectos.

Cada uno de estos tipos de usuario representa a un actor y su interacción con el sistema se reflejara en los diagramas de casos de uso presentados en la siguiente sección.

➤ **Prototipo inicial del Sistema Gestor de Defectos:**

▪ **Login:**



The image shows a login form on a light green background. The form is white and contains two input fields: 'Nombre Usuario' and 'Contraseña'. Below the fields are two buttons: 'Cancelar' (underlined) and 'ENTRAR' (in a blue box). The text 'Nombre Usuario' and 'Contraseña' is positioned above their respective input boxes.

Figura 16. Prototipo Login SGD

Descripción de campos:

- Nombre de Usuario: Nombre obtenido desde el Web Services BEASA válido para el Sistema Gestor de Defectos (SGD).
- Contraseña: Contraseña obtenida desde el Web Services BEASA válida para el sistema SGD.

Observaciones:

▪ **Cabecera de aplicación con Menú:**



Figura 17. Prototipo Cabecera de Aplicación con Menú

La cabecera posee el logo de la institución bancaria, Combo Box desplegable para seleccionar un proyecto en particular y el menú principal de opciones.

Observaciones:

▪ **MiVista:**



Figura 18. Prototipo Menú opción MiVista

Esta primera opción del menú principal muestra todos los defectos relacionados con el usuario autenticado en el sistema, resaltando la siguiente información:

- Defectos Pendientes por asignar.
- Defectos Reportados por mí.
- Defectos Modificados recientemente.
- Defectos Resueltos.

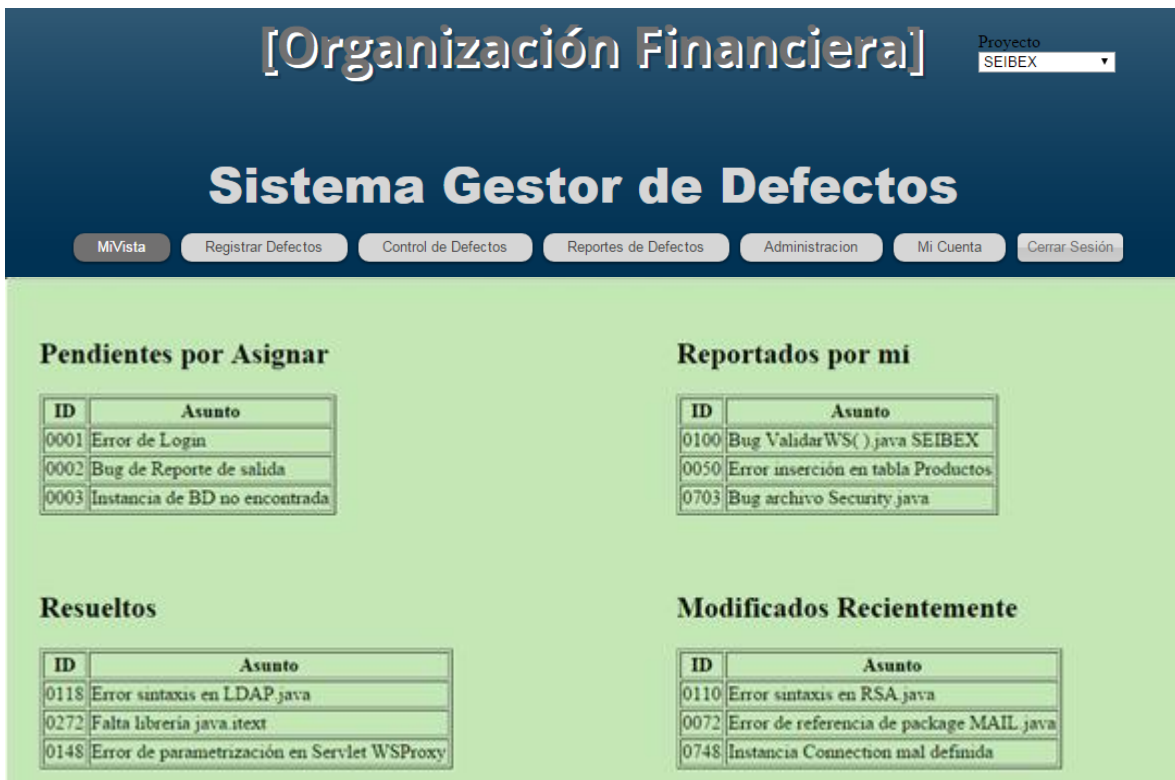


Figura 19. Prototipo MiVista.

Observaciones:

- **Gestión de Defectos:**



Figura 20. Prototipo Menú opción Control de Defectos.

Esta opción permite realizar las operaciones RUD (Read, Update y Delete) sobre cada uno de los defectos asociados a un proyecto en particular.

Priorid	ID	Asunto	Descripción	Proyecto	Categoría	Líder de Proyecto	Archivos adjuntos	Estado
→	1	Registro de Usuario	Fallo lectura Username	Sistema de Ticket	Diseño	Ana Dasilva	📎	Cerrado
→	2	Bug de Seguridad		DW Corporativo	Calidad	Rommel Torrez		Asignado
→	3	Registro de Usuario	Cambiar tipo de dato en el campo CI del cliente	DW Corporativo	Calidad	Rommel Torrez		Asignado
↓	4	Registro de Usuario	Cambiar fuente de botones	Sistema de Ticket	Seguridad	Ana Dasilva		Re-abierto
↗	5	Bug Seguridad	Alteracion de permisologias al usuario Admin	DW Corporativo	Seguridad	Rommel Torrez	📎	Cerrado
↗	6	Bug Servicio de Calidad		Sistema BEASA	Otros	Elvis Perez		Abierto

Figura 21. Prototipo Control y Gestión de Defectos.

- **Registrar Defecto:**

Sistema Gestor de Defectos

MiVista
Registrar Defectos
Control de Defectos
Reportes de Defectos
Administracion
Mi Cuenta
Cerrar Sesión

||Reportar Defecto||

Asignado a:

Reportado por:

Prioridad: Alta
 Media
 Baja

Estado: Abierto

Fecha Estimada de Solución:

Adjuntar Archivo
 Ningún archivo seleccionado

Categoría:

Figura 22. Prototipo Reportar Defectos.

Operaciones CRUD (CREATE,READ,UPDATE,DELETE):

<div style="display: flex; justify-content: space-between; align-items: center;"> New Duplicate Edit Delete Send Filter Refresh More Actions <div style="border: 2px solid red; padding: 2px; border-radius: 5px;"> <input type="text" value="Search on text..."/> </div> Clear </div>							
Filter:	All	Priorid	ID	Asunto	Descripcion	Proyecto	Categoria
<input type="checkbox"/>			1	Registro de Usuario	Fallo lectura Username	Sistema de Tickets	Diseño
<input type="checkbox"/>			2	Bug de Seguridad		DW Corporativo	Calidad
<input type="checkbox"/>			3	Registro de Usuario	Cambiar tipo de dato en el campo CI del cliente	DW Corporativo	Calidad
<input type="checkbox"/>			4	Registro de Usuario	Cambiar fuente de botones	Sistema de Tickets	Seguridad
<input type="checkbox"/>			5	Bug Seguridad	Alteracion de permisologias al usuario Admin	DW Corporativo	Seguridad
<input type="checkbox"/>			6	Bug Servicio de Calidad		Sistema BEASA	Otros
<input type="checkbox"/>			7	Bug Seguridad	Cambiar colores a los botones	Sistema BEASA	Diseño
<input type="checkbox"/>			8	Bug Servicio de Calidad		Sistema de Tickets	Diseño
<input type="checkbox"/>			9	Login de usuario	Aumentar tamaño del	Sistema ARS400	Diseño

Figura 23. Prototipo Operaciones CRUD.

- Filter: Filtros de ordenamiento de datos (Ej: Usuario puede filtrar por columnas u ordenar los datos alfabéticamente).
- Search: Mecanismo de búsqueda rápida en el sistema.

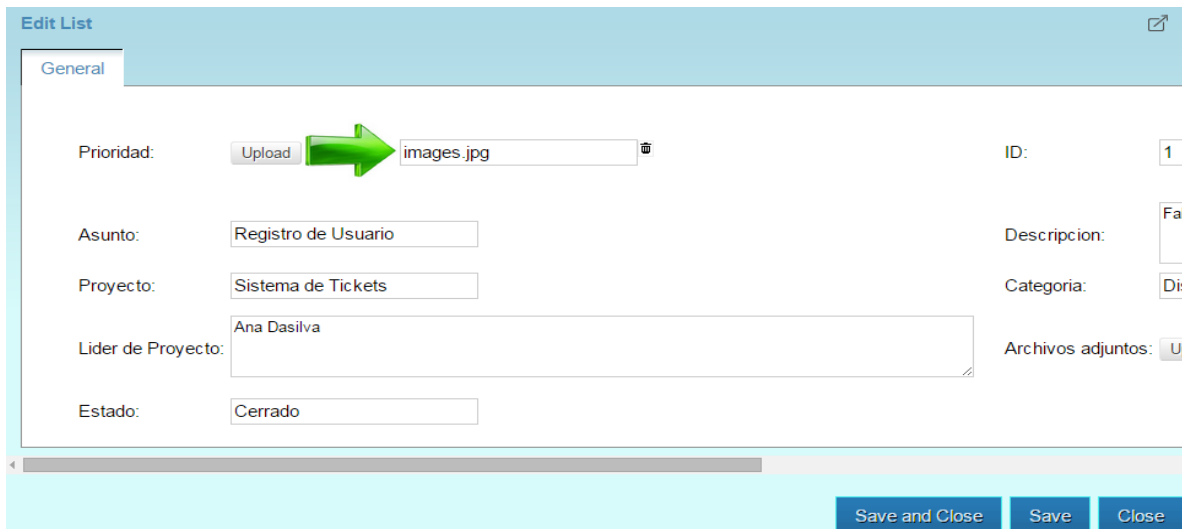


Figura 24. Prototipo Editar Defecto.

Observaciones de operaciones del CRUD:

- **Flujo Estados de Defectos:**

En la vista detalle de cada defecto, se muestra el flujo de estados por lo que ha pasado el mismo a través de su ciclo de vida en el sistema, desde su etapa inicial (Status="Abierto") hasta su estado actual (Status= {"Abierto", "Asignado", "Re-Abierto", "Resuelto", "Cerrado"}).



Figura 25. Flujo de Estados de Defectos en el Sistema

➤ Tabla de Seguimientos de Estados de Defectos:

ID_Defecto	Asunto	Proyecto	Categoría	Estado
0058	Error en cierre de sesion	SGDBEX	Seguridad	Abierto
0058	Error en cierre de sesion	SGDBEX	Seguridad	Asignado
0058	Error en cierre de sesion	SGDBEX	Seguridad	Resuelto
0083	Error en conexion jdbc BD	SEIBEX	Programacion	Abierto

Figura 26. Prototipo Tabla Seguimientos de Defectos.

Observaciones:

- **Histórico de Defectos:**

En ésta opción se muestra una tabla con todos los datos relacionados a los cambios en un Defecto realizados por un Usuario determinado como se muestra a continuación:

Usuario	Perfil_Usuario	ID_Defecto	Operacion	Campo	Valor_Antiguo	Valor_Actual	Fecha
Alvaro Marciales	Developer	0001	Delete	Prioridad	N/A	Baja	17/10/2014
Rommel Torrez	Developer	0002	Update	Estado	Nuevo	Asignado	21/10/2014
Claudio Torrez	Developer	0003	Insert	Todos	N/A	N/A	03/11/2014
Ana DaSilva	Líder de Proyecto	0004	Update	Estado	Resuelto	Cerrado	05/11/2014

Figura 27. Prototipo Tabla Histórico de Defectos.

Observaciones:

▪ **Reportes:**



Figura 28. Prototipo Menú opción Reportes.

En ésta opción se despliegan los reportes de Defectos que se muestran a continuación:

- Reporte General



Figura 29. Prototipo Menú opción Reportes Desplegable.

❖ Reporte General:

Priorid	ID	Asunto	Descripcion	Proyecto	Categoria	Lider de Proyecto	Archivos adjuntos	Estado
	1	Registro de Usuario	Fallo lectura Username	Sistema de Ticket	Diseño	Ana Dasilva		Cerrado
	2	Bug de Seguridad		DW Corporativo	Calidad	Rommel Torrez		Asignado
	3	Registro de Usuario	Cambiar tipo de dato en el campo CI del cliente	DW Corporativo	Calidad	Rommel Torrez		Asignado
	4	Registro de Usuario	Cambiar fuente de botones	Sistema de Ticket	Seguridad	Ana Dasilva		Re-abierto
	5	Bug Seguridad	Alteracion de permisologias al usuario Admin	DW Corporativo	Seguridad	Rommel Torrez		Cerrado

Figura 30. Prototipo Menú opción Reporte General.

Observaciones:

- **Cerrar Sesión:**



Figura 31. Prototipo Menú opción Cerrar Sesión.

Al presionar el botón “Cerrar Sesión” se ejecuta el logout del Usuario Activo y el sistema re-direcciona a la vista Login nuevamente.

Observaciones:

- **Modelo de casos de uso del sistema**

En esta sección se detallaran los procesos de negocio asociados a los usuarios, funcionalidades y límites del sistema a través del modelo de casos de uso.

- **Casos de Uso (Nivel 0):** En el diagrama 2 se muestra los actores que interactúan con Sistema de Gestión de Defectos a través del nivel 0 de Casos de Uso.

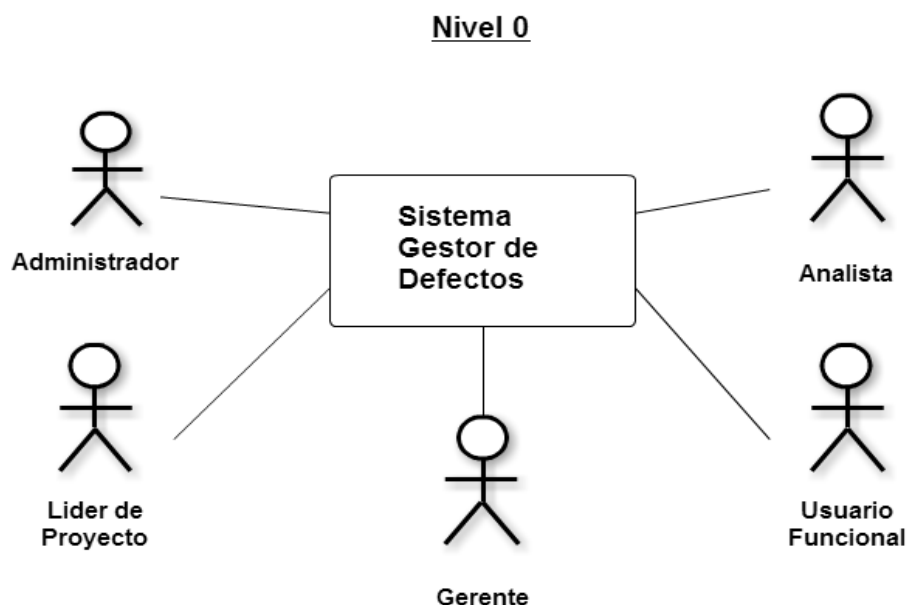


Diagrama 2. Casos de Uso Nivel 0.

- **Casos de Uso (Nivel 1):** En el diagrama 3 se puede visualizar de forma más detallada las interacciones de los usuarios con cada funcionalidad del sistema.

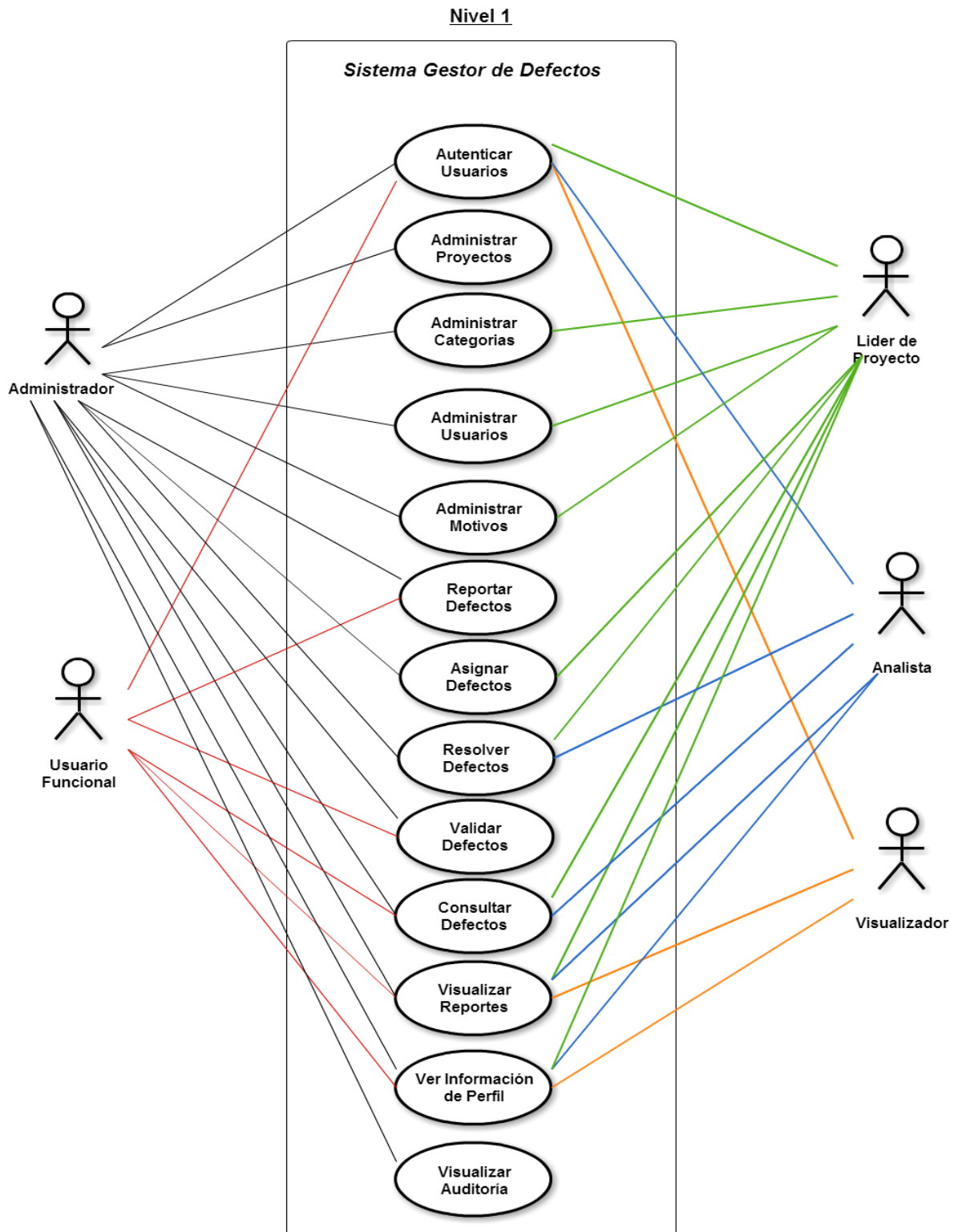


Diagrama 3. Casos de Uso Nivel 1

- **Casos de Uso (Nivel 2):** El diagrama 4 representa los include y extend de cada funcionalidad del sistema.

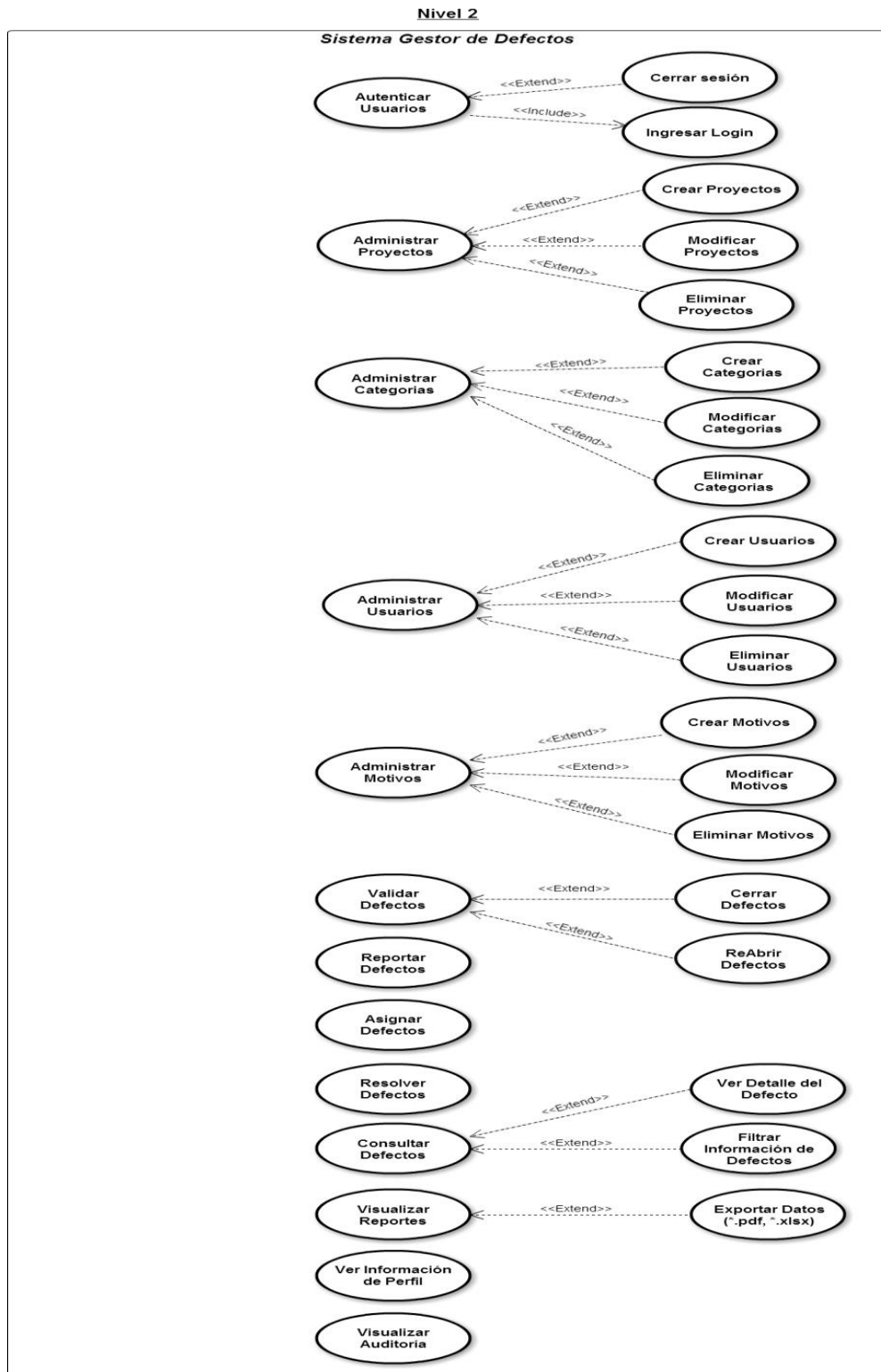


Diagrama 4. Casos de Usos Nivel 2

- Especificación de Casos de Uso:

Caso de Uso:	Autenticar Usuario
Resumen:	El sistema SGDBEX a través de un proceso de verificación, identifica si los datos suministrados en la interfaz 'login.xhtml' son correctos o no.
Actor(es):	Administrador, Líder de Proyectos, Analista, Usuario Funcional y Visualizador.
Precondición:	La cuenta de Usuario ingresada a la Aplicación debe estar almacenada en la BD de Web Services (WS) BEASA para ser válida.
Postcondición:	N/A
Flujo Básico:	
[Actor(es)]	[Sistema]
	1. Presenta la interfaz 'login.xhtml' para ingresar el login y el password.
2. El usuario ingresa los datos de la cuenta que posee en los campos correspondientes.	
	3. El sistema verifica si los datos suministrados por el usuario son válidos respecto a la información de dicha cuenta en el WS BEASA.
	4. Si los datos son válidos, el sistema re-direcciona a la interfaz principal de la Aplicación, en caso contrario re-direcciona a la misma interfaz 'login.xhtml'.
Observaciones:	

Caso de Uso:	Ingresar Login
Resumen:	El sistema muestra la vista 'login.xhtml' para el ingreso de datos de usuario y llevar a cabo el proceso de autenticación.
Actor(es):	Administrador, Líder de Proyectos, Analista, Usuario Funcional y Visualizador.

Precondición:	El usuario debe tener asociada una cuenta con perfil de usuario.
Postcondición:	N/A
Flujo Básico:	
[Actor(es)]	[Sistema]
	1. Presenta la interfaz 'login.xhtml' para ingresar el login y el password.
2. El usuario se encarga de ingresar los datos correspondientes a su cuenta con perfil de usuario.	
	3. El sistema se encarga de leer los datos ingresados por el usuario para su posterior proceso de verificación de cuenta.
Observaciones:	

Caso de Uso:	Cerrar Sesión
Resumen:	El sistema muestra la opción de 'Cerrar Sesión' dentro del menú de la aplicación una vez que el usuario ha sido autenticado de forma exitosa, de tal forma que al culminar sus actividades puede llevar al cabo el cierre de sesión.
Actor(es):	Administrador, Líder de Proyectos, Analista, Usuario Funcional y Visualizador.
Precondición:	El usuario debe tener asociada una cuenta con perfil de usuario válida.
Postcondición:	N/A
Flujo Básico:	
[Actor(es)]	[Sistema]

	1. Presenta el menú principal de la aplicación una vez que el usuario ha sido autenticado de manera satisfactoria presentando la opción de 'Cerrar Sesión'.
2. El usuario puede presionar el botón de cierre de sesión.	
	3. El sistema se encarga de eliminar completamente la sesión del usuario y re-direcciona a la vista 'login.xhtml'.
Observaciones:	

Caso de Uso:	Administrar Proyectos
Resumen:	El sistema contiene el módulo de 'Administración' donde el usuario autenticado (sólo el usuario con rol de Administrador) puede hacer click en la opción 'Administrar Proyectos' y así gestionar los proyectos almacenados en la Base de Datos de la Aplicación, es decir, Agregar, Editar o Eliminar.
Actor(es):	Administrador.
Precondición:	Usuario Autenticado debe tener rol de Administrador.
Postcondición:	Confirmar gestión de Proyectos en el módulo de Administración para que sean almacenados los datos correctamente en Base de Datos.
Flujo Básico:	
[Actor(es)]	[Sistema]
	1. Presenta el menú principal del Sistema.
2. El usuario debe seleccionar la opción 'Administración' del menú principal y luego hacer click en la opción 'Administrar	

Proyectos’.	
	3. Presenta interfaz que posee un Datatable que contiene todos los proyectos activos permitiendo la gestión de Proyectos con los botones correspondientes (Agregar, Editar y Eliminar).
4. El usuario escoge que operación realizar para llevar a cabo la gestión de Proyectos activos.	
Observaciones:	

Caso de Uso:	Crear Proyectos
Resumen:	El sistema contiene el módulo de ‘Administración’ donde el usuario autenticado (sólo el usuario con rol de Administrador) puede hacer click en la opción ‘Administrar Proyectos’ y así poder llevar a cabo el proceso de creación de proyectos.
Actor(es):	Administrador.
Precondición:	Usuario Autenticado debe tener rol de Administrador.
Postcondición:	Confirmar la creación de Proyectos en el módulo de Administración para que sean almacenados los datos correctamente en Base de Datos.
Flujo Básico:	
[Actor(es)]	[Sistema]
	1. Presenta el menú principal del Sistema.
2. El usuario debe seleccionar la opción ‘Administración’ del menú principal y luego hacer click en la opción ‘Administrar	

Proyectos’.	
	3. Presenta interfaz que posee un Datatable que contiene todos los proyectos activos permitiendo la gestión de Proyectos con los botones correspondientes (Agregar, Editar y Eliminar).
4. El usuario escoge la operación ‘Agregar’ para llevar a cabo el proceso de creación de nuevos proyectos activos.	
Observaciones:	

Caso de Uso:	Editar Proyectos
Resumen:	El sistema contiene el módulo de ‘Administración’ donde el usuario autenticado (sólo el usuario con rol de Administrador) puede hacer click en la opción ‘Administrar Proyectos’ y así poder llevar a cabo el proceso de edición de proyectos.
Actor(es):	Administrador.
Precondición:	Usuario Autenticado debe tener rol de Administrador.
Postcondición:	Confirmar la edición de Proyectos en el módulo de Administración para que sean almacenados los datos correctamente en Base de Datos.
<u>Flujo Básico:</u>	
[Actor(es)]	[Sistema]
	1. Presenta el menú principal del Sistema.
2. El usuario debe seleccionar la opción ‘Administración’ del menú principal y luego hacer click en la opción ‘Administrar	

Proyectos’.	
	3. Presenta interfaz que posee Datatable que contiene todos los proyectos activos permitiendo la gestión de Proyectos con los botones correspondientes (Agregar, Editar y Eliminar).
4. El usuario escoge la operación ‘Editar’ para llevar a cabo el proceso de edición de un proyecto seleccionado.	
Observaciones:	

Caso de Uso:	Eliminar Proyectos
Resumen:	El sistema contiene el módulo de ‘Administración’ donde el usuario autenticado (sólo el usuario con rol de Administrador) puede hacer click en la opción ‘Administrar Proyectos’ y así poder llevar a cabo el proceso de eliminación de proyectos.
Actor(es):	Administrador.
Precondición:	Usuario Autenticado debe tener rol de Administrador.
Postcondición:	Confirmar la eliminación de Proyectos en el módulo de Administración para que la Base de Datos tenga información consistente.
<u>Flujo Básico:</u>	
[Actor(es)]	[Sistema]
	1. Presenta el menú principal del Sistema.
2. El usuario debe seleccionar la opción ‘Administración’ del menú principal y luego hacer click en la opción ‘Administrar	

Proyectos’.	
	3. Presenta interfaz que posee un Datatable que contiene todos los proyectos activos permitiendo la gestión de Proyectos con los botones correspondientes (Agregar, Editar y Eliminar).
4. El usuario escoge la operación ‘Eliminar’ para llevar a cabo el proceso de eliminación de proyectos activos.	
Observaciones:	

Caso de Uso:	Administrar Categorías
Resumen:	El sistema contiene el módulo de ‘Administración’ donde el usuario autenticado (sólo el usuario con rol de Administrador o Líder de proyecto) puede hacer click en la opción ‘Administrar Categorías’ y así gestionar las categorías almacenadas en la Base de Datos de la Aplicación, es decir, operaciones como Agregar, Editar o Eliminar.
Actor(es):	Administrador y Líder de Proyecto.
Precondición:	Usuario Autenticado debe tener rol de Administrador o Líder de Proyecto.
Postcondición:	Confirmar gestión de Categorías en el módulo de Administración para que sean almacenados los datos correctamente en Base de Datos.
Flujo Básico:	
[Actor(es)]	[Sistema]
	1. Presenta el menú principal del Sistema.
2. El usuario debe seleccionar la opción ‘Administración’ del menú principal y luego hacer click en la opción ‘Administrar	

Categorías’.	
	3. Presenta interfaz que posee un Datatable con todas las Categorías activas permitiendo la gestión de Categorías con los botones correspondientes (Agregar, Editar y Eliminar).
4. El usuario escoge que operación realizar para llevar a cabo la gestión de Categorías activas.	
Observaciones:	

Caso de Uso:	Crear Categorías
Resumen:	El sistema contiene el módulo de ‘Administración’ donde el usuario autenticado (sólo el usuario con rol de Administrador o Líder de proyecto) puede hacer click en la opción ‘Administrar Categorías’ y así llevar a cabo el proceso de creación de nuevas categorías activas.
Actor(es):	Administrador y Líder de Proyecto.
Precondición:	Usuario Autenticado debe tener rol de Administrador o Líder de Proyecto.
Postcondición:	Confirmar creación de Categorías en el módulo de Administración para que sean almacenados los datos correctamente en Base de Datos.
Flujo Básico:	
[Actor(es)]	[Sistema]
	1. Presenta el menú principal del Sistema.
2. El usuario debe seleccionar la opción ‘Administración’ del menú principal y luego hacer click en la opción ‘Administrar	

Categorías’.	
	3. Presenta interfaz que posee un Datatable con todas las Categorías activas permitiendo la gestión de Categorías con los botones correspondientes (Agregar, Editar y Eliminar).
4. El usuario escoge la operación ‘Agregar’ para llevar a cabo la creación de nuevas Categorías activas.	
Observaciones:	

Caso de Uso:	Editar Categorías
Resumen:	El sistema contiene el módulo de ‘Administración’ donde el usuario autenticado (sólo el usuario con rol de Administrador o Líder de proyecto) puede hacer click en la opción ‘Administrar Categorías’ y así llevar a cabo el proceso de edición de categorías activas.
Actor(es):	Administrador y Líder de Proyecto.
Precondición:	Usuario Autenticado debe tener rol de Administrador o Líder de Proyecto.
Postcondición:	Confirmar edición de Categorías en el módulo de Administración para que sean almacenados los datos correctamente en Base de Datos.
Flujo Básico:	
[Actor(es)]	[Sistema]
	1. Presenta el menú principal del Sistema.
2. El usuario debe seleccionar la opción ‘Administración’ del menú principal y luego hacer click en la opción ‘Administrar	

Categorías’.	
	3. Presenta interfaz que posee un Datatable con todas las Categorías activas permitiendo la gestión de Categorías con los botones correspondientes (Agregar, Editar y Eliminar).
4. El usuario escoge la operación ‘Editar’ sobre una Categoría seleccionada para llevar a cabo la edición de Categorías activas.	
Observaciones:	

Caso de Uso:	Eliminar Categorías
Resumen:	El sistema contiene el módulo de ‘Administración’ donde el usuario autenticado (sólo el usuario con rol de Administrador o Líder de proyecto) puede hacer click en la opción ‘Administrar Categorías’ y así llevar a cabo el proceso de eliminación de categorías activas.
Actor(es):	Administrador y Líder de Proyecto.
Precondición:	Usuario Autenticado debe tener rol de Administrador o Líder de Proyecto.
Postcondición:	Confirmar eliminación de Categorías en el módulo de Administración para que la información almacenada en Base de Datos sea consistente.
Flujo Básico:	
[Actor(es)]	[Sistema]
	1. Presenta el menú principal del Sistema.
2. El usuario debe seleccionar la opción ‘Administración’ del menú principal y luego hacer click en la opción ‘Administrar	

Categorías’.	
	3. Presenta interfaz que posee un Datatable con todas las Categorías activas permitiendo la gestión de Categorías con los botones correspondientes (Agregar, Editar y Eliminar).
4. El usuario escoge la operación ‘Eliminar’ sobre una Categoría seleccionada para llevar a cabo la eliminación de Categorías activas.	
Observaciones:	

Caso de Uso:	Administrar Usuarios
Resumen:	El sistema contiene el módulo de ‘Administración’ donde el usuario autenticado (sólo el usuario con rol de Administrador o Líder de proyecto) puede hacer click en la opción ‘Administrar Usuarios’ y así gestionar los usuarios almacenados en la Base de Datos de la Aplicación, es decir, operaciones como Agregar, Editar o Eliminar.
Actor(es):	Administrador y Líder de Proyecto.
Precondición:	Usuario Autenticado debe tener rol de Administrador o Líder de Proyecto.
Postcondición:	Confirmar gestión de Usuarios en el módulo de Administración para que sean almacenados los datos correctamente en Base de Datos.
Flujo Básico:	
[Actor(es)]	[Sistema]
	1. Presenta el menú principal del Sistema.
2. El usuario debe seleccionar la opción ‘Administración’ del menú principal y luego hacer click en la opción ‘Administrar	

Usuarios’.	
	3. Presenta interfaz que posee un Datatable con todos los usuarios activos permitiendo la gestión de Usuarios con los botones correspondientes (Agregar, Editar y Eliminar).
4. El usuario escoge que operación realizar para llevar a cabo la gestión de Usuarios activos.	
Observaciones:	

Caso de Uso:	Crear Usuarios
Resumen:	El sistema contiene el módulo de ‘Administración’ donde el usuario autenticado (sólo el usuario con rol de Administrador o Líder de proyecto) puede hacer click en la opción ‘Administrar Usuarios’ y así llevar a cabo el proceso de creación de nuevos Usuarios activos.
Actor(es):	Administrador y Líder de Proyecto.
Precondición:	Usuario Autenticado debe tener rol de Administrador o Líder de Proyecto.
Postcondición:	Confirmar creación de Usuarios en el módulo de Administración para que sean almacenados los datos correctamente en Base de Datos.
<u>Flujo Básico:</u>	
[Actor(es)]	[Sistema]
	1. Presenta el menú principal del Sistema.
2. El usuario debe seleccionar la opción ‘Administración’ del menú principal y luego hacer click en la opción ‘Administrar	

Usuarios’.	
	3. Presenta interfaz que posee un Datatable con todos los usuarios activos permitiendo la gestión de Usuarios con los botones correspondientes (Agregar, Editar y Eliminar).
4. El usuario escoge la operación ‘Agregar’ para llevar a cabo la creación de Usuarios activos.	
Observaciones:	

Caso de Uso:	Editar Usuarios
Resumen:	El sistema contiene el módulo de ‘Administración’ donde el usuario autenticado (sólo el usuario con rol de Administrador o Líder de proyecto) puede hacer click en la opción ‘Administrar Usuarios’ y así llevar a cabo el proceso de edición de Usuarios activos.
Actor(es):	Administrador y Líder de Proyecto.
Precondición:	Usuario Autenticado debe tener rol de Administrador o Líder de Proyecto.
Postcondición:	Confirmar edición de Usuarios en el módulo de Administración para que sean almacenados los datos correctamente en Base de Datos.
Flujo Básico:	
[Actor(es)]	[Sistema]
	1. Presenta el menú principal del Sistema.
2. El usuario debe seleccionar la opción ‘Administración’ del menú principal y luego hacer click en la opción ‘Administrar Usuarios’.	
	3. Presenta interfaz que posee un Datatable con todos los usuarios activos permitiendo la gestión de Usuarios con los botones

	correspondientes (Agregar, Editar y Eliminar).
4. El usuario escoge la operación 'Editar' después de haber seleccionado un usuario en particular para llevar a cabo la edición de Usuarios activos.	
Observaciones:	

Caso de Uso:	Eliminar Usuarios
Resumen:	El sistema contiene el módulo de 'Administración' donde el usuario autenticado (sólo el usuario con rol de Administrador o Líder de proyecto) puede hacer click en la opción 'Administrar Usuarios' y así llevar a cabo el proceso de eliminación de Usuarios activos.
Actor(es):	Administrador y Líder de Proyecto.
Precondición:	Usuario Autenticado debe tener rol de Administrador o Líder de Proyecto.
Postcondición:	Confirmar eliminación de Usuarios en el módulo de Administración la información almacenada en la Base de Datos sea consistente.
<u>Flujo Básico:</u>	
[Actor(es)]	[Sistema]
	1. Presenta el menú principal del Sistema.
2. El usuario debe seleccionar la opción 'Administración' del menú principal y luego hacer click en la opción 'Administrar Usuarios'.	
	3. Presenta interfaz que posee un Datatable con todos los usuarios activos permitiendo la gestión de Usuarios con los botones

	correspondientes (Agregar, Editar y Eliminar).
4. El usuario escoge la operación 'Eliminar' después de haber seleccionado un usuario en particular para llevar a cabo la eliminación de Usuarios activos.	
Observaciones:	

Caso de Uso:	Administrar Motivos
Resumen:	El sistema contiene el módulo de 'Administración' donde el usuario autenticado (sólo el usuario con rol de Administrador o Líder de proyecto) puede hacer click en la opción 'Administrar Motivos' y así gestionar los Motivos de Rechazo almacenados en la Base de Datos de la Aplicación, es decir, operaciones como Agregar, Editar o Eliminar.
Actor(es):	Administrador y Líder de Proyecto.
Precondición:	Usuario Autenticado debe tener rol de Administrador o Líder de Proyecto.
Postcondición:	Confirmar gestión de los Motivos de Rechazo en el módulo de Administración para que sean almacenados los datos correctamente en Base de Datos.
<u>Flujo Básico:</u>	
[Actor(es)]	[Sistema]
	1. Presenta el menú principal del Sistema.
2. El usuario debe seleccionar la opción 'Administración' del menú principal y luego hacer click en la opción 'Administrar Motivos'.	
	3. Presenta interfaz que posee un Datatable que contiene todos los motivos de rechazo permitiendo la gestión de los Motivos de

	Rechazo con los botones correspondientes (Agregar, Editar y Eliminar).
4. El usuario escoge que operación realizar para llevar a cabo la gestión de los Motivos de Rechazo activos.	
Observaciones:	

Caso de Uso:	Crear Motivos
Resumen:	El sistema contiene el módulo de 'Administración' donde el usuario autenticado (sólo el usuario con rol de Administrador o Líder de proyecto) puede hacer click en la opción 'Administrar Motivos' y así llevar a cabo el proceso de creación de motivos activos.
Actor(es):	Administrador y Líder de Proyecto.
Precondición:	Usuario Autenticado debe tener rol de Administrador o Líder de Proyecto.
Postcondición:	Confirmar creación de los Motivos de Rechazo en el módulo de Administración para que sean almacenados los datos correctamente en Base de Datos.
Flujo Básico:	
[Actor(es)]	[Sistema]
	1. Presenta el menú principal del Sistema.
2. El usuario debe seleccionar la opción 'Administración' del menú principal y luego hacer click en la opción 'Administrar Motivos'.	
	3. Presenta interfaz que posee un Datatable que contiene todos los motivos de rechazo permitiendo la gestión de los Motivos de

	Rechazo con los botones correspondientes (Agregar, Editar y Eliminar).
4. El usuario escoge la operación 'Agregar' para llevar a cabo la creación de los Motivos de Rechazo activos.	
Observaciones:	

Caso de Uso:	Editar Motivos
Resumen:	El sistema contiene el módulo de 'Administración' donde el usuario autenticado (sólo el usuario con rol de Administrador o Líder de proyecto) puede hacer click en la opción 'Administrar Motivos' y así llevar a cabo el proceso de edición de motivos activos.
Actor(es):	Administrador y Líder de Proyecto.
Precondición:	Usuario Autenticado debe tener rol de Administrador o Líder de Proyecto.
Postcondición:	Confirmar edición de los Motivos de Rechazo en el módulo de Administración para que sean almacenados los datos correctamente en Base de Datos.
Flujo Básico:	
[Actor(es)]	[Sistema]
	1. Presenta el menú principal del Sistema.
2. El usuario debe seleccionar la opción 'Administración' del menú principal y luego hacer click en la opción 'Administrar Motivos'.	
	3. Presenta interfaz que posee un Datatable que contiene todos los motivos de rechazo permitiendo la gestión de los Motivos de Rechazo con los botones correspondientes (Agregar, Editar y Eliminar).

<p>4. El usuario escoge la operación 'Editar' de un motivo seleccionado para llevar a cabo la edición de los Motivos de Rechazo activos.</p>	
<p>Observaciones:</p>	

<p>Caso de Uso:</p>	<p>Eliminar Motivos</p>
<p>Resumen:</p>	<p>El sistema contiene el módulo de 'Administración' donde el usuario autenticado (sólo el usuario con rol de Administrador o Líder de proyecto) puede hacer click en la opción 'Administrar Motivos' y así llevar a cabo el proceso de eliminación de motivos activos.</p>
<p>Actor(es):</p>	<p>Administrador y Líder de Proyecto.</p>
<p>Precondición:</p>	<p>Usuario Autenticado debe tener rol de Administrador o Líder de Proyecto.</p>
<p>Postcondición:</p>	<p>Confirmar eliminación de los Motivos de Rechazo en el módulo de Administración para que la información en la Base de Datos sea consistente.</p>
<p>Flujo Básico:</p>	
<p>[Actor(es)]</p>	<p>[Sistema]</p>
	<p>1. Presenta el menú principal del Sistema.</p>
<p>2. El usuario debe seleccionar la opción 'Administración' del menú principal y luego hacer click en la opción 'Administrar Motivos'.</p>	
	<p>3. Presenta interfaz que posee un Datatable que contiene todos los motivos de rechazo permitiendo la gestión de los Motivos de Rechazo con los botones correspondientes</p>

	(Agregar, Editar y Eliminar).
4. El usuario escoge la operación 'Eliminar' de un motivo seleccionado para llevar a cabo la eliminación de los Motivos de Rechazo activos.	
Observaciones:	

Caso de Uso:	Reportar Defectos
Resumen:	El sistema a través del menú principal provee la opción de 'Reportar Defecto', por lo cual sólo los usuarios con privilegios de ejecutar dicha opción pueden reportar un defecto correspondiente a un proyecto en particular y éste sea definido con estado 'Abierto' para su postrera asignación.
Actor(es):	Administrador y Usuario Funcional.
Precondición:	Usuario Autenticado debe tener rol de Administrador o Usuario Funcional.
Postcondición:	El usuario debe asegurarse de haber llenado correctamente los campos del formulario para registrar defectos para que sean creados y almacenados en Base de Datos de forma correcta.
<u>Flujo Básico:</u>	
[Actor(es)]	[Sistema]
	1. Presenta la interfaz principal con el menú en la parte superior, el cual posee la opción 'Reportar Defecto'.
2. El usuario selecciona la opción 'Reportar Defecto' desde el menú de opciones desplegable en la interfaz principal.	

	3. Muestra una nueva vista con el formulario que debe llenar el usuario para realizar el reporte del defecto a registrar.
4. El usuario llena los campos necesarios del formulario para reportar el defecto y presiona el botón 'Crear'.	
	5. Presenta un mensaje de diálogo pidiendo confirmación al usuario para crear el defecto de forma correcta.
6. El usuario decide si conformar la creación del defecto o rechazarla dicha creación.	
Observaciones:	

Caso de Uso:	Asignar Defectos
Resumen:	El sistema a través del menú principal provee la opción de 'Asignar Defecto', por lo cual sólo los usuarios con privilegios de ejecutar dicha opción pueden asignar uno o más defectos en específico correspondiente a un proyecto en particular y éste o estos defectos que se encuentran en estado Abierto estén listos para su postrera asignación.
Actor(es):	Administrador y Líder de Proyecto.
Precondición:	-Usuario Autenticado debe tener rol de Administrador o Líder de Proyecto. -Debe existir previamente un defecto en específico para llevar a cabo cierta asignación.
Postcondición:	El sistema enviará una notificación vía correo al usuario que le fue asignado dicho defecto creado.
Flujo Básico:	
[Actor(es)]	[Sistema]

	1. Presenta en el menú principal la opción 'Control de Defectos' el cual posee la opción 'Asignar Defectos'.
2. El usuario selecciona la opción 'Asignar Defectos' desde el menú principal a través de la opción desplegable 'Control de Defectos'.	
	3. Presenta la vista que muestra el conjunto de defectos en estado Abierto para que sean asignados a un usuario en particular.
4. El usuario asigna el defecto a otro usuario con rol Analista, Líder de Proyecto o Administrador el cual será el responsable de resolver dicho defecto asignado.	
Observaciones:	

Caso de Uso:	Resolver Defectos
Resumen:	El sistema a través del menú principal provee la opción de 'Resolver Defecto', por lo cual sólo los usuarios con privilegios de ejecutar dicha opción pueden llevar a cabo la resolución de los defectos en la que se encuentran involucrados y son responsables de dicha resolución correspondiente a un proyecto en particular y éste o estos defectos que se encuentran en estado Asignado estén listos para su postrera validación.
Actor(es):	Administrador, Líder de Proyecto y Analista.
Precondición:	Usuario Autenticado debe tener rol de Administrador, Líder de Proyecto o Analista.
Postcondición:	El sistema enviará una notificación vía correo al

	usuario con rol Usuario Funcional o Administrador para su posible validación (estado Cerrado del defecto).
Flujo Básico:	
[Actor(es)]	[Sistema]
	1. Presenta en el menú principal la opción 'Control de Defectos' el cual posee la opción 'Resolver Defectos'.
2. El usuario selecciona la opción 'Resolver Defectos' desde el menú principal a través de la opción desplegable 'Control de Defectos'.	
	3. Presenta la vista que muestra el conjunto de defectos en estado Asignado (defectos asignados al usuario autenticado) para que sean resueltos por el mismo usuario ya que es el responsable.
4. El usuario autenticado lleva a cabo el proceso de resolución del o los defectos asignados a él mismo a través del llenado del formulario que presenta la vista de resolución, y así el defecto quede a la espera de su postrera validación.	
Observaciones:	

Caso de Uso:	Validar Defectos
--------------	------------------

Resumen:	El sistema a través del menú principal provee la opción de 'Control de Defectos' el cual posee la opción 'Validar Defectos', ésta segunda opción permite mostrar otra vista que posee 2 opciones más que son Validar y Rechazar brindando así la libertad al usuario que tenga estos privilegios de escoger entre Validar (Cerrar Defecto) o Rechazar (Re-Abrir Defecto).
Actor(es):	Administrador y Usuario Funcional.
Precondición:	Usuario Autenticado debe tener rol de Administrador o Usuario Funcional.
Postcondición:	El Defecto que haya sido Validado cambiará a estado Cerrado en caso de que el usuario haya escogido la opción Validar, en caso de que el defecto haya sido rechazado, cambiará automáticamente a estado Re-Abierto para su posterior Asignación y Resolución.
<u>Flujo Básico:</u>	
[Actor(es)]	[Sistema]
	1. Presenta en el menú principal la opción 'Control de Defectos' el cual posee la opción Validar Defectos'.
2. El usuario selecciona la opción 'Validar Defectos' desde el menú principal a través de la opción desplegable 'Control de Defectos'.	
	3. Presenta la vista que muestra el conjunto de defectos en estado Asignado o Resuelto, la cual posee 2 opciones adicionales que son Validar y Rechazar, dando al usuario la libertad de seleccionar la opción a ejecutar según su preferencia.
4. Una vez presentada la vista 'Validar Defectos', el usuario autenticado puede escoger si Validar o Rechazar el Defecto seleccionado.	

	5. Dependiendo de la opción que elija el usuario, el sistema tomará las acciones correspondientes a dicha elección.
Observaciones:	

Caso de Uso:	Re-Abrir Defectos
Resumen:	El sistema a través del menú principal provee la opción de 'Control de Defectos' el cual posee la opción 'Validar Defectos', ésta segunda opción permite mostrar otra vista que posee 2 opciones más que son Validar y Rechazar y el Defecto cambia a estado Re-Abierto cuando se presiona el botón 'Rechazar', por lo cual sólo los usuarios con privilegios de ejecutar dicha opción pueden llevar a cabo la Validación de defectos correspondiente a un proyecto en particular para que de esta forma éste o estos defectos que se encuentran en estado Asignado o Resuelto estén listos para su postrera asignación ya que estarían definidos con estado Re-Abierto..
Actor(es):	Administrador y Usuario Funcional.
Precondición:	-Usuario Autenticado debe tener rol de Administrador o Usuario Funcional. -Debe estar existentes uno o varios defectos en estado Resuelto o Asignado para su posible Re-Apertura.
Postcondición:	El defecto que estuvo en estado Resuelto y pasó a estado Re-Abierto será cambiado a estado Asignado nuevamente debido a su Re-Apertura.
Flujo Básico:	
[Actor(es)]	[Sistema]
	1. Presenta en el menú principal la opción 'Control de Defectos' el cual posee la opción

	Validar Defectos’.
2. El usuario selecciona la opción ‘Validar Defectos’ desde el menú principal a través de la opción desplegable ‘Control de Defectos’.	
	3. Presenta la vista que muestra el conjunto de defectos en estado Asignado o Resuelto, la cual posee 2 opciones que son Validar y Rechazar, al presionar el botón Rechazar, a un Defecto en particular, éste será cambiado de estado Asignado o Resuelto a Re-Abierto, para que de ésta forma el Defecto quede en Reapertura.
4. Una vez presentada la vista ‘Validar Defectos’, el usuario autenticado selecciona un defecto en estado Asignado o Resuelto desde la información que arroja el sistema.	
5. El usuario selecciona la opción ‘Rechazar’ para que el defecto seleccionado cambie a estado ‘Re-Abierto’, y así quede a la espera de su postrera asignación, repitiendo de ésta forma el flujo de estados que tiene definido el sistema.	
Observaciones:	

Caso de Uso:	Cerrar Defectos
Resumen:	El sistema a través del menú principal provee la opción de ‘Control de Defectos’ el cual posee la opción ‘Validar Defectos’, ésta segunda opción permite mostrar otra vista que posee 2

	<p>opciones más que son Validar y Rechazar y el Defecto cambia a estado Cerrado cuando se presiona el botón 'Validar', por lo cual sólo los usuarios con privilegios de ejecutar dicha opción pueden llevar a cabo la Validación de defectos correspondiente a un proyecto en particular para que de esta forma éste o estos defectos que se encuentran en estado Asignado o Resuelto queden Cerrados de manera definitiva.</p>
Actor(es):	Administrador y Usuario Funcional.
Precondición:	Usuario Autenticado debe tener rol de Administrador o Usuario Funcional.
Postcondición:	El Defecto que haya sido Validado cambiará a estado Cerrado y no podrá ser editado ya que no requiere de cambios postreros.
<u>Flujo Básico:</u>	
[Actor(es)]	[Sistema]
	1. Presenta en el menú principal la opción 'Control de Defectos' el cual posee la opción Validar Defectos'.
2. El usuario selecciona la opción 'Validar Defectos' desde el menú principal a través de la opción desplegable 'Control de Defectos'.	
	3. Presenta la vista que muestra el conjunto de defectos en estado Asignado o Resuelto, la cual posee 2 opciones que son Validar y Rechazar, al presionar el botón Validar, a un Defecto en particular, éste será cambiado de estado Asignado o Resuelto a Cerrado, para que de ésta forma el Defecto quede definido de manera estática y no pueda ser modificado ya que el caso se cerró.
4. Una vez presentada la vista 'Validar Defectos', el usuario autenticado selecciona	

un defecto en estado Asignado o Resuelto desde la información que arroja el sistema.	
5. El usuario selecciona la opción 'Validar' para que el defecto seleccionado cambie a estado 'Cerrado', y así quede cerrado totalmente y definitivamente el caso relacionado al Defecto.	
Observaciones:	

Caso de Uso:	Consultar Defectos
Resumen:	El sistema a través del menú principal provee la opción de 'Búsqueda', la cual recibe el id de un defecto en particular para así mostrar el resumen o detalle de toda la información relacionada a dicho defecto.
Actor(es):	Administrador, Líder de Proyecto, Analista y Usuario Funcional.
Precondición:	Usuario Autenticado debe tener rol de Administrador, Líder de Proyecto, Analista o Usuario Funcional.
Postcondición:	N/A
Flujo Básico:	
[Actor(es)]	[Sistema]
	1. Presenta en el menú principal la opción de 'Búsqueda'.
2. El usuario suministra en la opción de 'Búsqueda' el id de un defecto en particular.	
	3. Se encarga de realizar el proceso de consulta y filtrado de datos en base al id suministrado por el usuario para así mostrar toda la información de tipo resumen o detallada de dicho defecto.

4. El usuario puede visualizar la información resumen o detallada que retorna la opción de 'Búsqueda'.	
Observaciones:	

Caso de Uso:	Ver Detalle del Defecto
Resumen:	Una vez que el sistema presenta los Datatables con los datos relacionados a los defectos, dichas tablas poseen una columna con el ID del defecto el cual es un tipo de CommandLink que al hacer click sobre dicho enlace, el sistema redirecciona a otra vista la cual muestra en forma más detallada información asociada a un defecto en particular.
Actor(es):	Administrador, Líder de Proyecto, Analista y Usuario Funcional.
Precondición:	Usuario Autenticado debe tener rol de Administrador, Líder de Proyecto, Analista o Usuario Funcional.
Postcondición:	N/A
Flujo Básico:	
[Actor(es)]	[Sistema]
	1. Presenta un Datatable con información referente a los defectos en la etapa de seleccionar opción 'Consultar Defecto' como vista resumen.
2. El usuario selecciona algún defecto en particular y hace click sobre el ID de dicho defecto que es de tipo CommandLink para visualizar información más detallada sobre dicho defecto seleccionado.	

	3. Una vez que se ejecuta por parte del usuario click sobre el ID correspondiente a un defecto, el sistema re-direcciona a otra vista donde se muestra información en detalle del defecto seleccionado.
4. El usuario puede visualizar la información detallada del defecto seleccionado.	
Observaciones:	

Caso de Uso:	Filtrar información de Defectos
Resumen:	Una vez que el sistema presenta los Datatables con la información relacionada a los defectos de forma resumida, dichas tablas poseen por cada columna filtros de búsqueda para facilitar al usuario los defectos que él desea visualizar según sus intereses y no se presente la necesidad de tener que visualizar todos los defectos en general si el usuario solo desea ver un subconjunto de ellos.
Actor(es):	Administrador, Líder de Proyecto, Analista y Usuario Funcional.
Precondición:	Usuario Autenticado debe tener rol de Administrador, Líder de Proyecto, Analista o Usuario Funcional.
Postcondición:	N/A
<u>Flujo Básico:</u>	
[Actor(es)]	[Sistema]
	1. Presenta un Datatable con información referente a los defectos en la etapa de

	seleccionar opción 'Consultar Defecto' como vista resumen, los cuales muestran por cada columna filtros según el tipo de dato cargado por cada columna.
2. El usuario ingresa los valores que considere necesarios en un cualquiera de los filtros que sean de su interés o preferencia.	
	3. Una vez que los filtros son cargados con datos, el sistema se encarga de ejecutar dicho filtro sobre cada fila de la tabla y así ir descartando los datos que no están relacionados a los valores cargados en los filtros.
4. El usuario puede visualizar la información referente a los defectos filtrados según el interés del usuario.	
Observaciones:	

Caso de Uso:	Visualizar Reportes
Resumen:	El sistema a través del menú principal provee la opción de 'Reportes de Defectos', la cual posee otra opción llamada 'Reporte General' que va vinculada a una vista que posee todos los tipos de reportes definidos con filtros y alternativas para exportar la información que se muestra al usuario.
Actor(es):	Administrador, Líder de Proyecto, Analista, Usuario Funcional y Visualizador.
Precondición:	Usuario Autenticado debe tener cualquier tipo de rol asignado.
Postcondición:	N/A
Flujo Básico:	
[Actor(es)]	[Sistema]

	1. Presenta en el menú principal la opción 'Reportes de Defectos' el cual posee la opción 'Reporte General'.
2. El usuario selecciona la opción 'Reporte General' desde el menú principal a través de la opción desplegable 'Reportes de Defectos'.	
	3. Presenta la vista que muestra el conjunto de reportes y gráficos para que el usuario pueda visualizar la información que sea de su interés.
4. El usuario escoge que tipo de reporte o gráfico desea observar de acuerdo a sus preferencias, lo cual le facilita saber estadísticas relacionadas a los defectos que posee un proyecto en particular según el tipo de Reporte.	
Observaciones:	

Caso de Uso:	Exportar Datos (*.pdf, *.xlsx)
Resumen:	En el área de Reportes, el sistema permite exportar toda la información que poseen los Datatables en relación a los defectos por cada tipo de Reporte en formato PDF (*.pdf) y en formato Excel (*.xlsx).
Actor(es):	Administrador, Líder de Proyecto, Analista, Usuario Funcional y Visualizador.
Precondición:	Usuario Autenticado debe tener cualquier tipo de rol asignado.
Postcondición:	N/A
Flujo Básico:	
[Actor(es)]	[Sistema]
	1. Presenta en el menú principal la opción 'Reportes de Defectos' el cual posee la opción

	'Reporte General'.
2. El usuario selecciona la opción 'Reporte General' desde el menú principal a través de la opción desplegable 'Reportes de Defectos'.	
	3. Presenta la vista que muestra el conjunto de reportes y gráficos para que el usuario pueda visualizar la información que sea de su interés conjuntamente con 2 opciones para exportar los datos que poseen dichos Datatables (formato PDF y Excel).
4. El usuario escoge que tipo de reporte o gráfico desea observar de acuerdo a sus preferencias y tomar la decisión de exportar o no la información en formato PDF o Excel.	
Observaciones:	

Caso de Uso:	Ver Información de Perfil
Resumen:	El sistema a través del menú principal provee la opción 'Mi Cuenta', la cual se encarga de mostrar al usuario autenticado en la aplicación una breve información acerca de su perfil de cuenta de usuario.
Actor(es):	Administrador, Líder de Proyecto, Analista, Usuario Funcional y Visualizador.
Precondición:	Usuario Autenticado debe tener cualquier tipo de rol asignado.
Postcondición:	N/A
Flujo Básico:	
[Actor(es)]	[Sistema]
	1. Presenta en el menú principal la opción 'Mi Cuenta' la cual presenta la vista que contiene la

	información de perfil de cuenta de usuario autenticado.
2. El usuario selecciona la opción 'Mi Cuenta' desde el menú principal.	
	3. Presenta la vista que muestra correspondiente a la información de perfil de usuario autenticado.
4. El usuario puede visualizar una breve descripción de su perfil de cuenta de usuario.	
Observaciones:	

Caso de Uso:	Visualizar Auditoría
Resumen:	El sistema en proceso de background provee la opción de 'Ver Auditoría', la cual se encarga de mostrar información referente a la auditoría y seguimiento general de los defectos durante su ciclo de vida en la aplicación.
Actor(es):	Administrador.
Precondición:	Usuario Autenticado debe tener rol 'Administrador'.
Postcondición:	N/A
Flujo Básico:	
[Actor(es)]	[Sistema]
	1. Presenta en el menú principal la opción 'Ver Auditoría' la cual re-direcciona a otra vista donde se muestra información referente a dicha opción.
2. El usuario selecciona la opción 'Ver Auditoría' desde el menú principal para que sea renderizada la vista de auditoría.	
	3. Presenta la vista de auditoría la cual muestra

	información referente al seguimiento de actividades y cambios en los defectos durante su ciclo de vida.
4. El usuario puede visualizar información referente al seguimiento de defectos.	
Observaciones:	

2.1.2. FASE DE ELABORACION Y DISEÑO

De acuerdo a la metodología RUP, para analizar los requisitos y las responsabilidades sobre el sistema, se recomienda el uso de diagramas de colaboración. Estos diagramas representan el modelo de análisis del sistema y expondrán sus funcionalidades.

- **Diagramas de Colaboración:** El diagrama de colaboración muestra una interacción organizada basándose en los objetos que toman parte en la interacción y los enlaces entre los mismos.

En el diagrama 5 se muestra los pasos a seguir para el proceso de autenticación de usuarios en el sistema a través de métodos entre objetos.

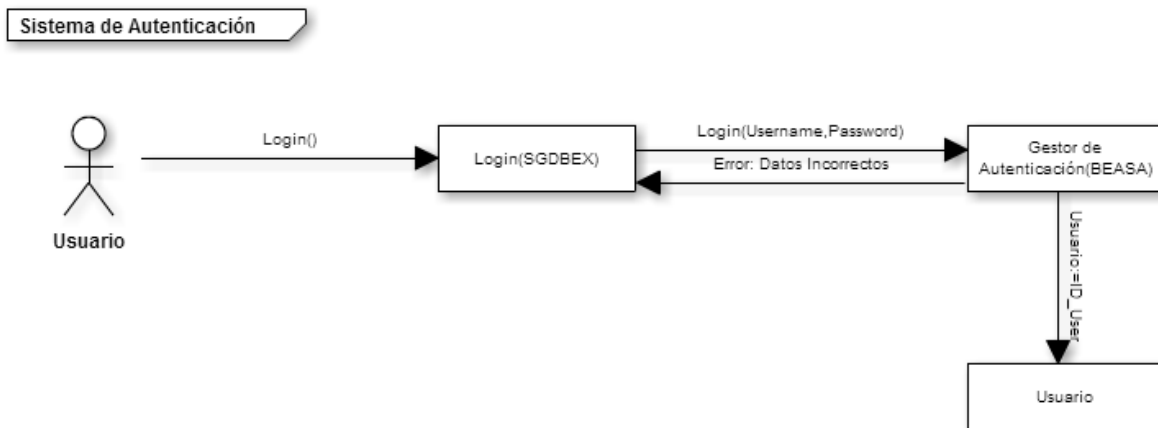


Diagrama 5 .Colaboración Sistema de Autenticación

En el diagrama 6 se muestra la secuencia de llamadas entre objetos para el registro de defectos en el sistema.

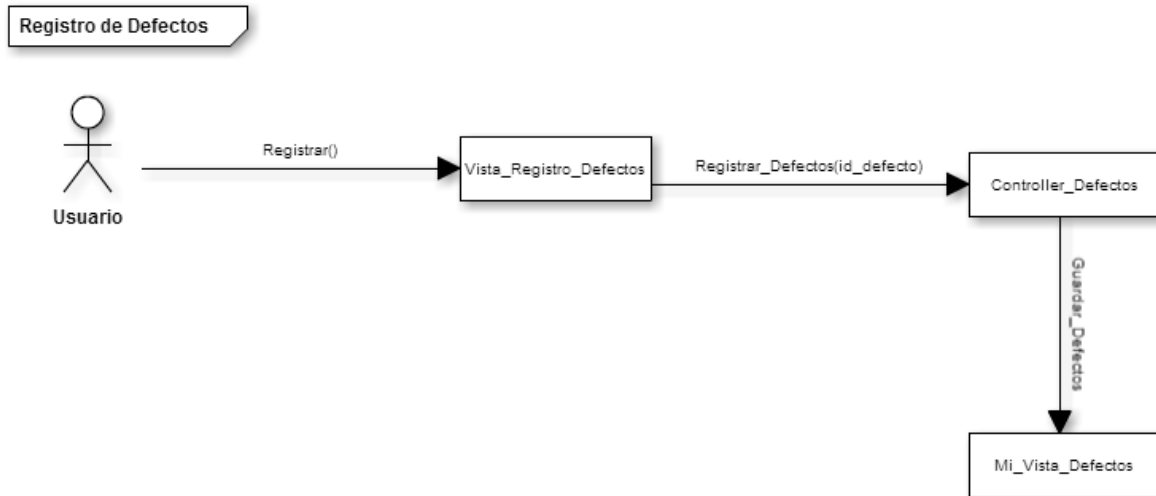


Diagrama 6. Colaboración Registro de Defectos

El diagrama 7 representa la secuencia de llamadas a métodos que interactúan entre objetos para llevar a cabo la consulta de defectos.

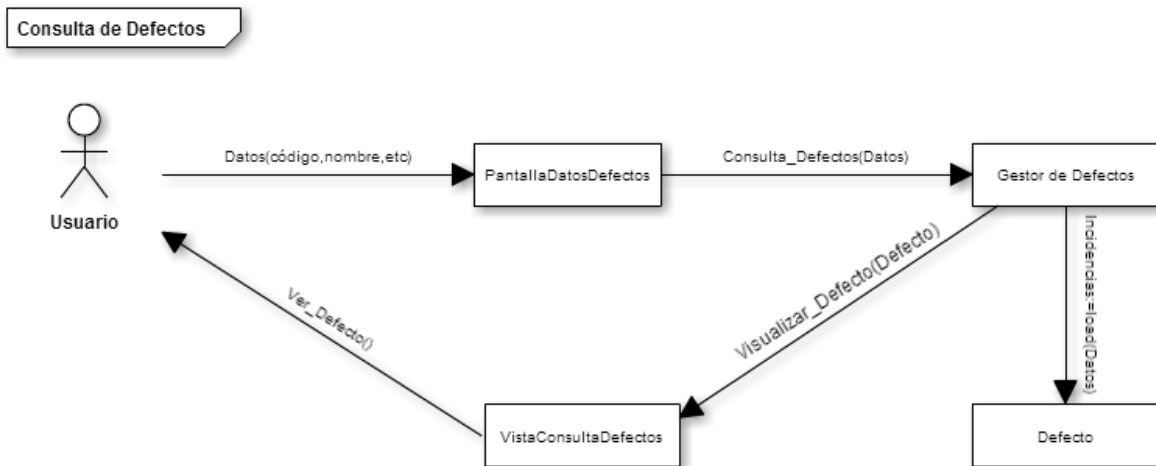


Diagrama 7. Colaboración Consulta de Defectos

El diagrama 8 consiste en mostrar la serie de invocaciones a métodos para que un usuario pueda asignar un defecto siempre y cuando posea los privilegios para realizar dicha funcionalidad.

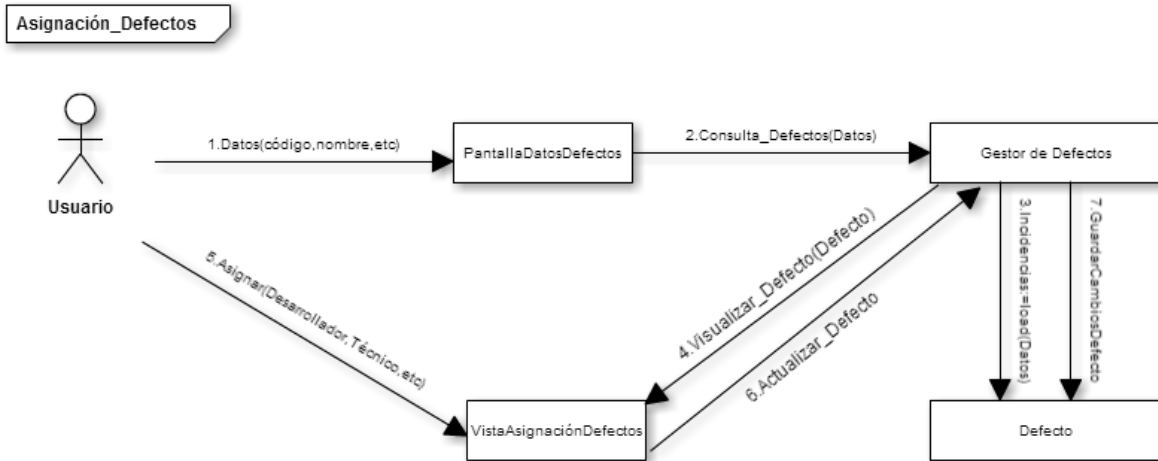


Diagrama 8. Colaboración Asignación de Defectos.

El diagrama 9 se encarga de mostrar la secuencia de llamadas a métodos enumeradas cuando un usuario interviene o modifica un defecto que corresponde a un proyecto en particular.

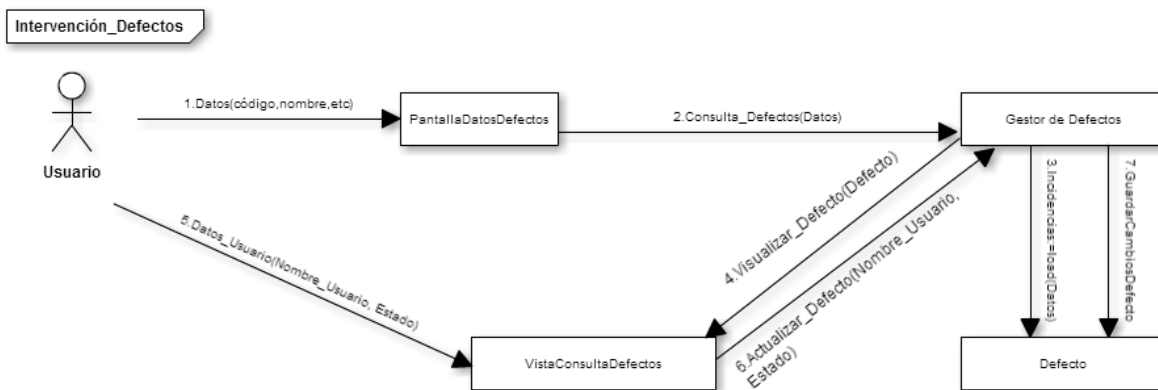


Diagrama 9. Colaboración Intervención de Defectos

- **Diagramas de Estados**

A continuación en el diagrama 10, se muestra los estados para expresar de una manera más clara las fases por las cuales pasa un defecto a través de su ciclo

de vida.

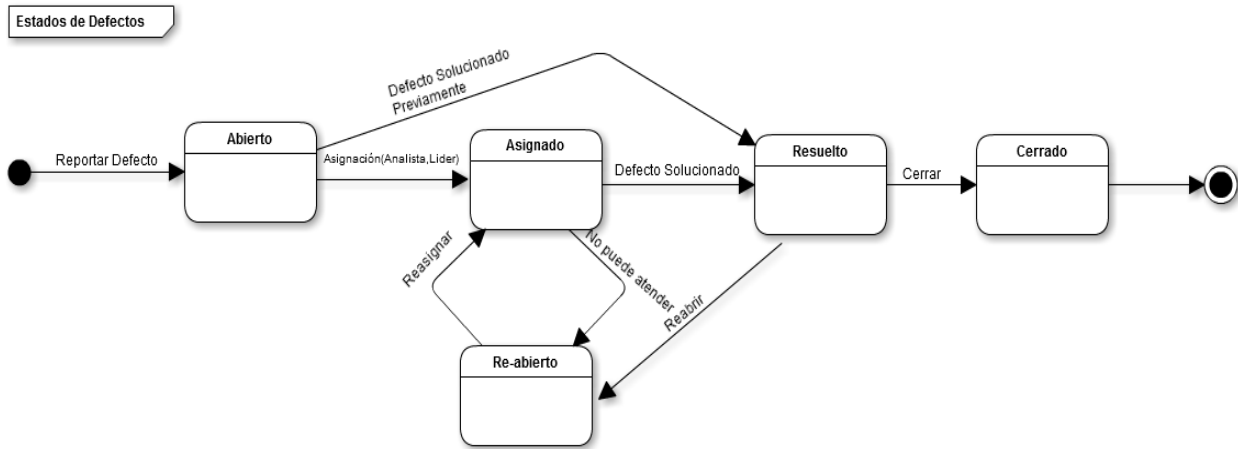


Diagrama 10. Diagrama de Estados de Defectos

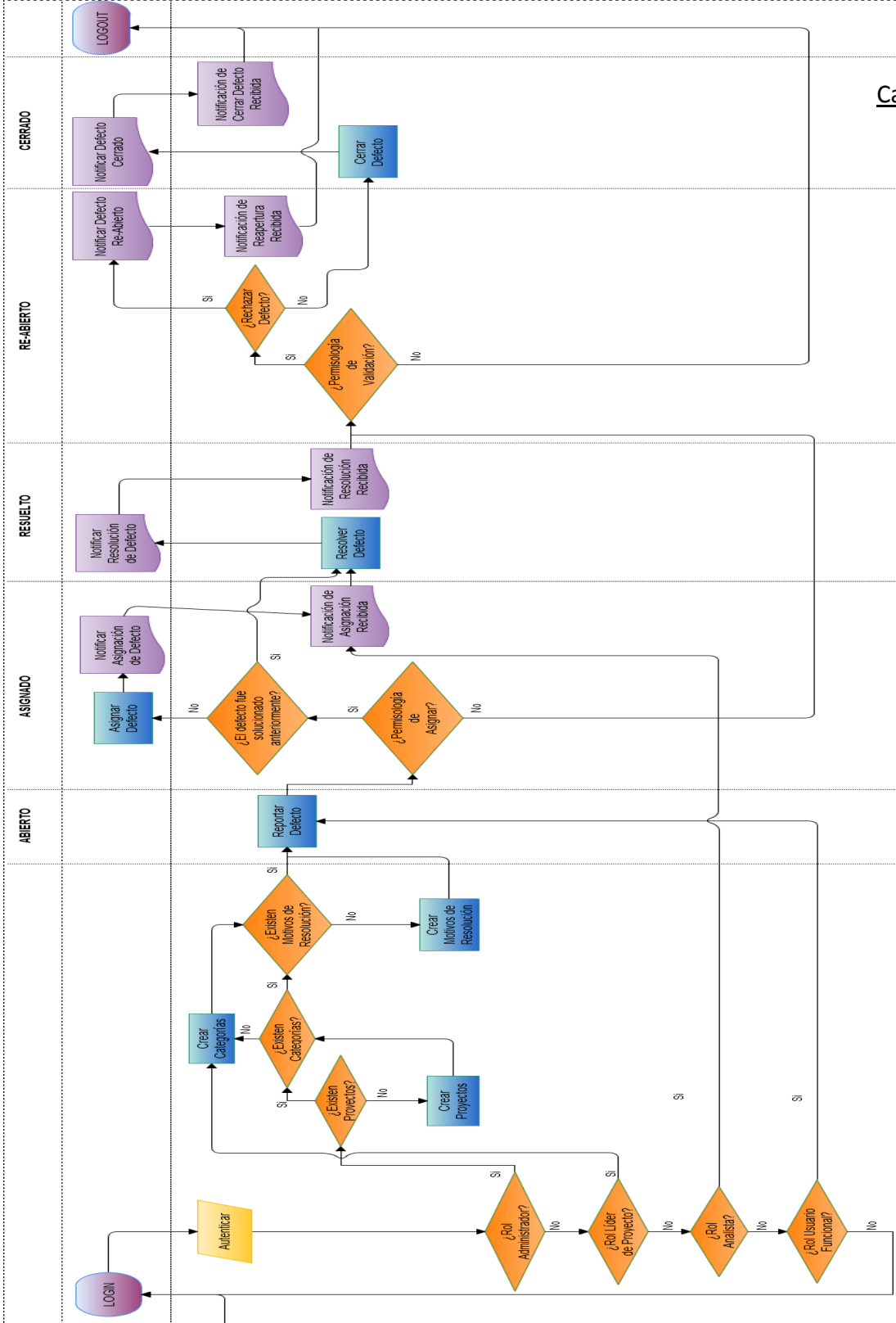
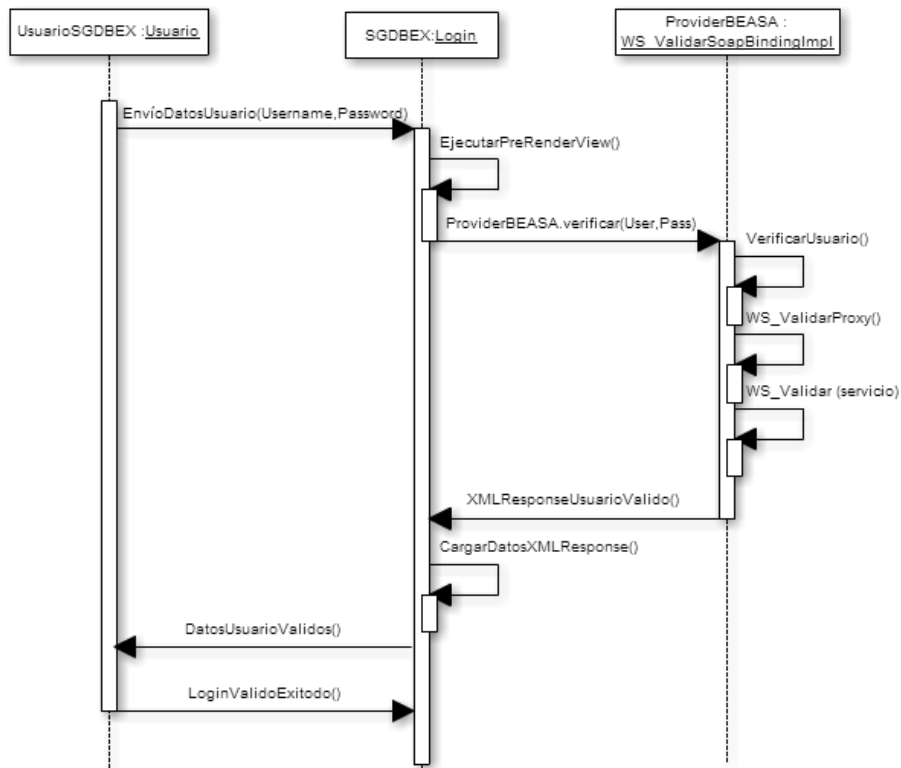


Diagrama 11. Diagrama de Flujo del ciclo de vida del Defecto.

• **Diagramas de Secuencia:**

El diagrama 12 representa el pase de mensajes entre objetos durante su ciclo de vida para la autenticación de usuarios.

SQ Autenticación de Usuario



Nota: El ProviderBEASA es parte del Web Services que consume el Sistema a través de una instancia para la autenticación de cualquier usuario antes de ingresar a la Aplicación.

Diagrama 12. Secuencia Autenticación de Usuario

El diagrama 13 se puede visualizar la secuencia de pase de mensajes entre objetos para crear una categoría.

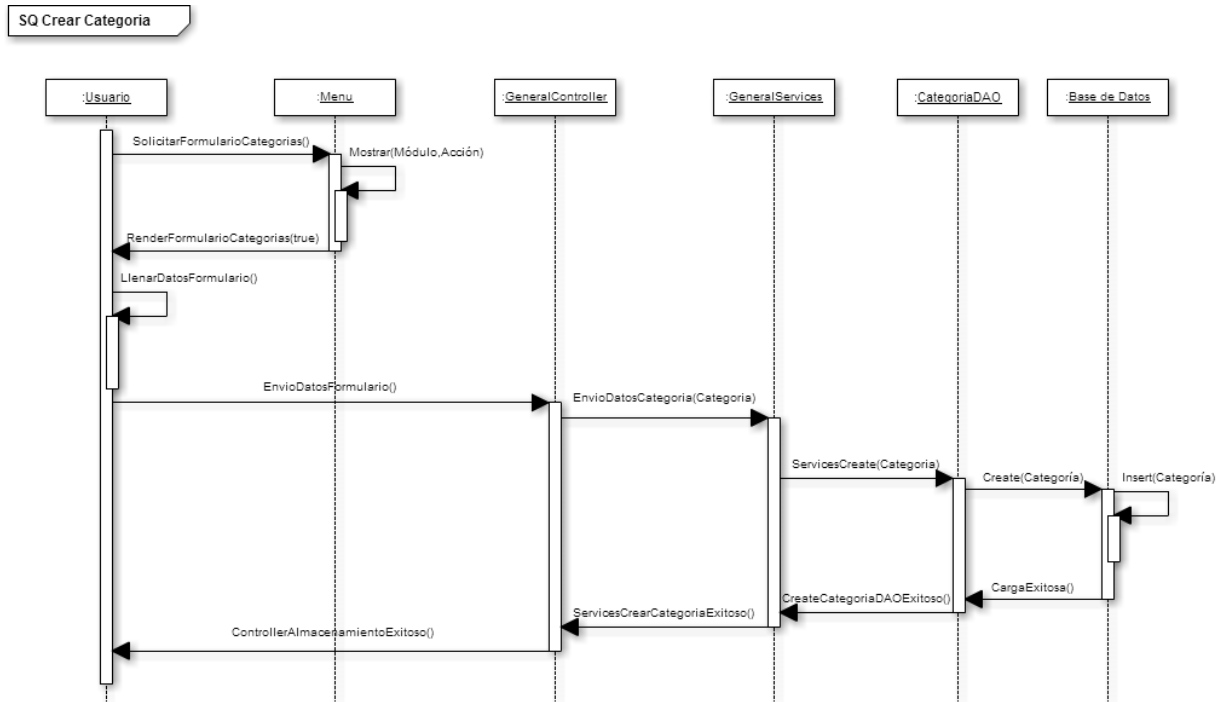


Diagrama 13. Secuencia Crear Categoría

El diagrama 14 se puede visualizar la secuencia de pase de mensajes entre objetos para modificar una categoría.

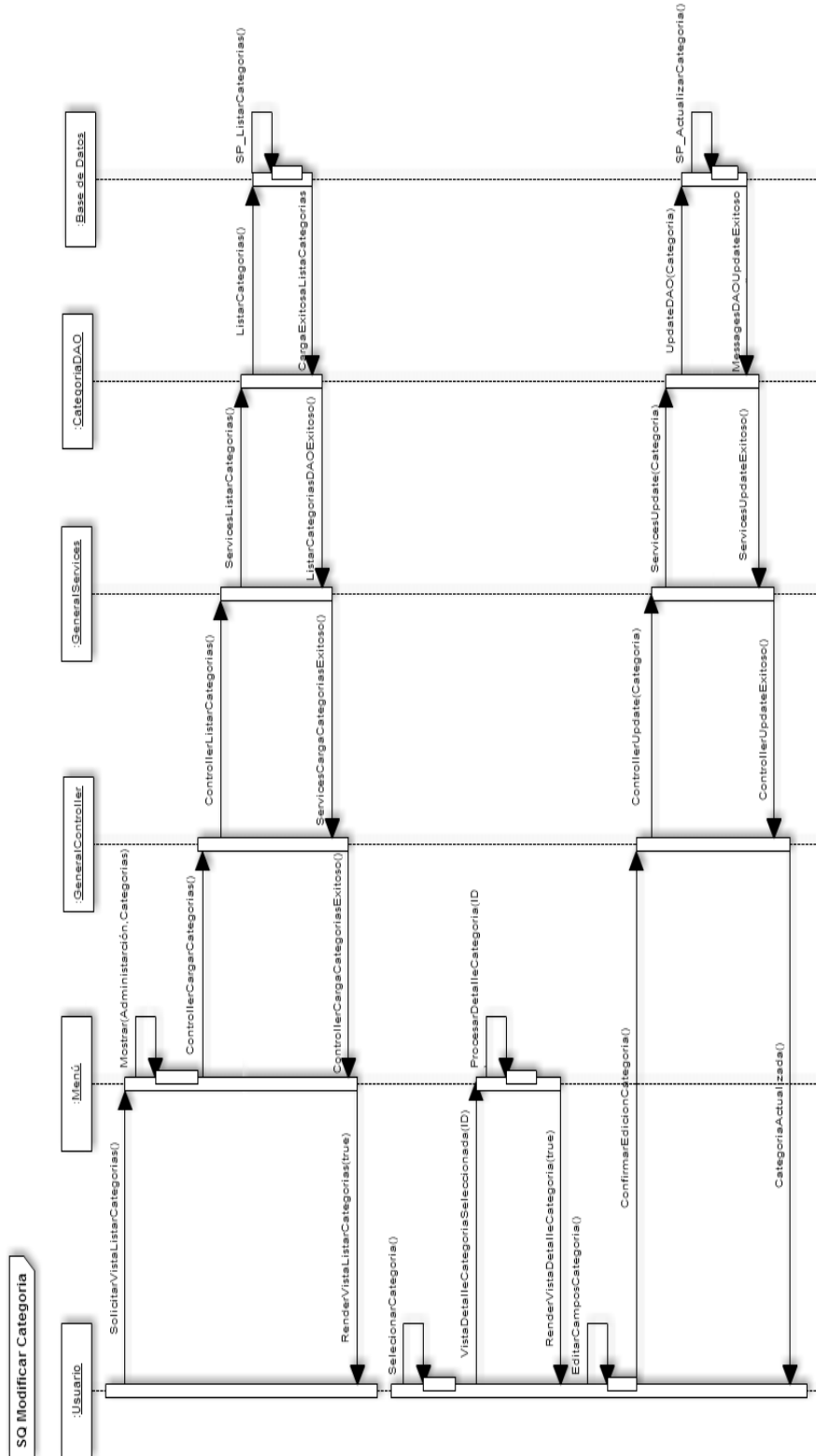


Diagrama 14. Secuencia Modificar Categoría

El diagrama 15 se puede visualizar la secuencia de pase de mensajes entre objetos para eliminar una categoría.

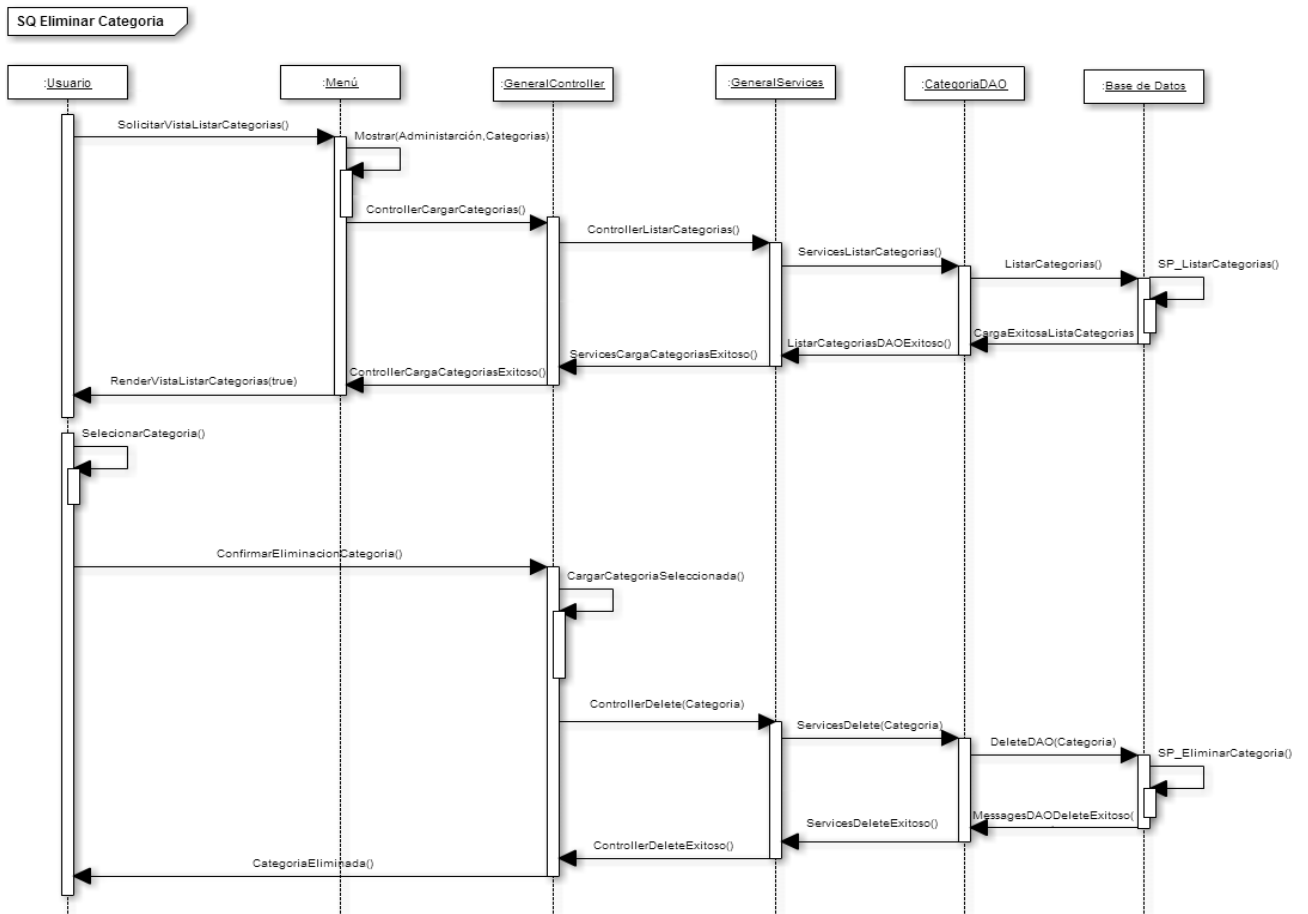


Diagrama 15. Secuencia Eliminar Categoría

El diagrama 16 se puede visualizar la secuencia de pase de mensajes entre objetos para crear un motivo de resolución.

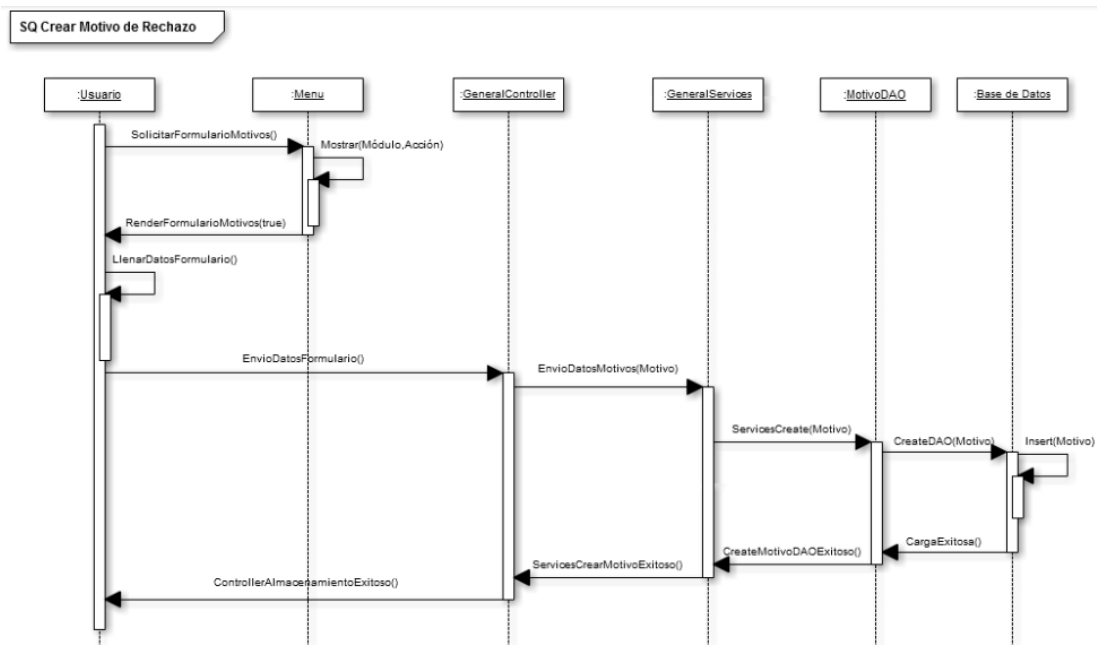


Diagrama 16. Secuencia Crear Motivos Rechazo

El diagrama 17 se puede visualizar la secuencia de pase de mensajes entre objetos para modificar un motivo de rechazo.

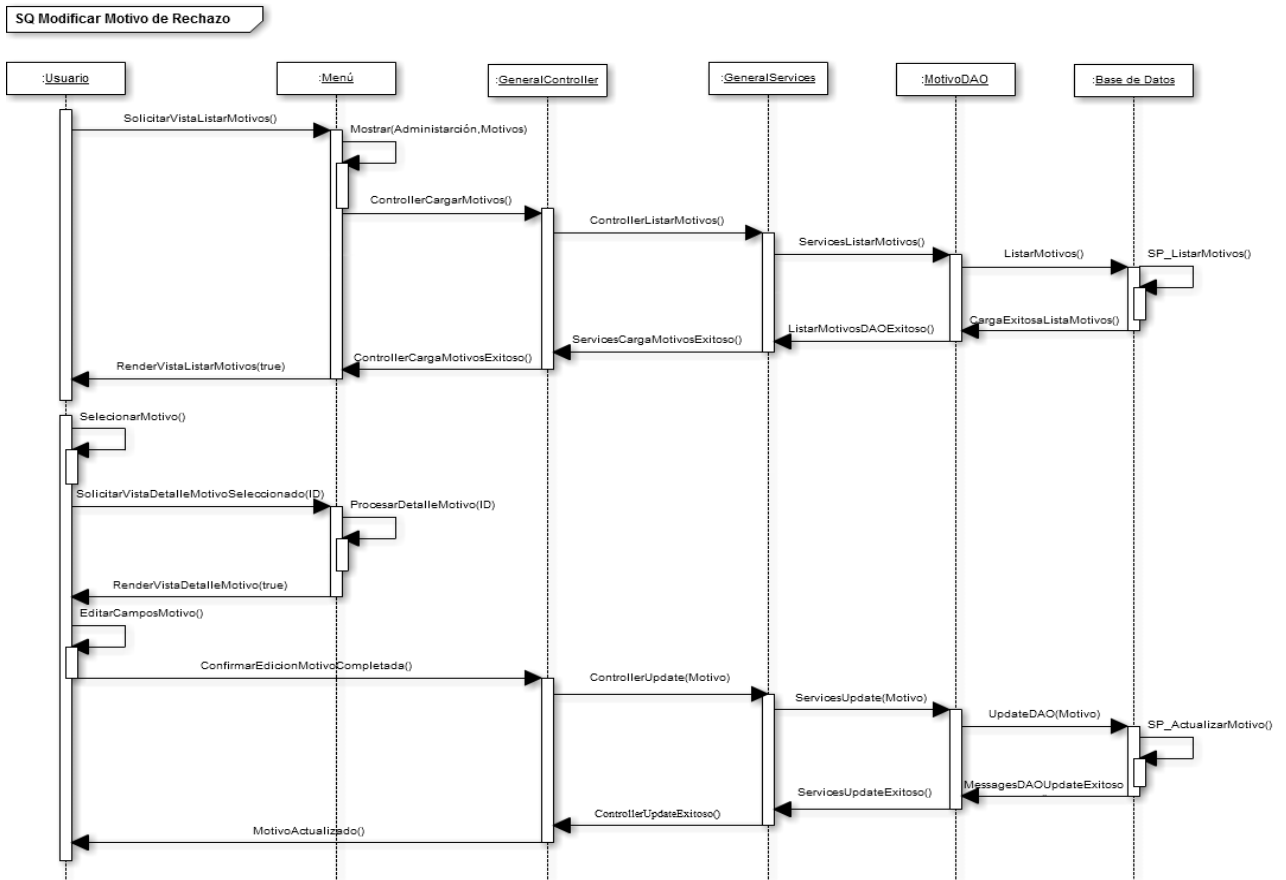


Diagrama 17. Secuencia Modificar Motivos Rechazo

El diagrama 18 se puede visualizar la secuencia de pase de mensajes entre objetos para eliminar un motivo de rechazo.

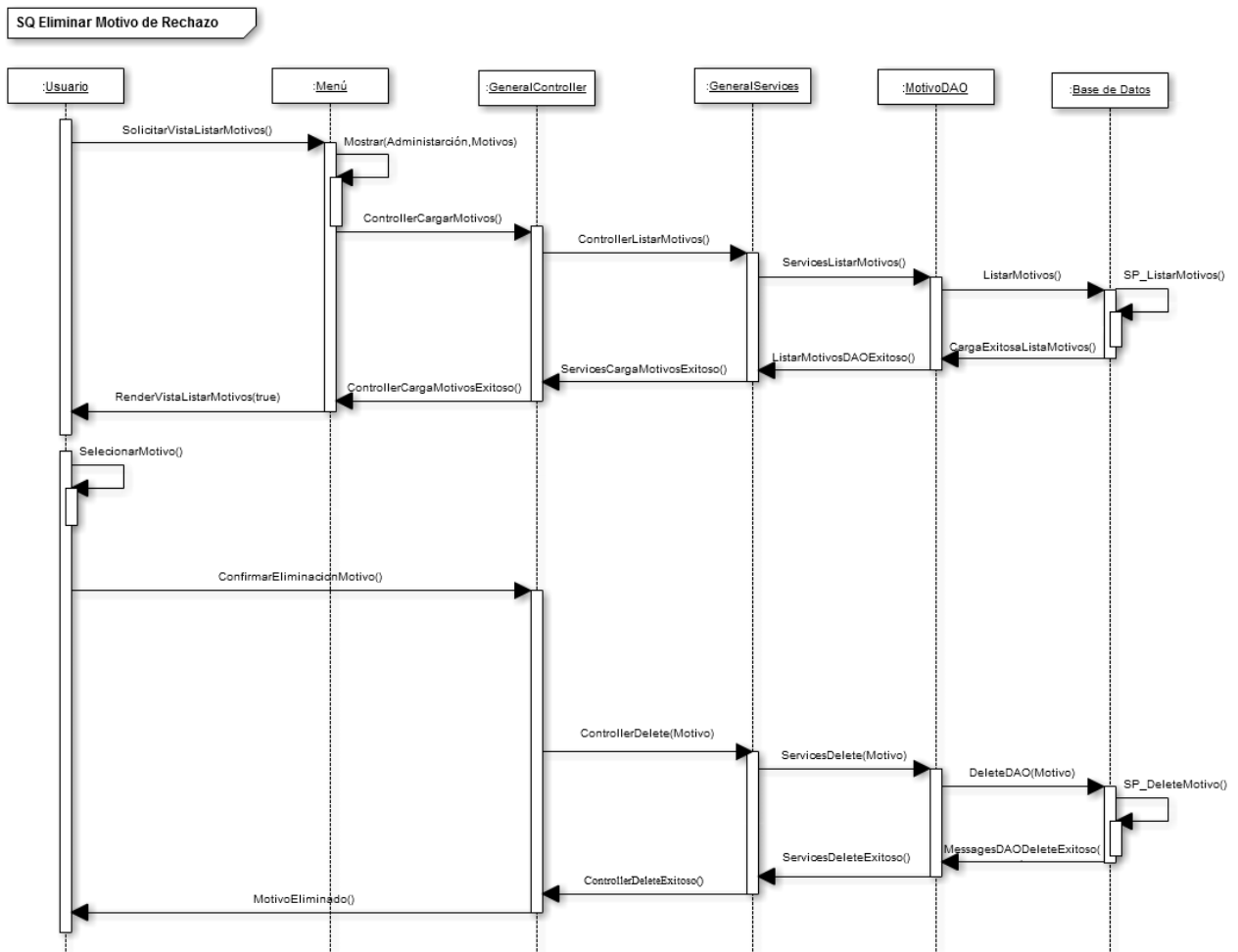


Diagrama 18. Secuencia Eliminar Motivos Rechazo

El diagrama 19 se puede visualizar la secuencia de pase de mensajes entre objetos para crear un proyecto.

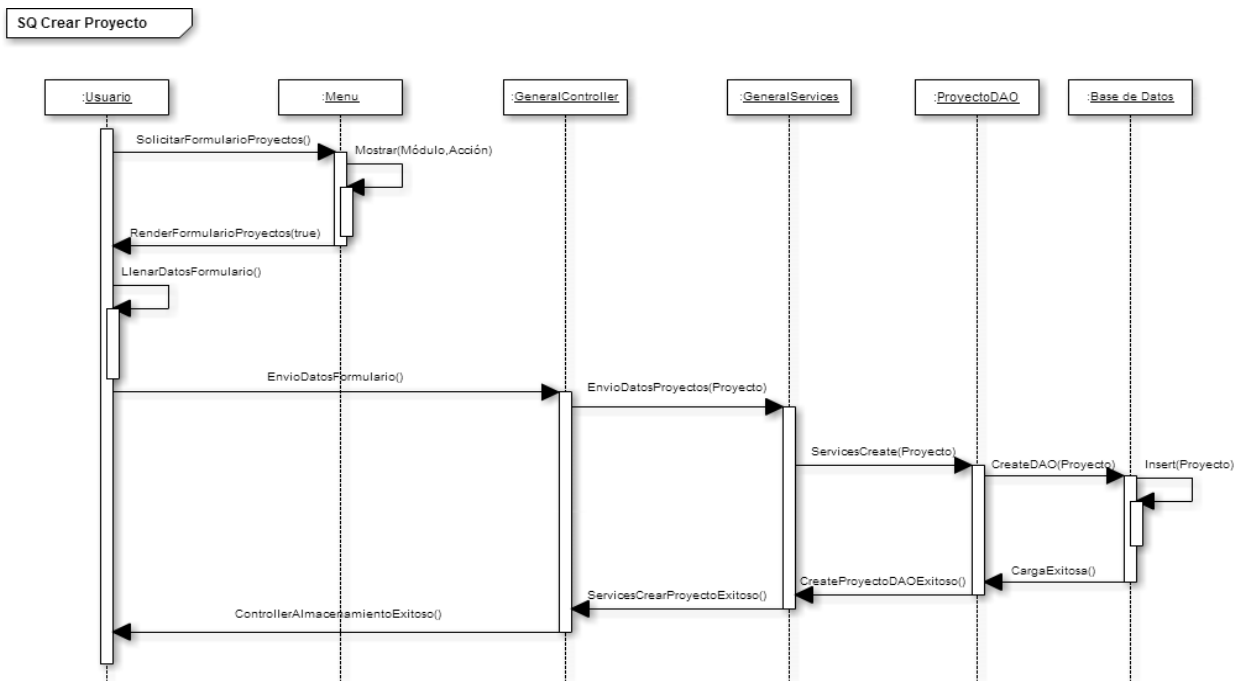


Diagrama 19. Secuencia Crear Proyectos

El diagrama 20 se puede visualizar la secuencia de pase de mensajes entre objetos para modificar un proyecto.

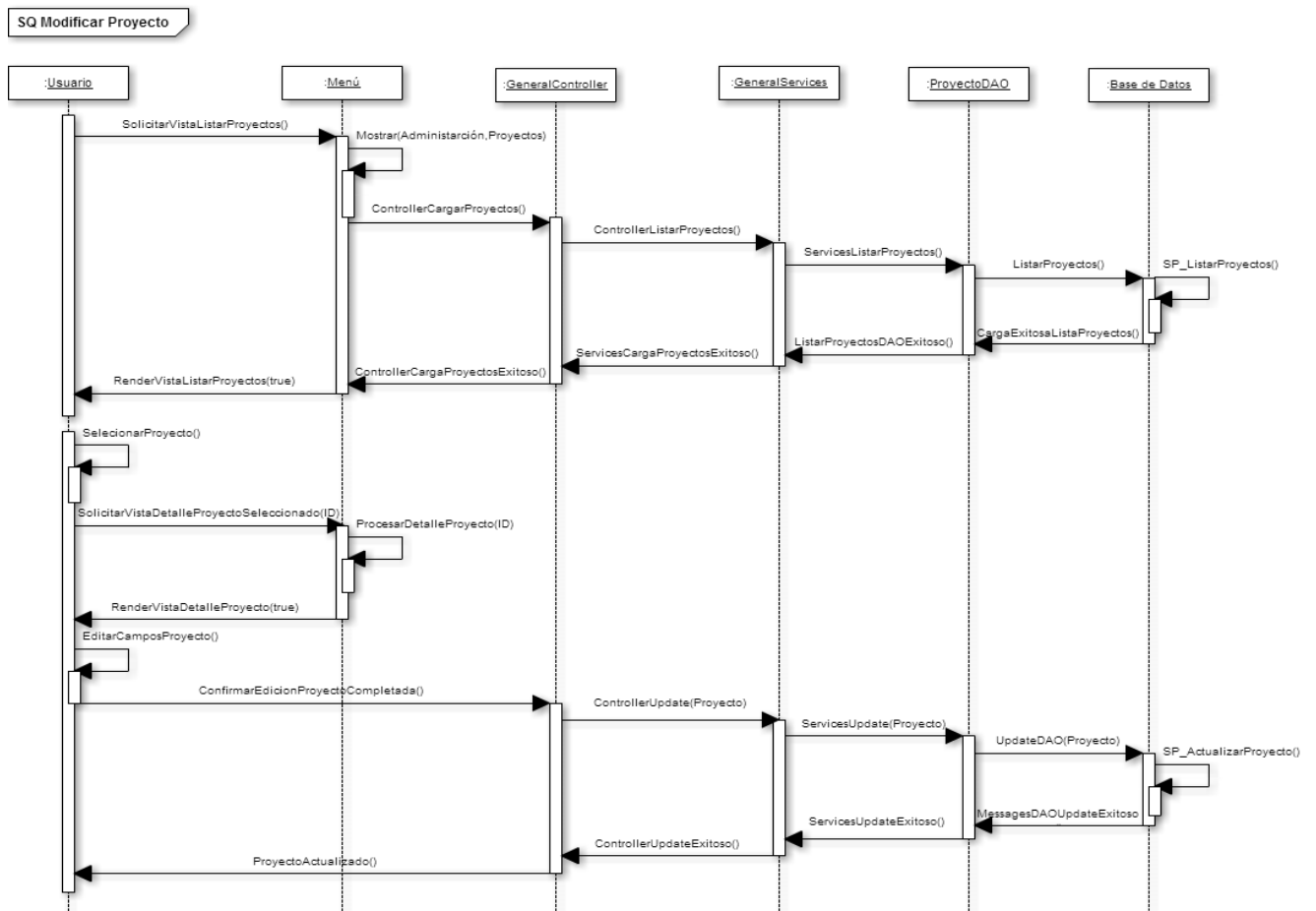


Diagrama 20. Secuencia Modificar Proyectos

El diagrama 21 se puede visualizar la secuencia de pase de mensajes entre objetos para eliminar un proyecto.

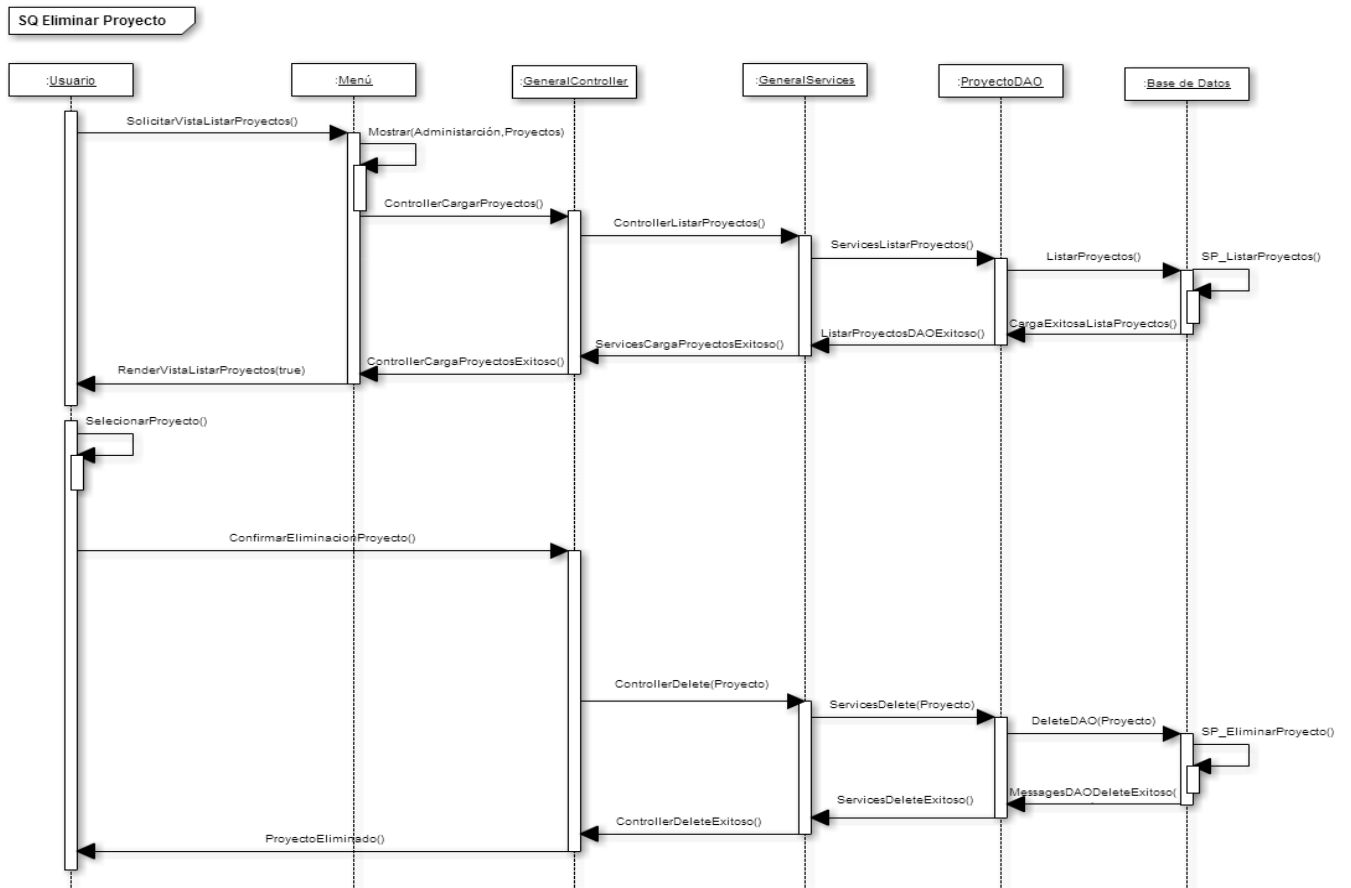


Diagrama 21. Secuencia Eliminar Proyectos

El diagrama 22 se puede visualizar la secuencia de pase de mensajes entre objetos para crear un usuario.

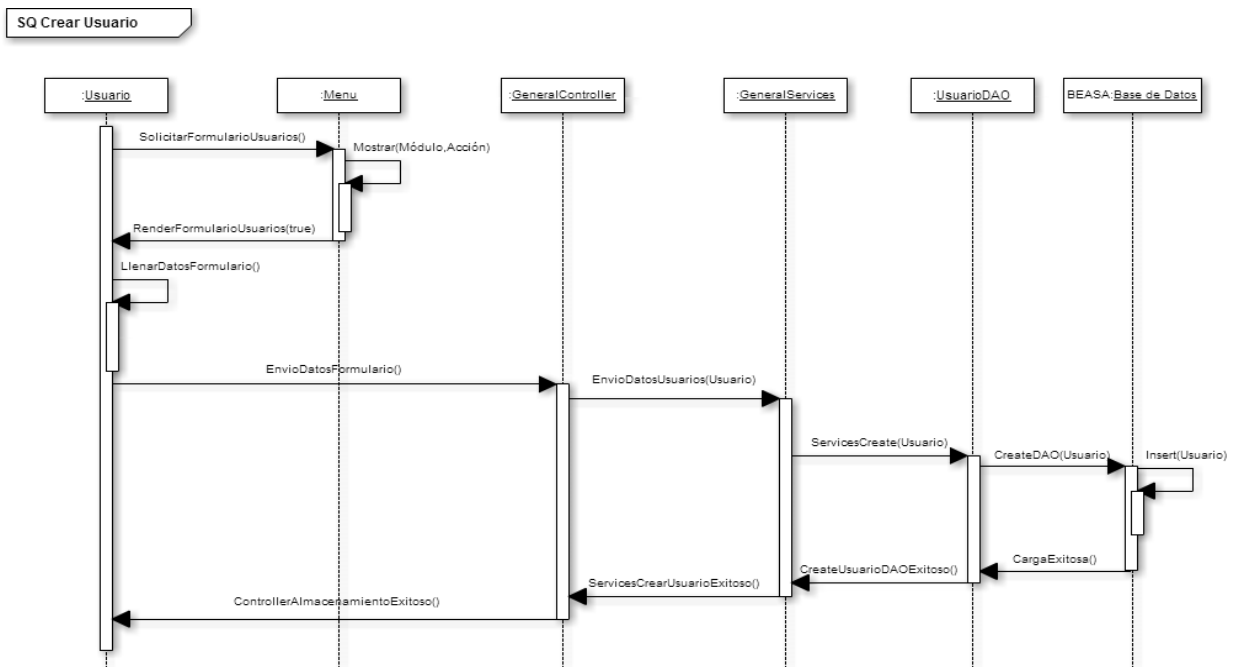


Diagrama 22. Secuencia Crear Usuario

El diagrama 23 se puede visualizar la secuencia de pase de mensajes entre objetos para modificar un usuario.

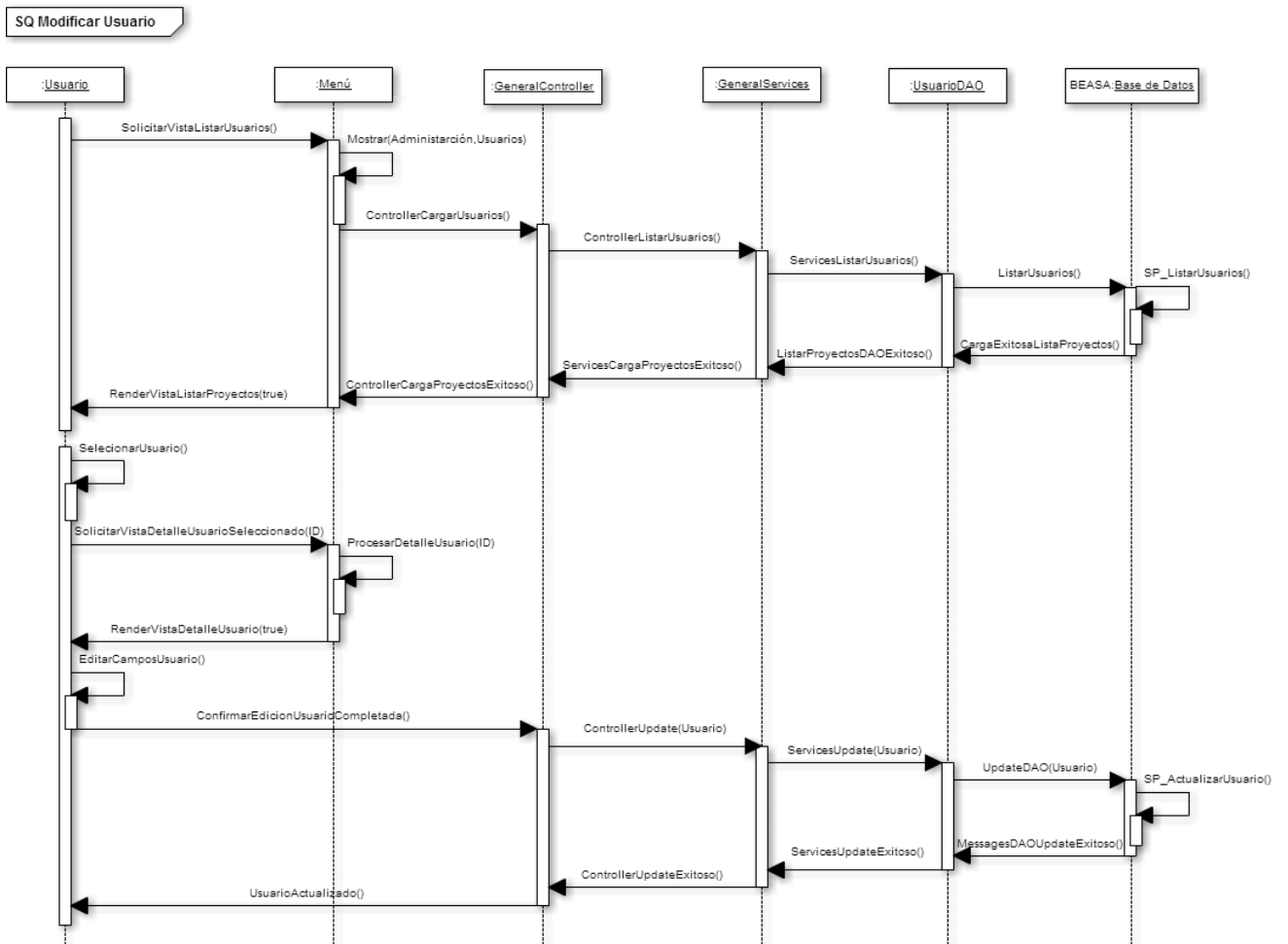


Diagrama 23. Secuencia Modificar Usuario

El diagrama 24 se puede visualizar la secuencia de pase de mensajes entre objetos para eliminar un usuario.

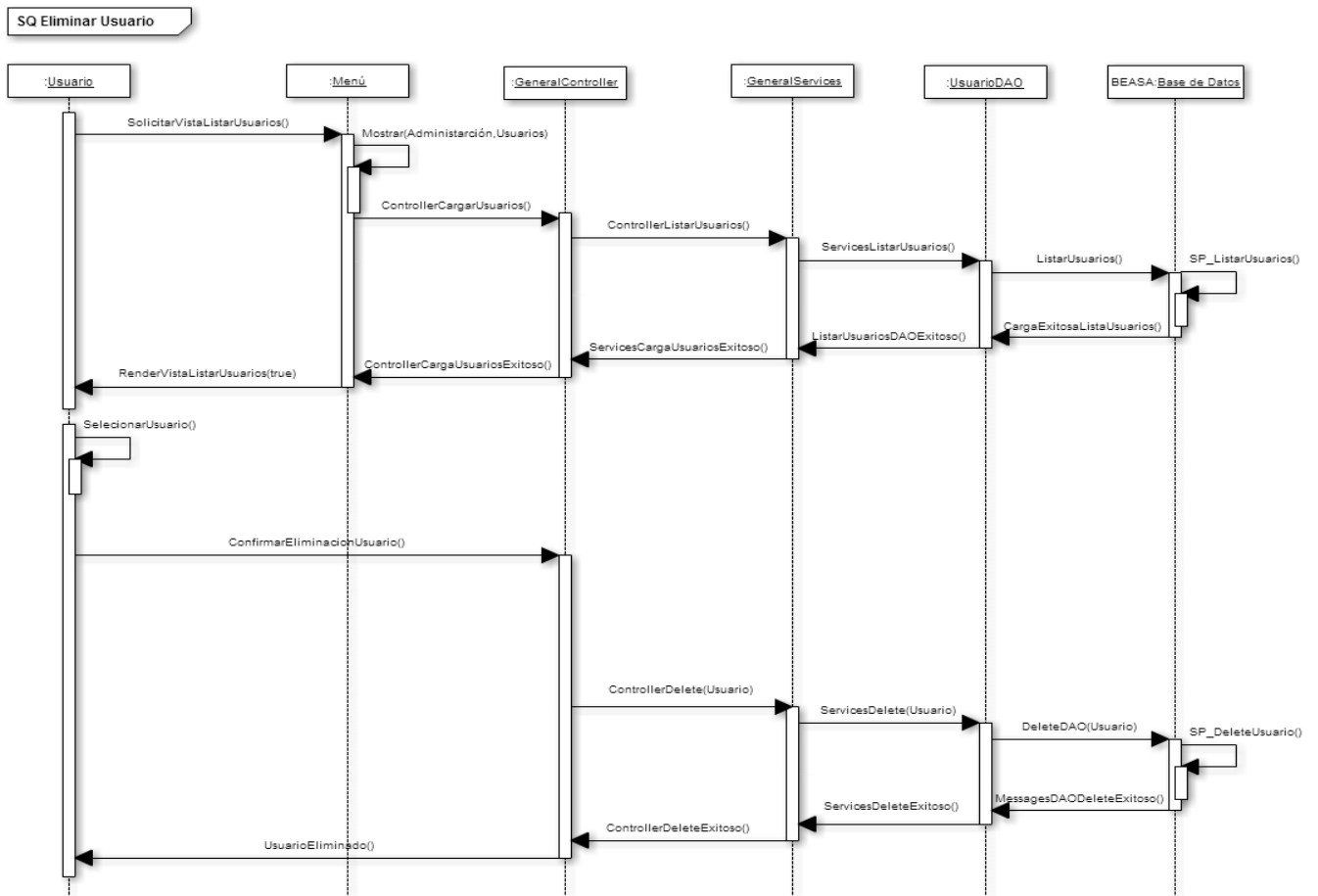


Diagrama 24. Secuencia Eliminar Usuario

El diagrama 25 se puede visualizar la secuencia de pase de mensajes entre objetos para

reportar o registrar un defecto.

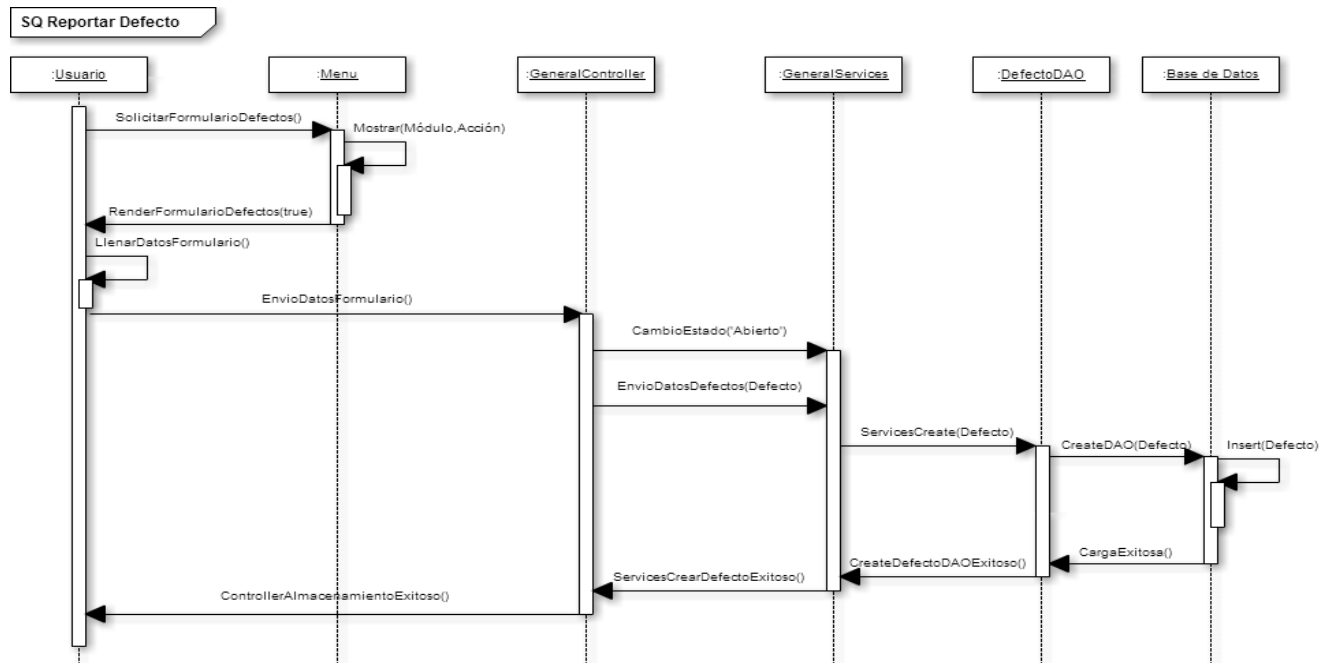


Diagrama 25. Secuencia Reportar Defecto

El diagrama 26 se puede visualizar la secuencia de pase de mensajes entre objetos para

modificar un defecto.

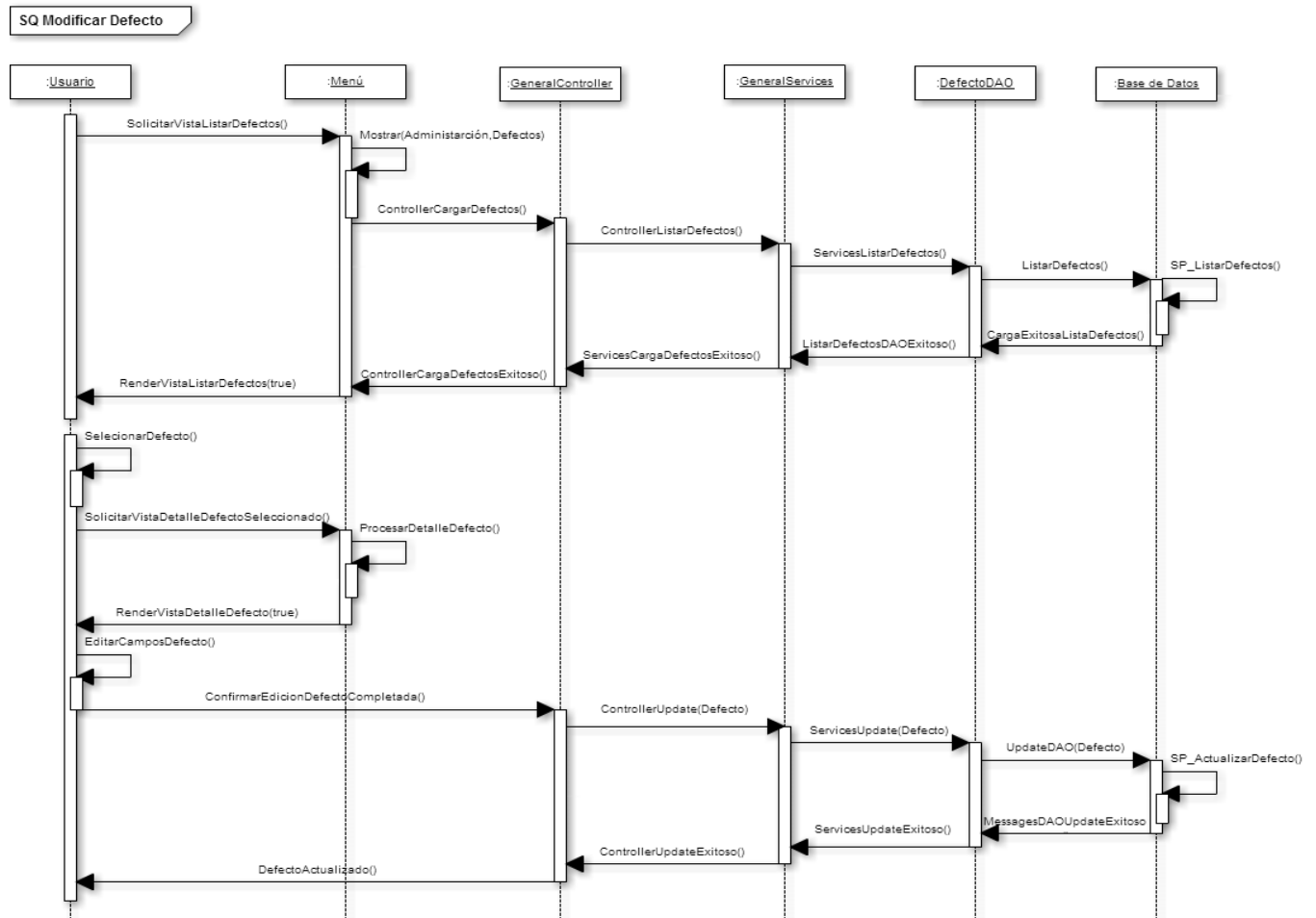


Diagrama 26. Secuencia Modificar Defecto

El diagrama 27 representa los pases de mensajes entre objetos para asignar un defecto.

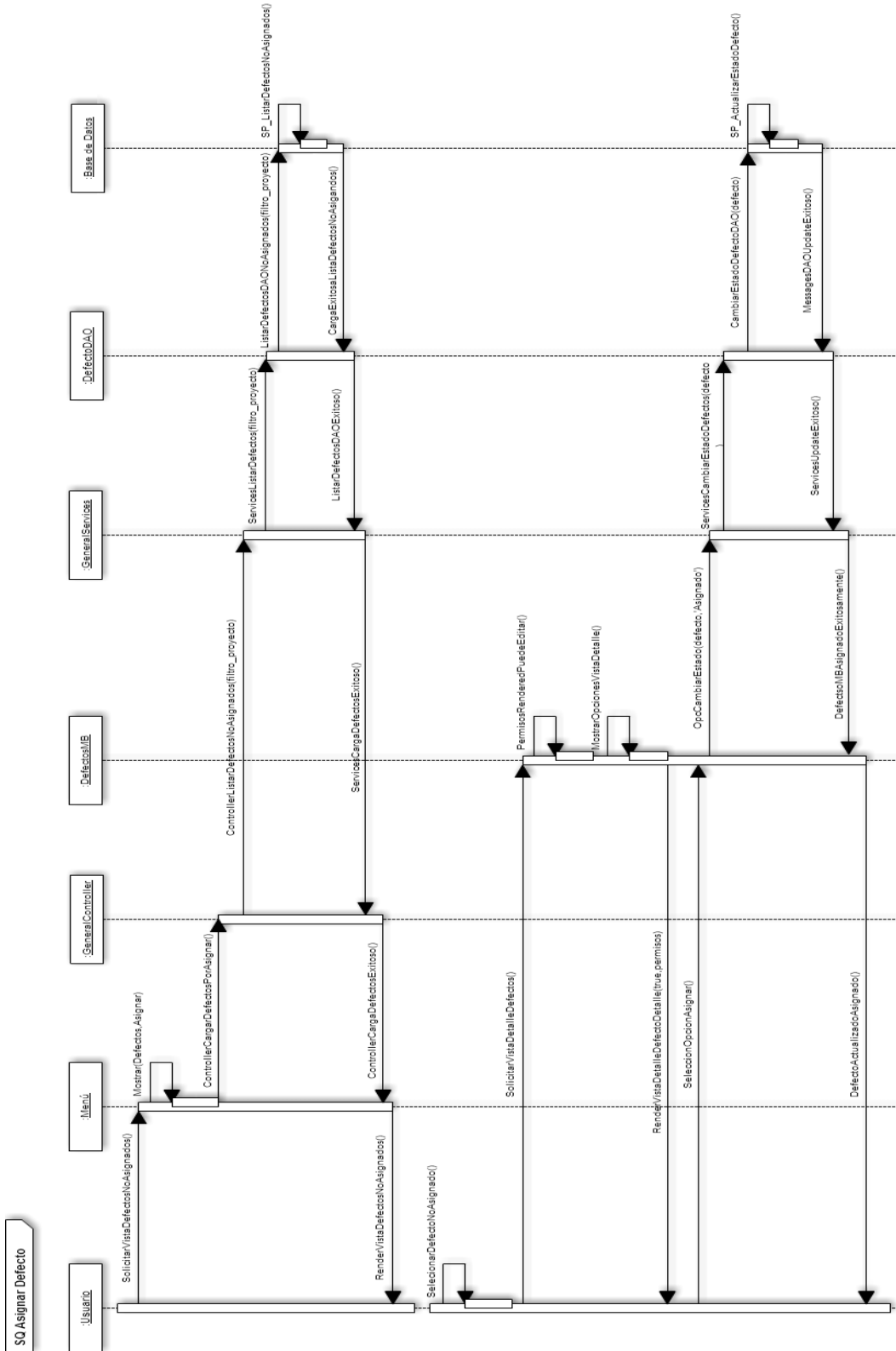


Diagrama 27. Secuencia Asignar Defecto

El diagrama 28 representa los pases de mensajes entre objetos para resolver un defecto.

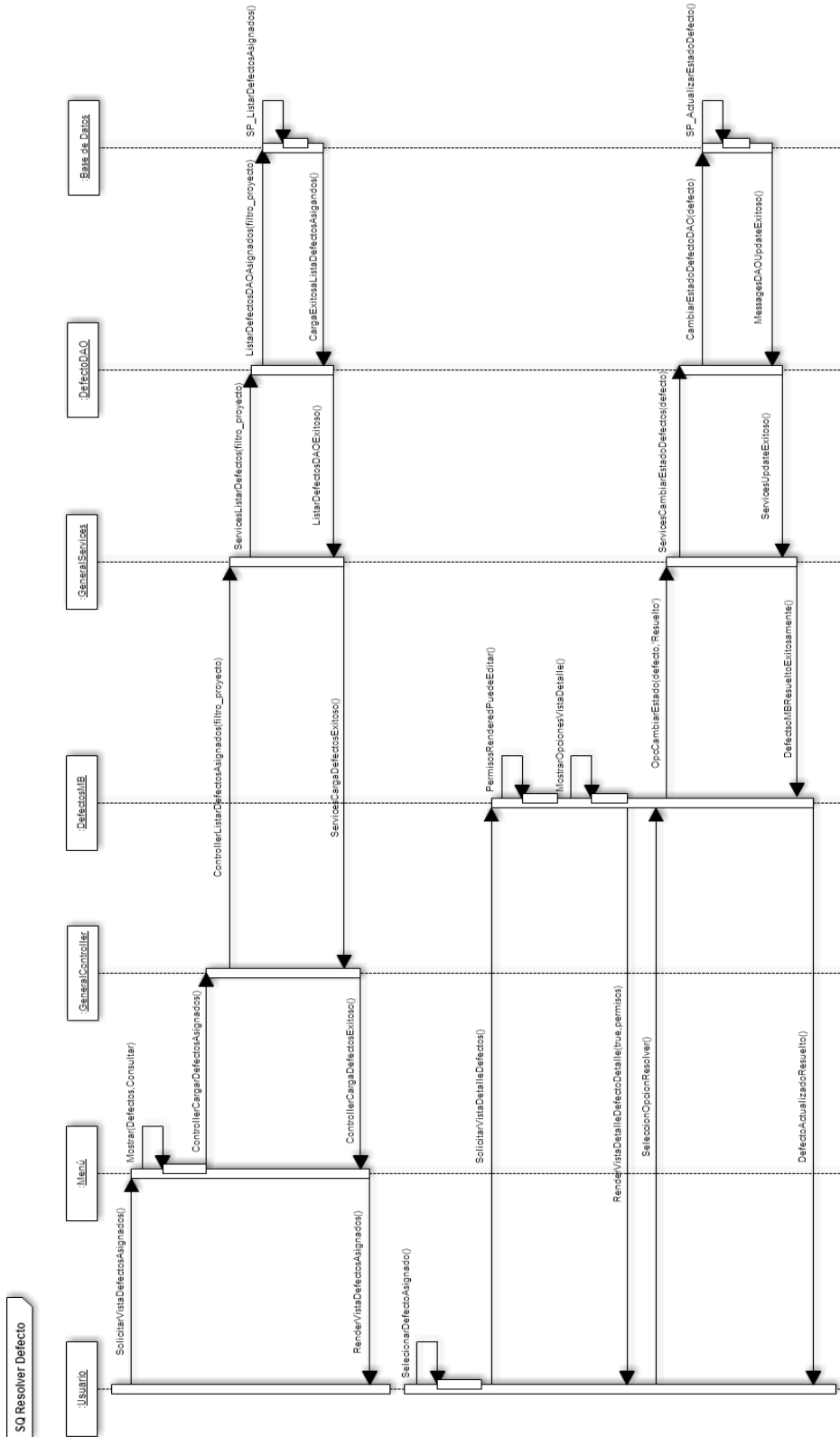


Diagrama 28. Secuencia Resolver Defecto

El diagrama 29 representa los pases de mensajes entre objetos para Re-Abrir un defecto.

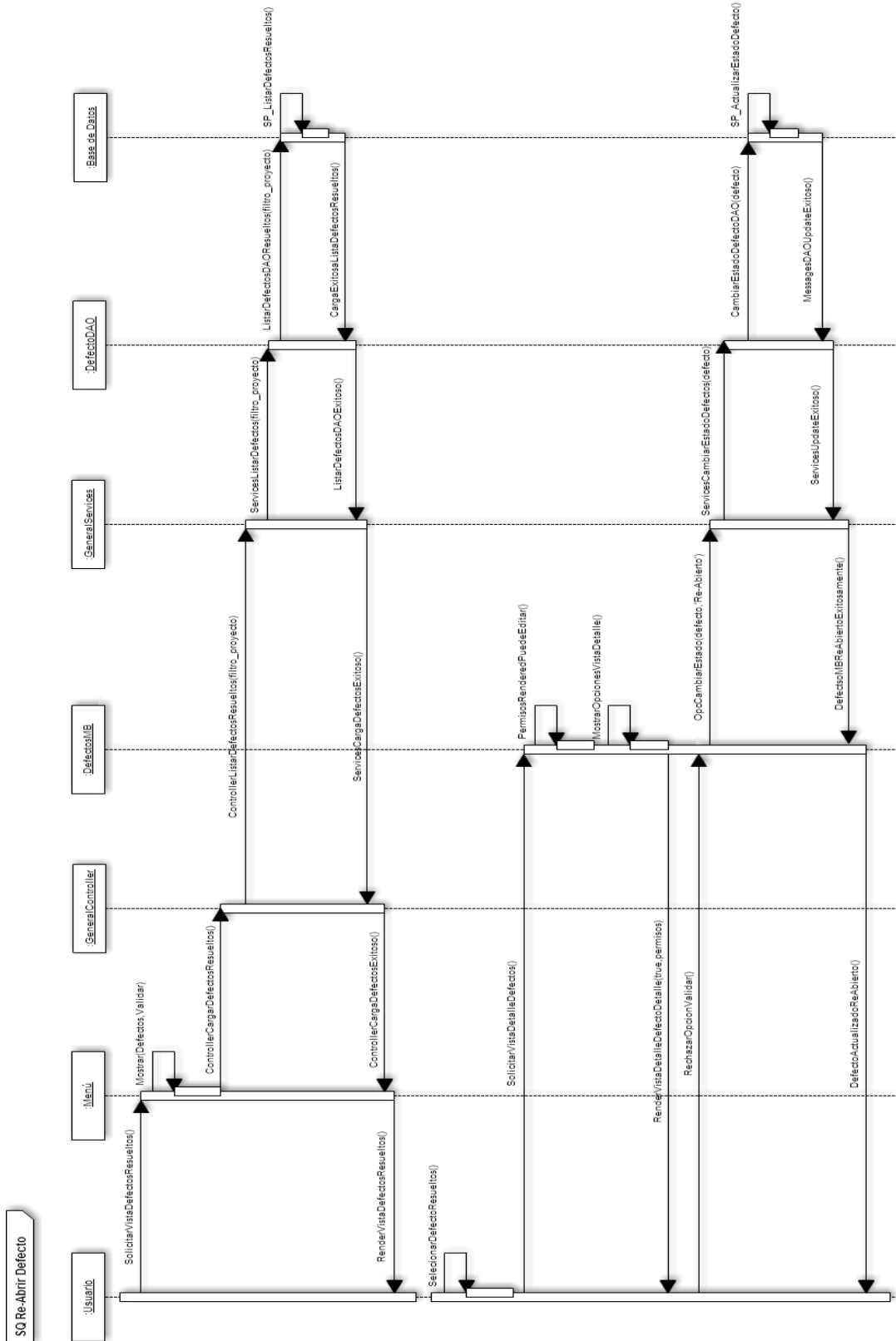


Diagrama 29. Secuencia Re- Abrir Defecto

El diagrama 30 representa los pases de mensajes entre objetos para Re-Abrir un defecto.

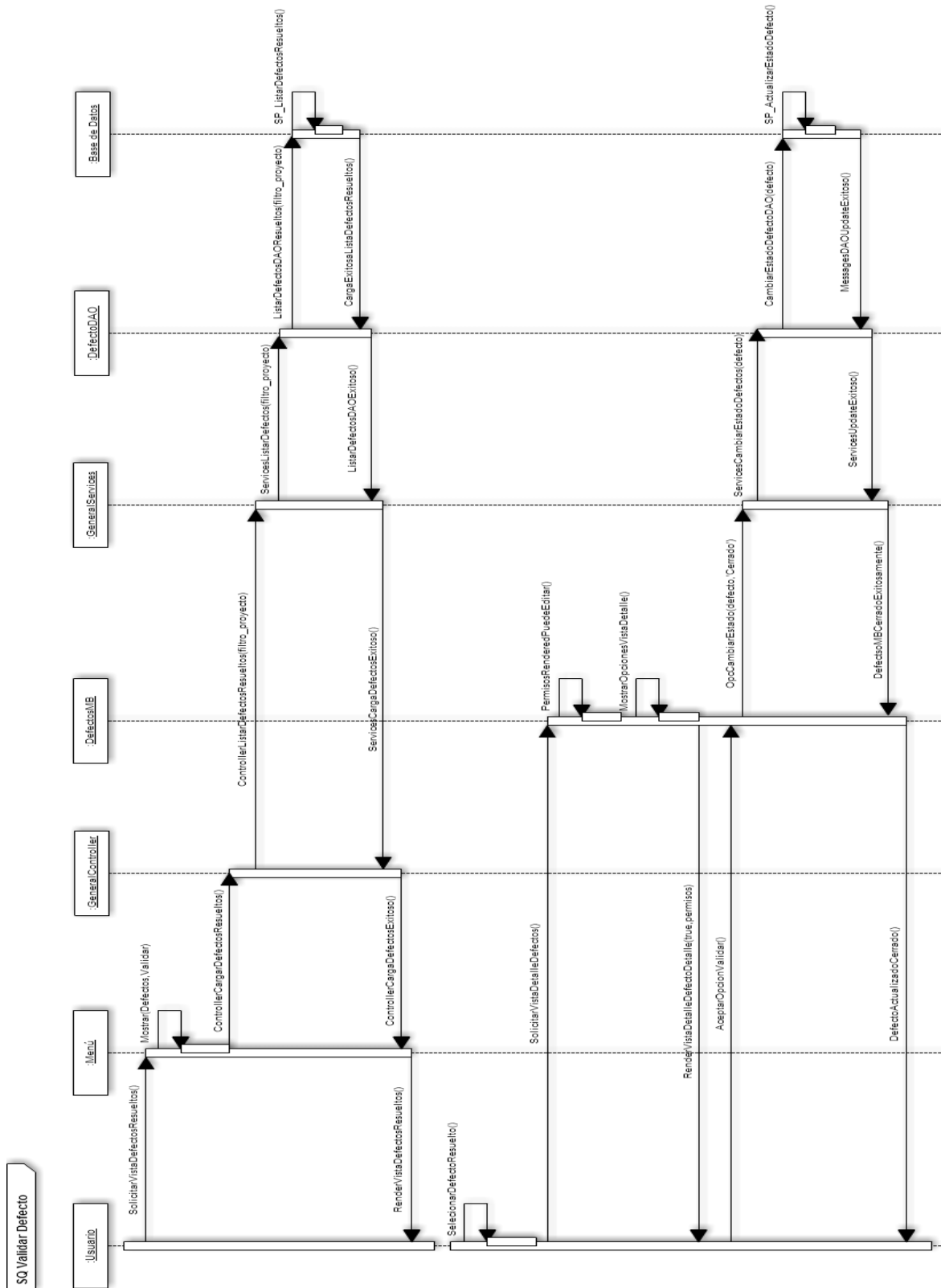


Diagrama 30. Secuencia Validar Defecto

- Diagrama Modelo de Vistas Prototipo del Sistema:** A continuación el diagrama 31 representa la estructura de vistas relacionadas entre sí a través de secuencias de re-direccionamiento que posee el sistema en forma ordenada.

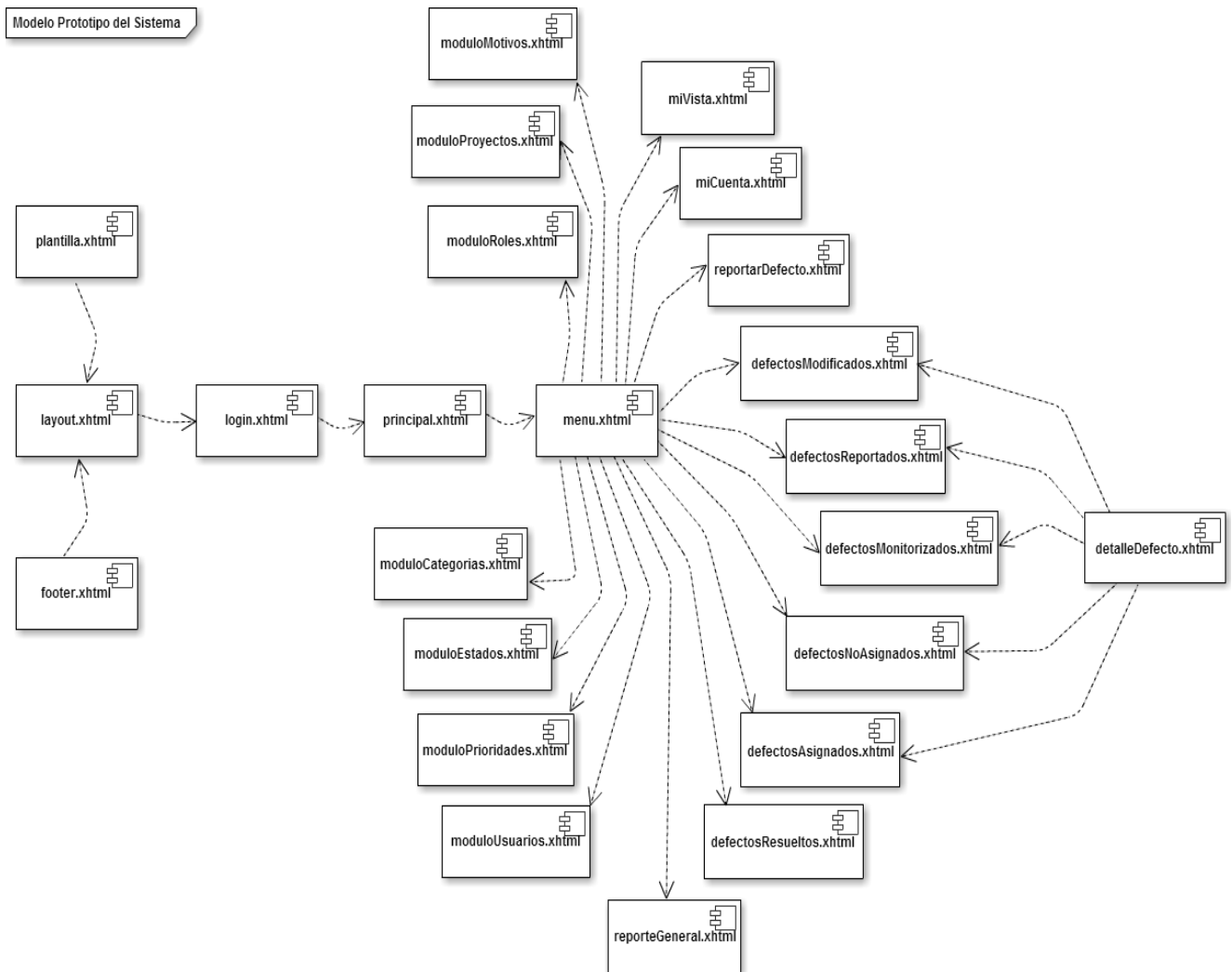


Diagrama 31. Prototipo Modelo Vistas del Sistema

- **Diagrama de Objeto por Estado de Defectos:**

El diagrama 32 representa las relaciones entre el usuario administrador y un defecto a través del estado que posee dicho defecto.

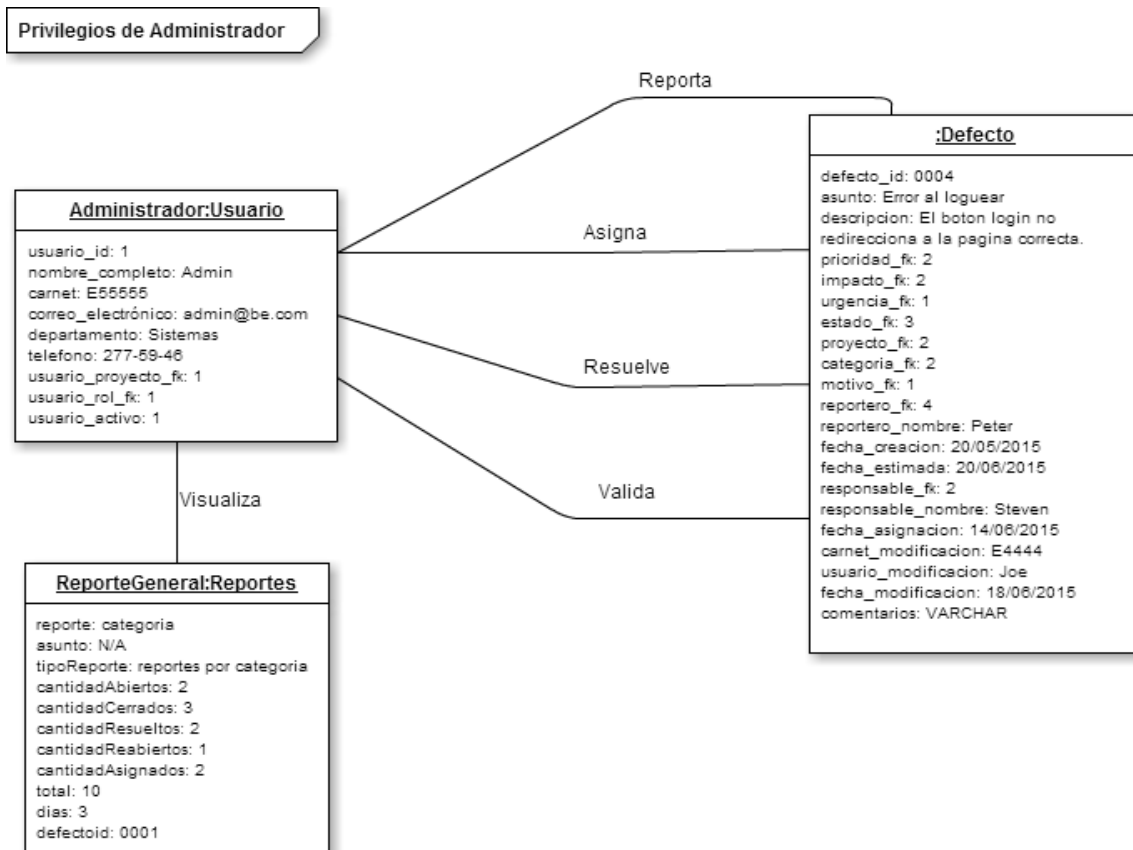


Diagrama 32. Diagrama de Estados Privilegios Administrador

El diagrama 33 representa las relaciones entre el usuario líder de proyecto y un defecto a través del estado que posee dicho defecto.

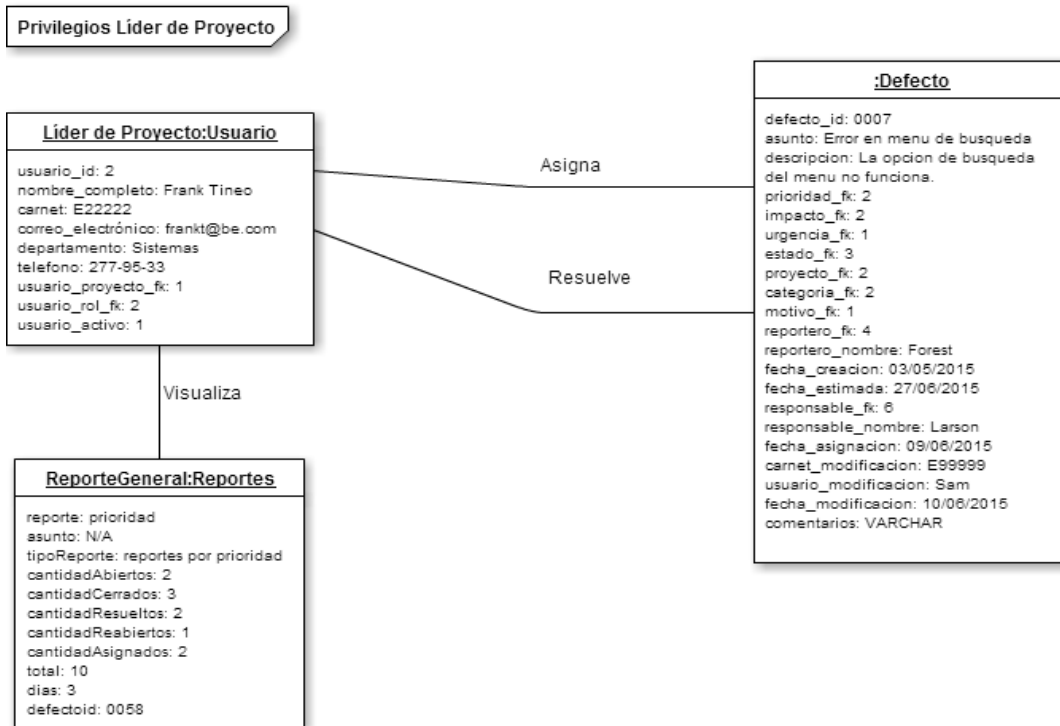


Diagrama 33. Diagrama de Estados Privilegios Líder de Proyecto

El diagrama 34 representa las relaciones entre el usuario analista y un defecto a través del estado que posee dicho defecto.

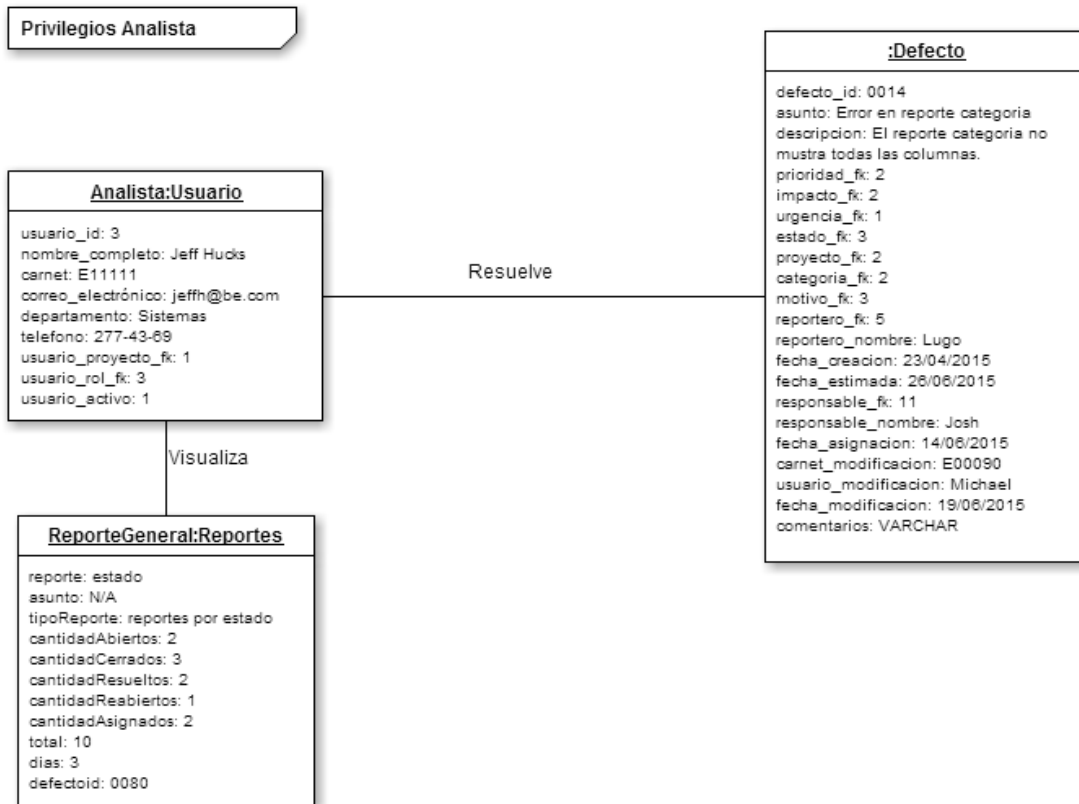


Diagrama 34. Diagrama de Estados Privilegios Analista

El diagrama 35 representa las relaciones entre el usuario funcional y un defecto a través del estado que posee dicho defecto.

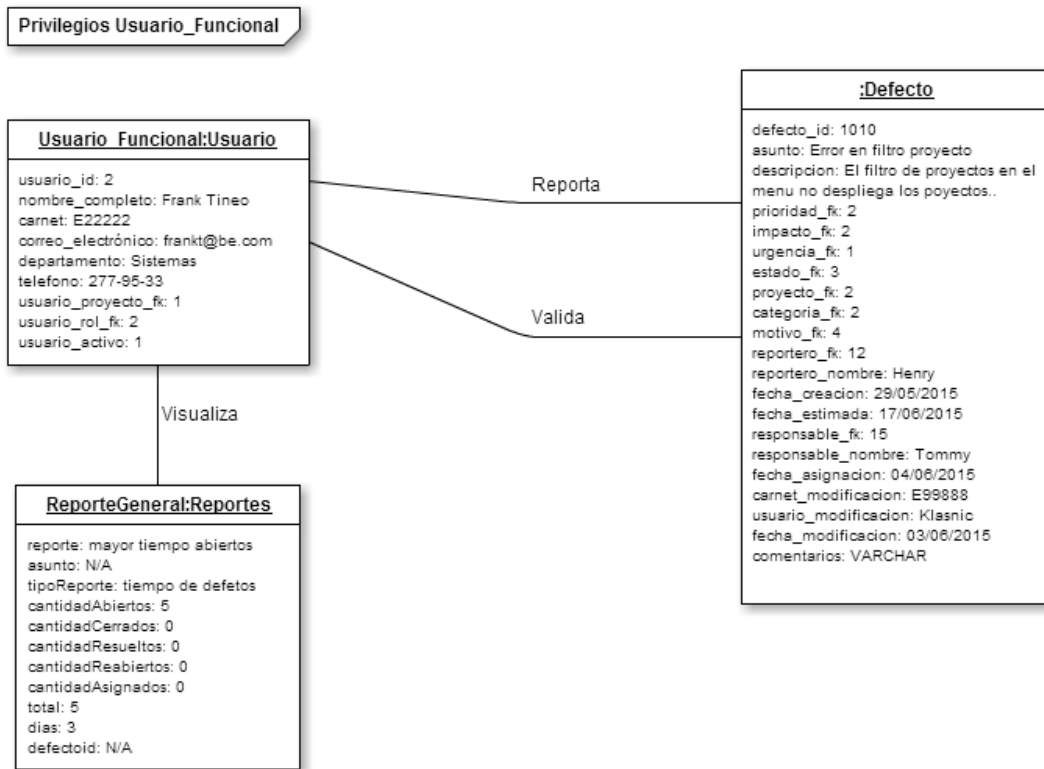


Diagrama 35. Diagrama de Estados Privilegios Usuario_Funcional

El diagrama 36 representa las relaciones entre el usuario visualizador o gerente y un defecto a través del estado que posee dicho defecto.

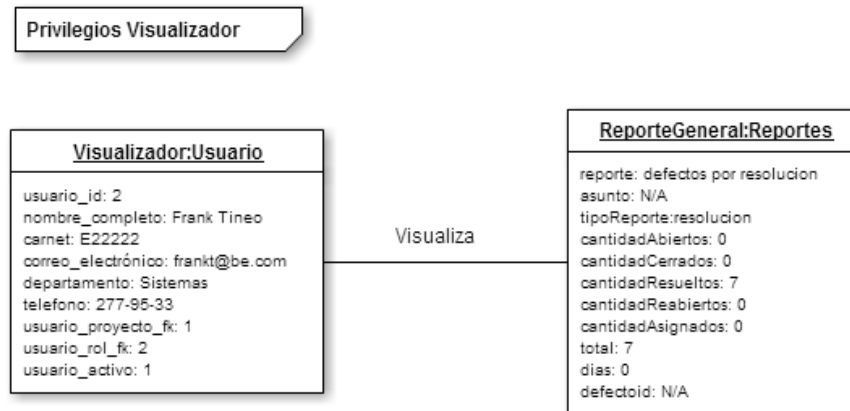


Diagrama 36. Diagrama de Estados Privilegios Visualizador

- Diseño de Base de Datos:** A continuación se muestra en detalle el diseño de toda la Base de datos con la que se conecta el sistema a través de diversos modelos de representación lógica.
 - Modelo E-R:** En el diagrama 37 se representa el modelo entidad-relación de la Base de Datos.

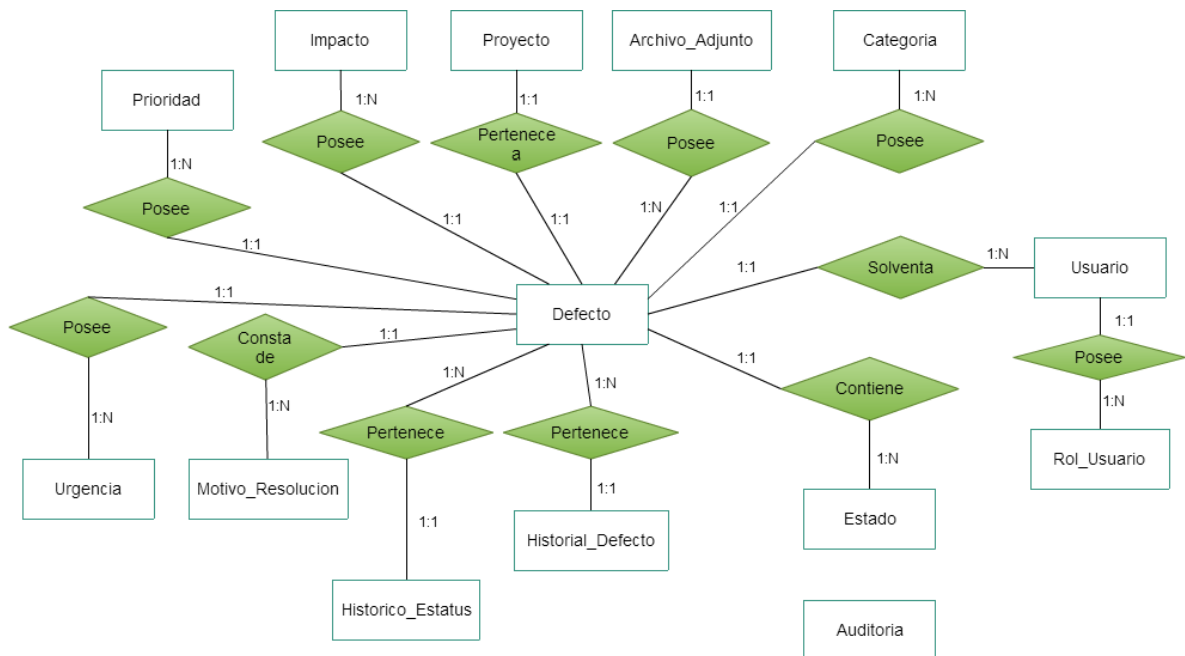


Diagrama 37. Diagrama Modelo E-R

➤ **Modelo Relacional:** En el diagrama 38 se muestra en detalle el modelo relacional que corresponde al diseño de la Base de Datos.

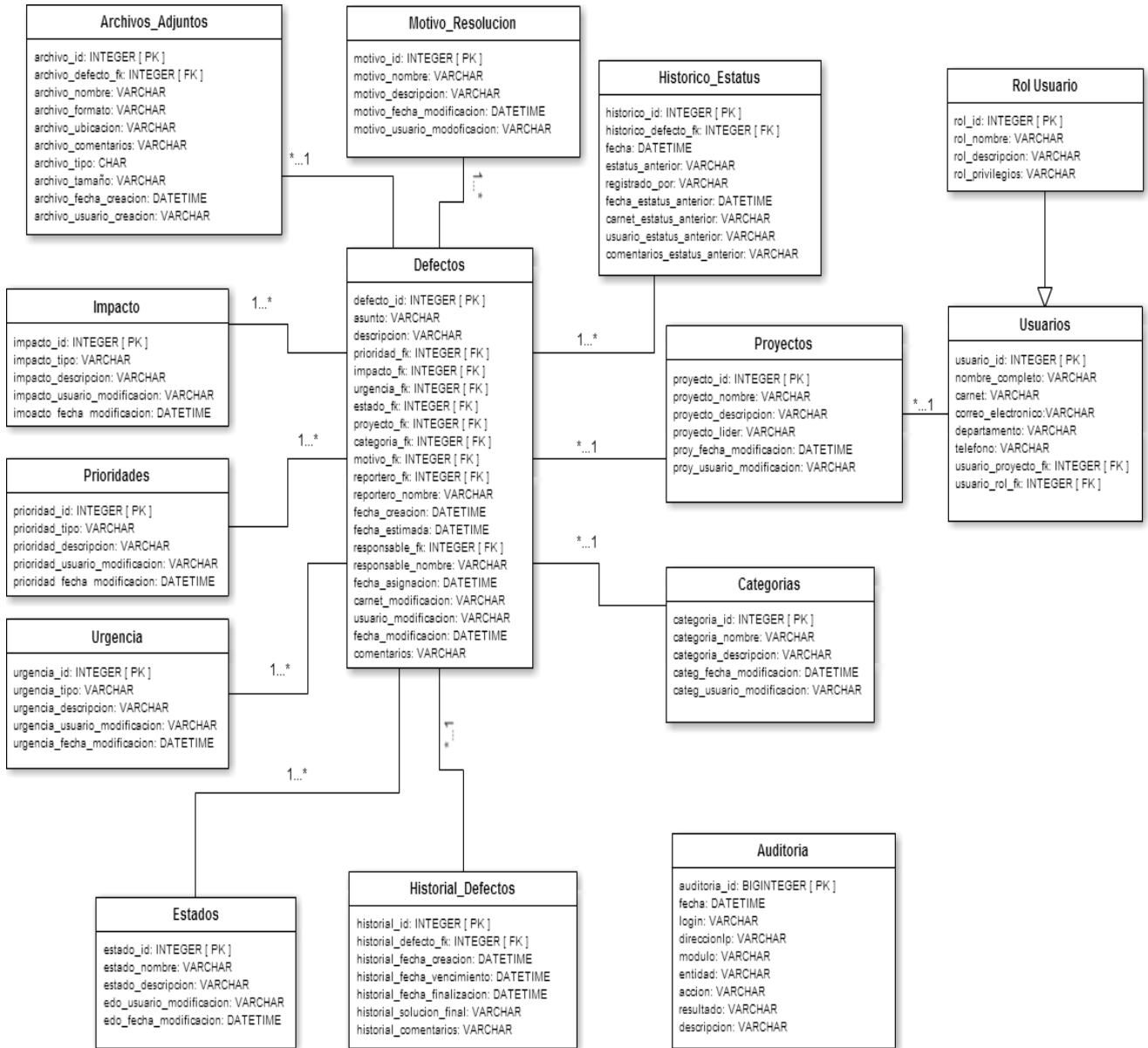


Diagrama 38. Modelo Relacional de Datos

- **Diccionario de Datos:** Tabla que muestra en detalle el diccionario de datos por cada tabla.

Atributo	Descripción	Tabla
defecto_id	Identificador unívoco de cada defecto.	
asunto	Asunto o título del defecto reportado.	
descripción	Descripción detallada de las características sobre el defecto reportado.	
prioridad_fk	Identificador de la tabla PRIORIDADES que indica la prioridad del defecto.	
impacto_fk	Identificador de la tabla IMPACTO que indica el impacto que generó el defecto.	
urgencia_fk	Identificador de la tabla URGENCIA que indica la urgencia que posee el defecto.	DEFECTOS
estado_fk	Identificador de la tabla ESTADOS que indica el estado el cual se encuentra el defecto.	
proyecto_fk	Identificador de la tabla PROYECTOS que indica el proyecto al cual pertenece el defecto.	
categoria_fk	Identificador de la tabla CATEGORIAS que indica la categoría al cual pertenece el defecto.	
reportero_fk	Identificador de la tabla USUARIO que indica que usuario reportó el defecto.	

fecha_creacion	Fecha de creación del defecto.	
fecha_estimada	Fecha estimada de resolución del defecto.	
usuario_modificacion	Nombre del usuario que modificó algún campo del defecto.	
fecha_modificacion	Fecha en que se modificó el defecto.	
comentarios	Comentarios adicionales a la descripción del defecto en caso de observaciones.	
defecto_activo	Bit que indica si el defecto está activo o no.	
reportero_nombre	Nombre del usuario que reportó el defecto.	
carnet_modificacion	Nº de carnet del usuario que modificó algún campo del defecto.	
motivo_fk	Identificador de la tabla MOTIVOS_RECHAZO que indica el tipo de motivo por el cual fue resuelto el defecto.	
responsable_fk	Identificador de la tabla USUARIO que indica el usuario que es responsable de la resolución satisfactoria o no del defecto.	
responsable_nombre	Nombre del usuario que es responsable de la resolución satisfactoria o no del defecto.	
fecha_asignacion	Fecha en la cual se le asignó el defecto a un usuario determinado.	

Atributo	Descripción	Tabla
archivo_id	Identificador único de cada archivo adjunto.	ARCHIVOS_ADJUNTOS
archivo_nombre	Nombre del archivo adjunto.	
archivo_formato	Formato del archivo adjunto. (.jpg/*.pdf, etc).	
archivo_ubicacion	Directorio en el cual se encuentra almacenado el archivo adjunto.	
archivo_comentarios	Comentarios relacionados al archivo adjunto.	
archivo_activo	Bit que indica si el archivo adjunto está activo o no.	
archivo_tamaño	Tamaño del archivo adjunto.	
archivo_tipo	Char que indica si el archivo adjunto es de tipo resolución ('R') o de apertura de relacionado a un defecto ('A')	
archivo_usuario_creacion	Nombre del usuario que adjunto el archivo.	
archivo_fecha_creacion	Fecha en la cual el archivo fue adjunto.	
archivo_defecto_fk	Identificador de la tabla DEFECTOS que indica el defecto al cual el archivo adjunto se encuentra asociado.	

Atributo	Descripción	Tabla
idAuditoria	Identificador único de cada auditoría registrada.	AUDITORIA
fecha	Fecha en la cual se registró la auditoría	
login	Nº del carnet del usuario autenticado al momento del registro de la auditoría.	
direccionIp	Dirección IP de la máquina en la cual se registró la auditoría.	
modulo	Modulo del sistema donde se registró la auditoría.	
entidad	Nombre de la tabla a la cual hace referencia la auditoría.	
accion	Operación que se realizó al momento del registro de auditoría.	
resultado	Estado exitoso o erróneo de la auditoría registrada.	
descripcion	Descripción detallada de la auditoría.	

Atributo	Descripción	Tabla
categoría_id	Identificador único de cada categoría.	CATEGORIAS
categoría_nombre	Nombre de la categoría.	
categoría_descripcion	Descripción detallada de la categoría.	
categoría_activo	Bit que determina si la categoría está activa o no.	
categ_fecha_modificacion	Fecha en que la categoría fue modificada.	
Categ_usuario_modificacion	Nombre del usuario que modificó la categoría.	

Atributo	Descripción	Tabla
estado_id	Identificador unívoco de cada tipo de estado en el cual se encuentra un defecto.	ESTADOS
Estado_nombre	Nombre del estado del defecto.	
Estado_descripcion	Descripción del estado del defecto.	
Estado_activo	Bit que determina si el estado del defecto se encuentra activo o no.	
edo_usuario_modificacion	Nombre del usuario que modificó el estado del defecto.	
Edo_fecha_modificacion	Fecha en que fue modificado algún campo del estado del defecto.	

Atributo	Descripción	Tabla
historial_id	Identificador unívoco de cada historial que hace referencia a un defecto en particular.	HISTORIAL_DEFECTOS
Historial_defecto_fk	Identificador de la tabla DEFECTOS que referencia a su historial.	
Historial_fecha_creacion	Fecha en la cual fue creado el registro historial del defecto.	
Historial_fecha_vencimiento	Fecha de expiración para que el defecto sea resuelto.	
historial_fecha_finalizacion	Fecha en la cual el caso del defecto fue cerrado.	
historial_solucion_final	Solución definitiva a la resolución del defecto.	
historial_comentarios	Comentarios relacionados al historial de un defecto.	

Atributo	Descripción	Tabla
historico_id	Identificador unívoco para cada histórico de seguimiento que pertenece a un defecto.	HISTORICO_ESTATUS
historico_defecto_fk	Identificador de la tabla DEFECTOS relacionado a su histórico correspondiente.	
fecha	Fecha en que se registró el cambio en el histórico de seguimiento del defecto.	
estatus_anterior	Estado anterior del defecto.	
registrado_por	Nº carnet del usuario que registró el histórico del defecto.	
fecha_estatus_anterior	Fecha de estado o modificación anterior del defecto.	
usuario_estatus_anterior	Nombre del usuario que modificó anteriormente el defecto.	
comentarios_estatus_anterior	Comentarios anteriores de cambios en el defecto.	
carnet_estatus_anterior	Nº de carnet del usuario que modificó el defecto en un estado anterior.	

Atributo	Descripción	Tabla
impacto_id	Identificador unívoco de cada tipo de impacto relacionado a un defecto.	IMPACTO
impacto_tipo	Tipo de impacto relacionado a un defecto.	
impacto_descripcion	Descripción en detalle de cada tipo de impacto.	
impacto_activo	Bit que indica si el impacto está activo o no.	
impacto_usuario_modificacion	Nombre del usuario que realizó alguna modificación sobre un tipo de impacto.	
impacto_fecha_modificacion	Fecha en que se realizó alguna modificación sobre un tipo de impacto.	

Atributo	Descripción	Tabla
motivo_id	Identificador unívoco de cada tipo de motivo de rechazo de defectos.	MOTIVO_RECHAZO
motivo_nombre	Nombre del motivo de rechazo del defecto.	
motivo_descripcion	Descripción detallada de cada tipo de motivo de rechazo de defectos.	
motivo_activo	Bit que indica si un tipo de motivo de rechazo de defectos está activo o no.	
motivo_usuario_modificacion	Nombre del usuario que realizó la última modificación sobre el motivo de rechazo del defecto.	
Motivo_fecha_modificacion	Fecha en que se realizó la última modificación sobre el motivo de rechazo del defecto.	

Atributo	Descripción	Tabla
prioridad_id	Identificador unívoco para cada tipo de prioridad.	PRIORIDADES
Prioridad_tipo	Nombre de cada tipo de prioridad.	
Prioridad_decripcion	Descripción detallada de cada tipo de prioridad.	
Prioridad_activo	Bit que determina si cada tipo de prioridad está activa o no.	
prior_usuario_modificacion	Nombre del último usuario que modificó alguna prioridad.	
Prior_fecha_modificacion	Fecha de la última modificación realizada sobre algún tipo de prioridad.	

Atributo	Descripción	Tabla
proyecto_id	Identificador unívoco de cada proyecto.	PROYECTOS
Proyecto_nombre	Nombre de cada proyecto.	
Proyecto_descripcion	Descripción general de cada proyecto.	
Proyecto_lider	Nombre del usuario que encargado del liderazgo de un proyecto en particular.	
Proyecto_activo	Bit que determina si el proyecto se encuentra activo o no.	
proy_fecha_modificacion	Fecha de la última modificación sobre un proyecto en particular.	
Proy_usuario_modificacion	Nombre de usuario que realizó la última modificación sobre un proyecto en particular.	

Atributo	Descripción	Tabla
rol_id	Identificador unívoco de cada rol de usuario.	ROL_USUARIO
Rol_nombre	Nombre de cada rol de usuario.	
Rol_descripcion	Características de cada rol de usuario.	
rol_privilegios	Permisos que tiene cada rol de usuario.	
rol_activo	Bit que define si el rol de usuario está activo o no.	

Atributo	Descripción	Tabla
urgencia_id	Identificador unívoco de cada tipo de urgencia.	URGENCIA
urgencia_tipo	Nombre de cada tipo de urgencia.	
urgencia_descripcion	Descripción detallada de cada tipo de urgencia.	
urgencia_activo	Bit que determina si algún tipo de urgencia se encuentra activa o no.	
urgencia_usuario_modificacion	Nombre del usuario que realizó la última modificación sobre algún tipo de urgencia.	
urgencia_fecha_modificacion	Fecha de la última modificación realizada sobre alguna urgencia en particular.	

Atributo	Descripción	Tabla
usuario_id	Identificador unívoco de cada usuario.	USUARIOS
nombre_completo	Nombre completo del usuario.	
carnet	Nº de carnet de usuario.	
correo_electronico	Correo electrónico del usuario.	
usuario_proyecto_fk	Identificador de la tabla PROYECTOS que establece la relación hacia un usuario en particular.	
usuario_rol_fk	Identificador de la tabla ROL_USUARIO que relaciona a un usuario con su respectivo rol.	
usuario_activo	Bit que define si el usuario está activo o no.	

2.1.3. FASE DE CONSTRUCCION

- **Disciplina Implementación**

The image shows a web form for user authentication. At the top, there is an orange header with the text "Autenticación de Usuarios". Below this, there are two input fields. The first is labeled "Usuario" and contains the text "admin". The second is labeled "Clave" and contains five asterisks "*****". At the bottom of the form, there are two blue buttons: "Ingresar" and "Borrar".

Figura 32. Inicio de Sesión.

Resumen de Defectos

ACTIVIDAD RECIENTE					
ID	Asunto	Prioridad	Estado	Categoría	Proyecto
Z	Bug Spring Security	MEDIA	ABIERTO	Externos	SGD
8	Limitar alcance de proyectos de funcional	ALTA	RE-ABIERTO	Exportar Datos	PDP2
8	Css	CRITICA	RE-ABIERTO	Defecto	SGD

CERRADOS					
ID	Asunto	Prioridad	Estado	Categoría	Proyecto
4	Prueba defecto	CRITICA	CERRADO	Defecto	SGD
1	Adjuntos	CRITICA	CERRADO	Defecto	SGD

Figura 33. Principal. Administrador.

Información de Usuario Autenticado

Nombre: Admin Torrez
 Carnet: E55555
 Email: Admin.torrez@bancoexterior.com
 Departamento: Desarrollo Sistema Gestion de Defectos
 Perfil: ADMINISTRADOR

Figura 34. Módulo Mi Cuenta.

GESTIÓN DE PROYECTOS

Id	Nombre	Lider Proyecto	Editar	Borrar
SGD	Sistema Gestion de Defectos	Lider SGD		
SC	Sistema de Contabilidad	Lider SGD		
SF	Sistema Final	Lider SGI		
SBE	Sistema de BE	Lider SBD		
SDP	Sistema de prueba	Lider SGI		
PDP	Proyecto de Pipeline	Lider SBD		
SU	Sistema Unificador	Lider SGI		
PDP2	Proyecto de Paralelismo 2	Lider SGI		
PA	Proyecto Ajax	Lider SGD		

Agregar

Figura 35. Módulo Proyectos.

GESTIÓN DE PROYECTOS Buscar en tabla:

Id	Nombre	Lider Proyecto	Editar	Borrar
SGD	Sistema Gestion de Defectos	Lider SGD		
SC	Sistema de C...			
SF	Sistema			
SBE	Sistema			
SDP	Sistema de			
PDP	Proyecto de			
SU	Sistema Un			
PDP2	Proyecto de Paralelismo 2	Lider SGI		
PA	Proyecto Ajax	Lider SGD		

NUEVO PROYECTO

Nombre:

Siglas:

Lider de Proyecto:

Figura 36. Crear Nuevo Proyecto.

GESTIÓN DE USUARIOS Buscar en tabla:

Camet	Nombre	Rol	Proyecto	Editar	Borrar
L11111	Lider SGI	LIDER_PROYECTO	SF		
E44444	Claudio Torrez	ANALISTA	SGD		
E55555	Alvaro Marciales	ANALISTA	SGD		
L00000	Lider SGD	LIDER_PROYECTO	SBE		
E66666	Analista 1	ANALISTA	PDP2		
L22222	Lider SBD	LIDER_PROYECTO	PA		

Figura 37. Módulo Usuarios.

GESTIÓN DE USUARIOS Buscar en tabla:

Camet	Nombre	Rol	Proyecto	Editar	Borrar
L11111	Lider SGI	LIDER_PROYECTO	SF		
E44444	Claudio Torrez	ANALISTA	SGD		
E55555	Alvaro Marciales	ANALISTA	SGD		
L00000	Lider SGD	LIDER			
E66666	Analista 1	ANAL			
L22222	Lider SBD	LIDER			

NUEVO USUARIO

Nombre de Proyecto:

Rol asignado:

Nombres:

Figura 38. Crear Nuevo Usuario.

CREAR DEFECTO
✕

Proyecto

Categoría

Urgencia

Impacto

Asunto

Descripción del defecto

Adjuntar archivos

Figura 39. Reportar Defecto.

[Bienvenido, Funcional Marciales]
[Cerrar Sesión](#)

INICIO
REGISTRAR DEFECTO
CONTROL DE DEFECTOS
REPORTES

Todos los Proyectos

Resúmen de Defectos

REPORTADOS POR MI
Buscar:

ID	Asunto	Prioridad	Estado	Categoría	Proyecto
No se encontraron defectos bajo este criterio					

MONITORIZADOS
Buscar:

ID	Asunto	Prioridad	Estado	Categoría	Proyecto
7	Bug Spring Security	MEDIA	ABIERTO	Externos	SGD
8	Limitar alcance de proyectos de funcional	ALTA	RE-ABIERTO	Exportar Datos	PDP2
9	Css	CRITICA	RE-ABIERTO	Defecto	SGD
5	Defecto sdp	MEDIA	ABIERTO	Defecto	SDP
4	Prueba defecto	CRITICA	CERRADO	Defecto	SGD
1	Adjuntos	CRITICA	CERRADO	Defecto	SGD
2	Problema al exportar en excel	MEDIA	ABIERTO	Exportar Datos	SDP
3	Reload content body	MEDIA	ABIERTO	Defecto	SBE

POR VALIDAR
Buscar:

ID	Asunto	Prioridad	Estado	Categoría	Proyecto
No se encontraron defectos bajo este criterio					

CERRADOS
Buscar:

ID	Asunto	Prioridad	Estado	Categoría	Proyecto
4	Prueba defecto	CRITICA	CERRADO	Defecto	SGD
1	Adjuntos	CRITICA	CERRADO	Defecto	SGD

Figura 40. Principal. Usuario Funcional.

DEFECTO No. 1

Asunto: Adjuntos **Proyecto:** SGD
Descripcion: Hola **Categoria:** Defecto
Funcional: E55555-Admin Torrez **Fecha Creacion:** 16/10/2015 06:26:23
Estado: CERRADO **Motivo:** Ya Resuelto
Prioridad: CRITICA **Impacto:** ALTA
Urgencia: ALTA **Fecha Estimada:** 16/10/2015 06:23:37
Responsable: Alvaro Marciales **Fecha Asignacion:** 16/10/2015 13:31:47
Comentarios: validado gracias **Añadir archivos adjuntos:** [Seleccionar](#)

ARCHIVOS ADJUNTOS DEL DEFECTO

Nombre de archivo	Tamano de archivo	
Capture arq tota.PNG	68.16015625KB	Descargar
defectos asignados.PNG	60.078125KB	Descargar

HISTORICO DEL DEFECTO

Fecha	Usuario	Estado	Duracion	Comentarios
16/10/2015 10:58:15	Admin Torrez	ABIERTO	0 Dias 07:03:31	
16/10/2015 18:08:03	Alvaro Marciales	A SIGNADO	0 Dias 00:00:24	Hoy realice un 10%
16/10/2015 18:08:27	Alvaro Marciales	RESUELTO	0 Dias 00:10:47	Termine Bug Login
16/10/2015 18:19:14	Admin Torrez	CERRADO	0	validado gracias

Figura 41. Detalle del Defecto.

Resumen de Defectos

SIN ASIGNAR **Buscar:**

ID	Asunto	Prioridad	Estado	Categoria	Proyecto
2	Problema al exportar en excel	MEDIA	ABIERTO	Exportar Datos	SDP
5	Defecto sdp	MEDIA	ABIERTO	Defecto	SDP
8	Limitar alcance de proyectos de funcional	ALTA	RE-ABIERTO	Exportar Datos	PDP2

ACTIVIDAD RECIENTE **Buscar:**

ID	Asunto	Prioridad	Estado	Categoria	Proyecto
8	Limitar alcance de proyectos de funcional	ALTA	RE-ABIERTO	Exportar Datos	PDP2

CERRADOS **Buscar:**

No se encontraron defectos bajo este criterio

Figura 42. Principal. Líder de Proyecto.

DEFECTO No. 2

Asunto: Problema al exportar en excel **Proyecto:** SDP

Descripción: Cuando exporto a excel falla **Categoría:** Exportar Datos

Funcional: F00000-Funcional Zero **Fecha Creación:** 16/10/2015 12:26:28

Estado: ABIERTO **Motivo:**

Prioridad: MEDIA **Impacto:** MEDIA

Urgencia: ALTA **Fecha Estimada:** 16/10/2015 12:24:07

Responsable: --- Sin Asignar ---
 --- Sin Asignar ---
 Alvaro Marciales

Fecha Asignación:

Comentarios:

Añadir archivos adjuntos:

Figura 43. Asignación. Detalle del Defecto.

Resumen de Defectos

ASIGNADOS

ID	Asunto	Prioridad	Estado	Categoría	Proyecto
No se encontraron defectos bajo este criterio					

ACTIVIDAD RECIENTE

ID	Asunto	Prioridad	Estado	Categoría	Proyecto
7	Bug Login	MEDIA	ABIERTO	Externos	SGD
6	Css	CRITICA	RE-ABIERTO	Defecto	SGD

MONITORIZADOS

ID	Asunto	Prioridad	Estado	Categoría	Proyecto
7	Bug Login	MEDIA	ABIERTO	Externos	SGD
6	Css	CRITICA	RE-ABIERTO	Defecto	SGD
5	Defecto sdp	MEDIA	ABIERTO	Defecto	SDP
1	Adjuntos	CRITICA	CERRADO	Defecto	SGD

CERRADOS

ID	Asunto	Prioridad	Estado	Categoría	Proyecto
4	Prueba defecto	CRITICA	CERRADO	Defecto	SGD
1	Adjuntos	CRITICA	CERRADO	Defecto	SGD

Figura 44. Principal. Usuario Analista.

DEFECTO No. 2

Asunto: Problema al exportar en excel **Proyecto:** SDP

Descripción: Cuando exporto a excel falla **Categoría:** Exportar Datos

Funcional: F00000-Funcional Zero **Fecha Creación:** 16/10/2015 12:26:28

Estado: ASIGNADO **Motivo:** --- Sin Asignar ---

Prioridad: MEDIA **Impacto:** --- Sin Asignar ---

Urgencia: ALTA **Fecha Estimada:** No reproducible

Responsable: Alvaro Marciales **Fecha Asignación:** No corregible

Comentarios: **Añadir archivos adjuntos:** Duplicada

--- Sin Asignar ---
 --- Sin Asignar ---
 No reproducible
 No corregible
 Duplicada
 Suspendida
 No se arreglara
 Ya Resuelto
 Indisponibilidad
 Motivo nuevo
 Motivo Desconocido
 Motivo Bien

Guardar Cambios **Cancelar**

Figura 45. Resolver. Detalle del Defecto.

Resumen de Defectos

REPORTADOS POR MI Buscar:

ID	Asunto	Prioridad	Estado	Categoría	Proyecto
No se encontraron defectos bajo este criterio					

MONITORIZADOS Buscar:

ID	Asunto	Prioridad	Estado	Categoría	Proyecto
7	Bug Spring Security	MEDIA	ABIERTO	Externos	SGD
8	Limitar alcance de proyectos de funcional	ALTA	RE-ABIERTO	Exportar Datos	PDP2
8	Css	CRITICA	RE-ABIERTO	Defecto	SGD
5	Defecto sdp	MEDIA	ABIERTO	Defecto	SDP
4	Prueba defecto	CRITICA	CERRADO	Defecto	SGD
1	Adjuntos	CRITICA	CERRADO	Defecto	SGD
2	Problema al exportar en excel	MEDIA	ABIERTO	Exportar Datos	SDP
3	Reload content body	MEDIA	ABIERTO	Defecto	SBE

POR VALIDAR Buscar:

ID	Asunto	Prioridad	Estado	Categoría	Proyecto
No se encontraron defectos bajo este criterio					

CERRADOS Buscar:

ID	Asunto	Prioridad	Estado	Categoría	Proyecto
4	Prueba defecto	CRITICA	CERRADO	Defecto	SGD
1	Adjuntos	CRITICA	CERRADO	Defecto	SGD

Figura 46. Principal. Usuario Funcional.

DEFECTO No. 2

Asunto: Problema al exportar en excel Proyecto: SDP

Descripción: Cuando exporto a excel falla Categoría: Exportar Datos

Funcional: F00000-Funcional Zero Fecha Creación: 16/10/2015 12:26:28

Estado: RESUELTO Motivo: Duplicada

Prioridad: MEDIA Impacto: MEDIA

Urgencia: ALTA Fecha Estimada: 16/10/2015 12:24:07

Responsable: Alvaro Marciales Fecha Asignación: 26/10/2015 10:44:10

Comentarios: Corregido formato de exportacion a .xlsx

Añadir archivos adjuntos:

Figura 47. Validación. Detalle del Defecto.

Urgencia: ALTA Fecha Estimada: 16/10/2015 12:24:07

Responsable: Alvaro Marciales Fecha Asignación: 26/10/2015 10:44:10

Comentarios: Corregido formato de exportacion a .xlsx

Añadir archivos adjuntos:

ARCHIVOS ADJUNTOS DEL DEFECTO

Nombre de archivo	Tamaño de archivo	
adminCreateProy.PNG	72.2255859375KB	Descargar

HISTORICO DEL DEFECTO

Fecha	Usuario	Estado	Duracion	Comentarios
16/10/2015 17:02:57	Funcional Zero	ABIERTO	9 Dias 22:11:13	
26/10/2015 15:14:10	Lider SGI	ASIGNADO	0 Dias 00:05:23	Corregir exportacion de archivos en formato .xls a .xlsx
26/10/2015 15:19:33	Alvaro Marciales	RESUELTO	0 Dias 00:02:12	Corregido formato de exportacion a .xlsx

Figura 48. Detalle del Defecto. Histórico de Estatus.

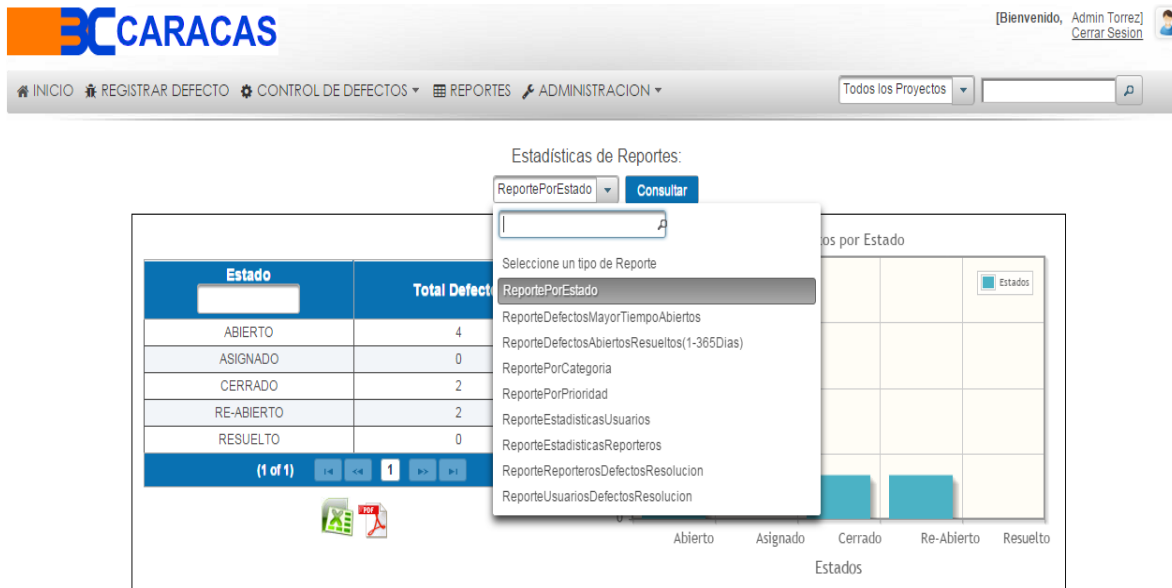


Figura 49. Menú de opciones para muestra de reportes.



Figura 50. Reporte y Gráfico de defectos por estado.

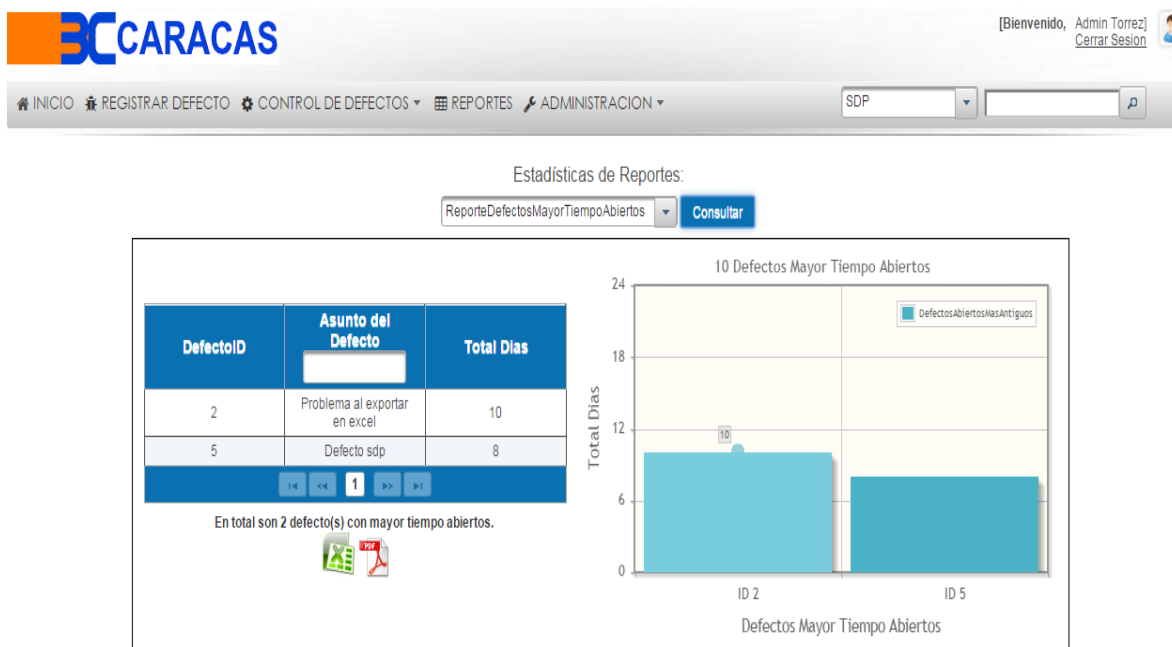


Figura 51. Reporte y Gráfico de defectos con mayor tiempo en estado “Abierto”.

2.1.4. FASE DE TRANSICION

Esta fase detalla la etapa de implantación y el plan de pruebas del sistema, entre las actividades realizadas se encuentran:

- **Carga inicial del sistema**
Se prepara la información necesaria para la carga inicial de los datos del sistema, por ejemplo: roles de usuarios, tipos de prioridad, impacto y urgencia.
- **Plan de integración de sistemas**
Se prepara la integración del sistema con las herramientas utilizadas en el banco tales como servidores de web, servidores de autenticación y servidores de correo.
- **Plan de pruebas integradas**
Para la aplicación web se realizó el siguiente esquema de pruebas.

Configuración general:

:

- Conexión con Web Services BEASA e iniciar sesión en la aplicación exitosamente.
- Creación de Proyectos, Categorías y Motivos de Rechazo.
- Creación de usuarios con roles: Analista y Líder de Proyecto y agregarlos a la base de datos.

- Mostrar opciones y acciones según el rol del usuario.
- Modificar usuarios, proyectos, categorías y motivos en la base de datos.
- Validación de campos obligatorios en la agregación y modificación de usuarios, proyectos, categorías y motivos de rechazo.
- Prueba filtro general de búsqueda de defectos en el menú principal conjuntamente con filtros secundarios.
- Visualización de datos relevantes del usuario autenticado.
- Registro correcto de histórico de defectos durante su ciclo de vida.
- Carga y descarga de archivos adjuntos de manera exitosa.
- Cierre de sesión exitoso.

Flujo de Defectos:

- Creación de defectos.
- Verificar flujo de estados de defectos y cumplimiento de todos los casos posibles.
- Conexión con BECA y comprobar notificación de correo por cada cambio de estado en el flujo del defecto.

Reportes y Gráficos:

- Realizar evaluación de desempeño individual (roles de usuario).
- Realizar evaluación de desempeño por proyecto.
- Realizar evaluación de desempeño por intervalos de tiempo predefinidos.
- Realizar evaluación de desempeño por estado de defectos.
- Uso de filtros de búsqueda de información por cada reporte.
- Exportar información de reportes en formato .pdf y .xlsx (Documentos PDF y Excel).

○ **Elaboración del manual de usuario**

Una vez finalizadas exitosamente las pruebas se procede a la elaboración de manuales para la formación de los usuarios del sistema.

2.2. APLICACIÓN DE ITIL AL SISTEMA

El objetivo del estándar de buenas prácticas ITIL para este proyecto es la gestión correcta y eficaz de los defectos en los proyectos de la organización financiera detectando así alteraciones en la etapa de desarrollo de dichos proyectos para su posterior resolución. Bajo este contexto se tomó como referencia la gestión de incidencias, como lo sugiere ITIL dentro de la fase de Servicio de Operación. Para esta gestión se definen los siguientes privilegios por cada rol de usuario los cuáles se explican a continuación:

▪ **Privilegios para Roles de Usuario:**

El sistema provee los siguientes privilegios a los roles siguientes:

El usuario Administrador:

- Administración general del sistema.
- Reportar defectos.
- Resolver defectos.
- Validación de defectos para su posterior reapertura en caso de no aprobación o cierre exitoso del caso reportado.

El usuario funcional:

- Reportar defectos.
- Consultar el motivo del cierre de su defecto reportado.
- Reabrir el defecto en caso de validación y rechazo de resolución del defecto.
- Cerrar defectos.

El usuario analista:

- Consultar la base de datos de soluciones para ver si existe alguna solución pre-establecida que se pueda aplicar al defecto reportado (duplicado de defectos).
- Actualizar la base de datos con la nueva técnica aplicada.
- Resolver defecto que le ha sido asignado.

El usuario líder de proyecto:

- Gestión de los usuarios analistas de su proyecto.
- Configurar el tiempo de resolución establecido por el SLA dependiendo de la prioridad definida por el defecto.

- Consultar las estadísticas para visualizar el tiempo de resolución de los defectos reportados a través del histórico de defectos.
- Asignar defecto reportado a un analista en particular.
- Resolver defecto asignado a sí mismo en caso de ser necesario.

El usuario gerente de proyecto:

- Consultar una serie de reportes tales como Defectos por estado, por Categoría y Prioridad, Estadísticas de Usuarios con Rol Analistas y Reportero, Defectos con mayor tiempo en estado abierto, entre otros reportes.

La aplicación de los procesos para la gestión de defectos que se tomaron en cuenta según las buenas prácticas de ITIL se explica a continuación:

▪ **Proceso de Gestión de Prioridades de defectos en el sistema:**

Cuando se atienden varios defectos en un proyecto, es necesario establecer prioridades para la resolución de los mismos.

Para hacer una evaluación subjetiva se deben discutir con el usuario, los siguientes criterios:

- Urgencia del Defecto: la demora aceptable para el usuario para resolver un defecto.
- Impacto del Defecto: determina la importancia del defecto dependiendo de cómo éste afecta el tiempo y costo en la vida del proyecto.

En la siguiente tabla 4 se muestra el modo de asignación de prioridades que tiene el sistema. Esta es medida a través del impacto y urgencia proporcionado por el usuario a partir del defecto reportado.

		Impacto		
		Alto	Medio	Bajo
Urgencia	Alto	Critico	Alto	Medio
	Medio	Alto	Medio	Bajo
	Bajo	Medio	Bajo	Planificado

Tabla 4. Prioridades del sistema.

En este ejemplo de prioridades, se define Crítica como la prioridad más alta y Planificada como la más baja, siendo Crítica un defecto que se debe resolver lo más pronto posible y Planificada como un defecto que no afecta en el comportamiento del sistema pero se pueden realizar mejoras.

▪ **Proceso de Registro y Clasificación:**

La admisión y registro de un defecto correspondiente a un proyecto en particular es el primer paso necesario para una correcta gestión del mismo. Por su parte el sistema posee el módulo para reportar defectos y además permite clasificar los defectos por categoría y motivos de rechazo (en caso de ser rechazado su desarrollo) para obtener una gestión más organizada de los defectos de un proyecto.

Las categorías son creadas libremente a través del módulo de administración. En cambio los motivos de rechazo de igual forma pueden ser creados libremente a través del módulo de administración pero el sistema por defecto posee los siguientes motivos de rechazo:

- No-Reproducible: El defecto no se puede reproducir.
- No-Corregible: El defecto no se puede corregir.
- Duplicado: El defecto se encuentra duplicado.
- Suspendido: El defecto fue suspendido.
- No Se Arreglara: El defecto no se puede arreglar.

Adicionalmente dentro del proceso de clasificación el sistema contiene procedimientos almacenados que se encargan del monitoreo y seguimiento de los cambios realizados en un defecto durante su ciclo de vida, lo cual permite ver la evolución de cambios leves o constantes por los que ha pasado dicho defecto.

▪ **Proceso de Resolución y Cierre:**

Cuando un analista realiza una resolución de un defecto que le ha sido asignado, el sistema provee la opción de adjuntar archivos para obtener un control eficaz de los defectos a través de captura de pantallas tales como errores por consola, errores en el navegador, bugs en algún módulo de proyecto, entre otros casos.

Una vez que el defecto es resuelto por el analista, el líder de proyecto o usuario

funcional procede a realizar la validación de aprobación o desaprobación de la resolución de dicho defecto. En caso de desaprobación, el defecto es re-abierto para su posterior asignación nuevamente, en caso contrario, el líder de proyecto o usuario funcional asigna un motivo de resolución y el defecto es cerrado culminando así su ciclo de vida.

▪ **Proceso de Escalado:**

El sistema no presenta reglas de escalado ya que cuando un defecto es asignado a un usuario con rol analista y éste no cumple con la resolución, el defecto es re-abierto por el líder de proyecto para su posterior asignación a otro analista repitiendo el mismo flujo hasta que el defecto sea resuelto. Este escenario demuestra que el conjunto de analistas tienen un mismo nivel de jerarquía motivo por el cual no se requiere de procesos de escalabilidad entre usuarios.

El estándar de buenas prácticas ITIL también posee recomendaciones con respecto a los roles y el flujo de estados por los que debe pasar un defecto, estos fueron tomados en cuenta y representados en la siguiente explicación de las fases de la metodología RUP conjuntamente con los requerimientos de la organización.

3. CONCLUSIONES

Luego de haber desarrollado el Sistema para la Gestión de Defectos se cumplieron todos los objetivos planteados para la solución de la problemática de manera satisfactoria los cuales se listan a continuación:

- El producto obtenido ha sido un sistema de administración y gestión de defectos que cumple con la funcionalidad mínima necesaria que requiere este tipo de sistemas añadiendo una serie de características interesantes como la asignación automática de prioridades y control de defectos basándose en prácticas para la gestión de servicios TI según ITIL.
- Se evaluaron los procesos actuales e identificaron todos los campos que serán registrados en un defecto, tanto los obligatorios como los opcionales.
- Se definió el flujo de trabajo y los roles en la asignación de los defectos, los estados que pueden tener y las posibles acciones a ejecutar en los mismos.
- Se llevó a cabo la implementación del proceso de invocación al Servicio de Autenticación de la organización financiera (BEASA), con el cual se permite el ingreso al sistema y se obtendrá el menú de opciones que puede utilizar un usuario.
- Se desarrollaron los módulos de Reportes que muestren estadísticas y gráficos de los defectos, y los módulos de administración de Proyectos, categorías, usuarios y motivos de rechazo.
- Se cumplieron los estándares y buenas prácticas durante el desarrollo de la aplicación, así como la elaboración de su documentación técnica.
- Queda abierta la posibilidad de ampliar el alcance de este proyecto orientado a la gestión de incidencias en un ambiente de producción.
- Para las notificaciones por correo se aplicó el consumo de un servicio web (BECA) el cual se encarga de realizar todas las funciones vía email dentro de la organización.
- Para la organización financiera le es de gran utilidad un sistema gestor de defectos ya que dentro de su entorno laboral se encuentran involucradas actividades como el desarrollo de proyectos, por lo cual pueden llevar un control riguroso de defectos que se generen durante la etapa de desarrollo de cualquier sistema, agilizando así la resolución de los mismos a través de ciertos flujos predefinidos encargados a usuarios del mismo grupo de trabajo y obteniendo resultados más satisfactorios en las entregas previas al ambiente de producción.

BIBLIOGRAFIA

- [1] Página Oficial de la Organización financiera. **Misión, visión y Valores.** Fuente: <http://www.bancoexterior.com/el-banco/mision-vision-y-valores> (Consultado el: 27/03/2014).
- [2] Página Oficial de la Organización financiera. **54 años de solidez, Evolución y Compromiso con el país.** Fuente: http://www.bancoexterior.com/files/files/ce405c163f35c52ede7a494b9b0a5ad1/54_aniversario.pdf (Consultado el: 25/03/2014).
- [3] Página Oficial de la Organización financiera. **Vanguardia Tecnológica para ofrecer más y mejores servicios.** Fuente: http://www.bancoexterior.com/files/files/a79671883728ffd6ce329b5876efa652/np_exterior_modernizacion_y_actualizacion_tecnologica.pdf (Consultado el: 28/03/2014).
- [4] GrupoNADD(2012). **Metodología RUP.** Fuente: <http://rupmetodologia.blogspot.com/> (Consultado el: 02/01/2014).
- [5] WIKIPEDIA. (22/11/2013). **Proceso Unificado de Racional.** Fuente: http://es.wikipedia.org/wiki/Proceso_Unificado_de_Racional (Consultado el: 07/01/2014).
- [6] Página Oficial de MANTIS. **Mantis Bug Tracker.** Fuente: <http://www.mantisbt.org/documentation.php> (Consultado el: 07/01/2014).
- [7] Carlos Gallo (2014). **Mantis Bug Tracker.** Fuente: <http://prezi.com/ptzjp4vvjsq1/mantis-bug-tracker/> (Consultado el: 08/01/2014).
- [8] Bilib(20/04/2011). **Mantis Bug Tracker.** Fuente: <http://www.bilib.es/recursos/analisis-de-aplicaciones/analisis/doc/analisis-de-aplicacion-mantis-bug-tracker/> (Consultado el: 09/01/2014).
- [9] OSIATIS S.A. (2009). **ITIL-Gestión de Servicios TI.** Fuente: http://itilv3.osiatis.es/operacion_servicios_TI/gestion_incidentes.php (Consultado el: 27/04/2014).

- [10] IT Process Maps(2013). *ITIL Gestión de Incidentes*. Fuente: [http://wiki.es.it-processmaps.com/index.php/ITIL Gestion de Incidentes](http://wiki.es.it-processmaps.com/index.php/ITIL_Gestion_de_Incidentes) (Consultado el: 05/02/2014).
- [11] Página Oficial de MICROSOFT. *Sql Server Books Online*. Fuente: [http://technet.microsoft.com/library/ms130214\(v=sql.100\)](http://technet.microsoft.com/library/ms130214(v=sql.100)) (Consultado el: 18/02/2014).
- [12] SLIDESHARE(2010). *Sql Server*. Fuente: <http://www.slideshare.net/JJRP80/sql-server-3390264> (Consultado el: 10/02/2014).
- [13] Página Oficial de ORACLE. *Javadoc Tool*. Fuente: <http://www.oracle.com/technetwork/java/javase/documentation/javadoc-137458.html> (Consultado el: 10/02/2014).
- [14] WIKIPEDIA(2013). *Javadoc*. Fuente: <http://es.wikipedia.org/wiki/Javadoc> (Consultado el: 10/02/2014).
- [15] Página Oficial de SPRING. *Spring Framework*. Fuente: <http://projects.spring.io/spring-framework/> (Consultado el: 10/02/2014).
- [16] Página Oficial de SPRING. *SPRING FRAMEWORK Reference Documentation*. Fuente: <http://docs.spring.io/spring/docs/4.0.0.RELEASE/spring-framework-reference/htmlsingle/> (Consultado el: 10/02/2014).
- [17] (WIKIPEDIA, 2014). **JAVASERVER FACES**. Fuente: [http://es.wikipedia.org/wiki/JavaServer Faces](http://es.wikipedia.org/wiki/JavaServer_Faces) (Consultado el: 10/02/2014).
- [18] Página Oficial de PRIMEFACES. **PRIMEFACES**. Fuente: <http://www.primefaces.org/documentation.html> (Consultado el: 10/02/2014).
- [19] TRAKER SUITE (2013). **TRAKER SUITE and ITIL**. Automation Centre 435 East 9th Street Tucson, AZ 85705 / Fuente: http://www.trackersuite.com/pdf/ts_itil.pdf (Consultado el: 12/02/2014).
- [20] Iglesias, Beatriz. **Análisis, Diseño e implementación de una herramienta de gestión de niveles de servicio en .NET integrada con gestión de incidencias** (OTRS: SLA, UC, SIP, Empresas y Usuarios). Madrid, España. 2010. Fuente: http://e-archivo.uc3m.es/bitstream/handle/10016/10525/PFC_BeatrizIglesiasPino.pdf?sequence=1 (Consultado el: 05/05/2014)

- [21] GENBETADEV (2011). **Spring framework: Introducción**. Fuente: <http://www.genbetadev.com/java-j2ee/spring-framework-introduccion> (Consultado el: 05/05/2014)
- [22] VAN HAREN PUBLISHING (2008). **Service Operation Based on ITIL V3: A Management Guide**. Fuente: http://books.google.co.ve/books?id=IWhG9U97fn4C&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false (Consultado el: 27/04/2014)
- [23] JAVA TUTORIALES (2010). **Spring 3. Contenedores de IoC e Inyección de Dependencias**. Fuente: <http://www.javatutoriales.com/2010/12/contenedores-de-ioc-e-inyeccion-de.html> (Consultado el 23/06/2014)
- [24] TCPSI (2014). **Introducción a Consultoría ITIL**. Fuente: http://www.tcpsi.com/servicios/gestion_ti.htm (Consultado el 23/06/2014)
- [25] Héctor Costa (2011). **Fundamentos de Spring: Los Bean**. Fuente: <http://blog.hcosta.info/fundamentos-de-spring-los-beans/> (Consultado el 26/06/2014)
- [26] (WIKIPEDIA, 2014). **BEAN**. Fuente: <http://es.wikipedia.org/wiki/Bean> (Consultado el 26/06/2014)
- [27] (Godoy, Gomez y Rubio 2004) **Análisis y Gestión del Desarrollo del Software**. Fuente: http://usuaris.tinet.cat/jgodoy/Resumenes/Superior/agds_1.pdf (Consultado el 26/02/2015)
- [28] (WebFinance, 2015) **Tecnologías de la información**. Fuente: <http://www.businessdictionary.com/definition/information-technology-IT.html> (Consultado el 26/02/2015)
- [29] (DeGerencia, 2014) **Importancia de las TI en las empresas**. Fuente: http://www.degerencia.com/tema/tecnologia_de_informacion (Consultado el 14/03/2014)
- [30] (Osiatís S.A., 2010) **Ciclo de vida de los servicios de TI**. Fuente: http://itilv3.osiatís.es/ciclo_vida_servicios_TI.php (Consultado el 14/03/2014)

[31] (GlobeTesting, 2014) Defecto. Fuente: <http://en.globetesting.com/glosario/defecto/> (Consultado el 11/03/15)

[32] (Diario de un Tester, 2014) Flujo de los defectos. Fuente: http://www.diariodeuntester.es/2014/04/el-flujo-de-los-defectos_11.html (Consultado el 11/03/15)

[33] <http://rolandojaldin.blogspot.com/search/label/Metodolog%C3%ADa%20RUP>

[34] (INFORMATION TECHNOLOGY STANDARD, 2014) Defect definition. Fuente: http://www.dhs.state.pa.us/cs/groups/webcontent/documents/document/p_031767.pdf (Consultado el 11/03/15)

ANEXOS

Manual de Usuario:

El sistema consta de los siguientes módulos:

1. Vista Principal: En este módulo se presentan tablas de vista previa de los defectos de acuerdo al rol que posea un usuario autenticado. Estas tablas son:
 - a. Defectos Cerrados.
 - b. Actividad Reciente de los Defectos.
 - c. Defectos Reportados por mí.
 - d. Defectos por Validar.
 - e. Defectos Monitorizados.
2. Registrar Defecto: Este módulo se encarga de mostrar un cuadro de diálogo el cual contiene un formulario para llenar ciertos campos para el registro de un defecto en el sistema.
3. Control de Defectos: Este módulo se encarga de la gestión de defectos tales como Asignar, Resolver, Validar y Rechazar Defectos.
4. Reportes: Este módulo contiene todos los tipos de reportes relacionados a los defectos en general. Algunos de ellos son:
 - a. Reporte de Defectos por estado.
 - b. Reporte de Defectos con mayor tiempo abierto.
 - c. Reporte de Defectos por categoría, entre otros.
5. Administración: Este módulo se encarga de permitir realizar operaciones CRUD (Create, Red, Update, Delete) de objetos como Proyectos, Usuarios, Categorías y Motivos de Rechazo.
6. Mi Cuenta: Este módulo se encarga de mostrar la información referente al usuario que se autenticó en el sistema.

Los roles de usuario se clasifican de la siguiente manera:

- **Administrador:** Este tipo de rol de usuario tiene los siguientes privilegios:
 - Resolver Defecto.
 - Asignar Defecto.
 - Reportar Defecto.
 - Validar o Rechazar Defecto.

- **Líder de Proyecto:** Los privilegios de este rol de usuario son los siguientes:
 - Resolver Defectos.
 - Asignar Defectos.

- **Analista:** Los permisos de este tipo de rol de usuario son:
 - Resolver Defecto.

- **Usuario Funcional:** Las permisologías que le son otorgadas a este tipo de rol son:
 - Reportar Defecto.
 - Validar o Rechazar Defecto.

- **Gerente o Visualizador:** Este tipo de rol comprende los siguientes privilegios:
 - Visualizar Reportes de Defectos.

A continuación se muestra el flujo del proyecto a través de cada rol de usuario con interfaces:

- **Administrador:**

El usuario con rol de Administrador procede a iniciar sesión en la aplicación (Figura 51).



The image shows a login form with an orange header bar containing the text "Autenticación de Usuarios". Below the header, there are two input fields. The first is labeled "Usuario" and contains the text "admin". The second is labeled "Clave" and contains six asterisks "*****". Below these fields are two blue buttons with white text: "Ingresar" and "Borrar".

Figura 52. Login Aplicación

Como muestra la Figura 52, al mostrar la interfaz principal este tipo de rol de usuario solo puede visualizar 2 tablas las cuales son:

- Tabla de Actividad Reciente.
- Tabla de Defectos Cerrados.



Figura 53. Pantalla principal

El Administrador puede reportar un defecto a través de la opción del menú principal “Registrar Defecto” (Figura 53).

The screenshot shows the 'CREAR DEFECTO' form. It has a blue header with the title and a close button. The form contains several fields: 'Proyecto' (SBE), 'Categoría' (Sintaxis), 'Urgencia' (MEDIA), and 'Impacto' (MEDIA), all with dropdown menus. The 'Asunto' field contains 'Error de Sintaxis'. Below it is a larger text area for 'Descripcion del defecto' containing 'Error de Sintaxis en modulo Tesorería.'. At the bottom, there is an 'Adjuntar archivos' section with a 'Seleccionar' button. At the very bottom, there are 'Crear' and 'Borrar' buttons.

Figura 54. Formulario Crear Defecto

Además, en la figura 54, se puede notar que el Administrador tiene la opción de visualizar y gestionar los defectos a validar, resolver o asignar a través de la opción “Control de Defectos”.



Por cada opción de validar, resolver o asignar defectos se muestra la tabla que

Figura 55. Opciones Control de Defectos

corresponde al usuario según su rol o filtro de proyecto.

- Defectos a Validar: Se muestran los defectos que ya han sido resueltos y esperan por la validación del reportero del defecto (Figura 55).



Figura 56. Tabla de Defectos Resueltos y Pendientes por Validar

- Defectos a Resolver: Se muestran los defectos que ya han sido asignados hacia un analista y esperan por ser resueltos (Figura 56).

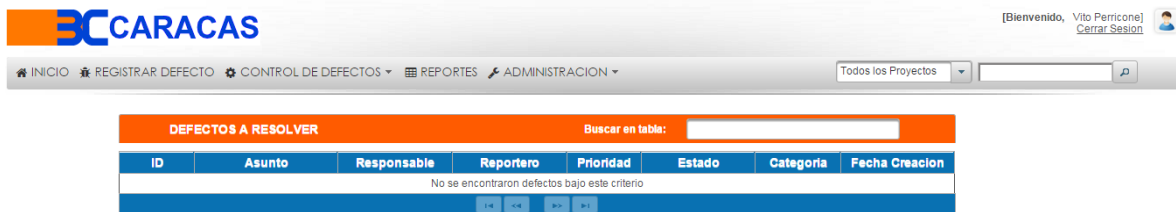


Figura 57. Tabla de Defectos Asignados y Pendientes por Resolver

Todos los usuarios pueden ver el detalle de un defecto haciendo click sobre el ID del defecto, esto conlleva a que se despliegue una interfaz con la vista en detalle del defecto (Figura 57):

DEFECTO No. 1

Asunto:	Adjuntos	Proyecto:	SGD
Descripción:	Hola	Categoría:	Defecto
Funcional:	E55555-Admin Torrez	Fecha Creacion:	16/10/2015 06:26:23
Estado:	CERRADO	Motivo:	Ya Resuelto
Prioridad:	CRITICA	Impacto:	ALTA
Urgencia:	ALTA	Fecha Estimada:	16/10/2015 06:23:37
Responsable:	Alvaro Marciales	Fecha Asignacion:	16/10/2015 13:31:47
Comentarios:	<input type="text" value="validado gracias"/>		

Añadir archivos adjuntos:

ARCHIVOS ADJUNTOS DEL DEFECTO

Nombre de archivo	Tamaño de archivo	
Capture arq total.PNG	68.16015625KB	Descargar
defectos asignados.PNG	60.078125KB	Descargar

HISTORICO DEL DEFECTO

Fecha	Usuario	Estado	Duracion	Comentarios
16/10/2015 10:58:15	Admin Torrez	ABIERTO	0 Dias 07:03:31	
16/10/2015 18:08:03	Alvaro Marciales	A SIGNADO	0 Dias 00:00:24	Hoy realice un 10%
16/10/2015 18:08:27	Alvaro Marciales	RESUELTO	0 Dias 00:10:47	Termine Bug Login
16/10/2015 18:19:14	Admin Torrez	CERRADO	0	validado gracias

Figura 58. Detalle del Defecto

El administrador puede elegir ver los diferentes reportes respecto a su preferencia (Figura 58).

Estadísticas de Reportes:

ReportePorEstado

Estado	Total Defectos
ABIERTO	4
ASIGNADO	0
CERRADO	2
RE-ABIERTO	2
RESUELTO	0

Seleccione un tipo de Reporte

- ReportePorEstado
- ReporteDefectosMayorTiempoAbiertos
- ReporteDefectosAbiertosResueltos(1-365Dias)
- ReportePorCategoria
- ReportePorPrioridad
- ReporteEstadisticasUsuarios
- ReporteEstadisticasReporteros
- ReporteReporterosDefectosResolucion
- ReporteUsuariosDefectosResolucion

Estados por Estado

Abierto Asignado Cerrado Re-Abierto Resuelto

Figura 59. Opciones de Reportes

Resultando en tablas de reportes y gráficos que pueden ser exportables en formato PDF y Excel, como muestra la Figura 59.

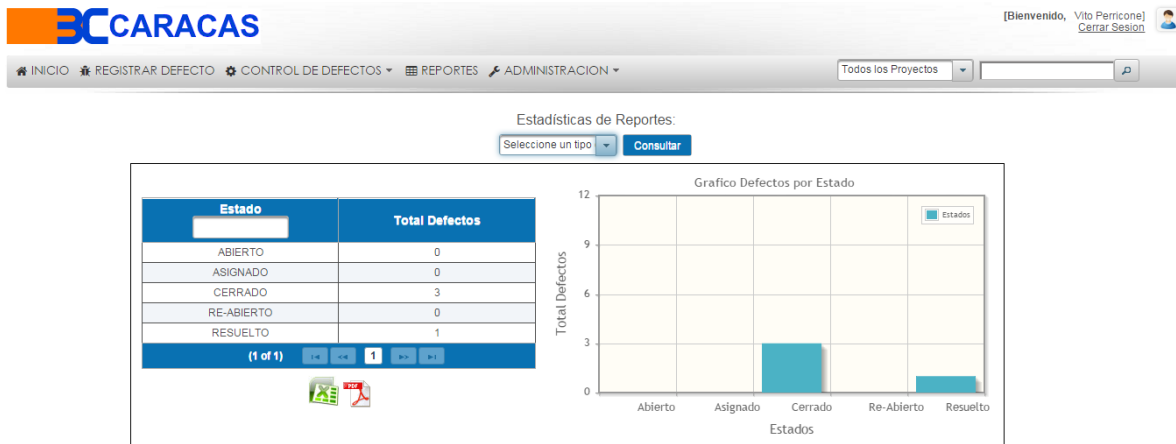


Figura 60. Resultados Reportes por Estado de Defecto

Adicionalmente el usuario Administrador puede acceder a distintos módulos de Administración del sistema (Figura 60) entre ellos:

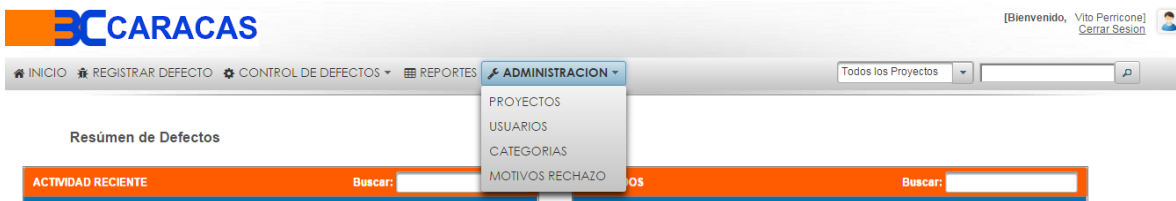


Figura 61. Administración del Sistema

- Gestión de Proyectos del sistema

GESTIÓN DE PROYECTOS Buscar en tabla:

Id	Nombre	Lider Proyecto	Editar	Borrar
SGD	Sistema Gestion de Defectos	Lider SGD		
SC	Sistema de Contabilidad	Lider SGD		
SF	Sistema Final	Lider SGI		
SBE	Sistema de BE	Lider SBD		
SDP	Sistema de prueba	Lider SGI		
PDP	Proyecto de Pipeline	Lider SBD		
SU	Sistema Unificador	Lider SGI		
PDP2	Proyecto de Paralelismo 2	Lider SGI		
PA	Proyecto Ajax	Lider SGD		

Figura 62. Gestión de Proyectos

- Gestión de Usuarios

GESTIÓN DE USUARIOS Buscar en tabla:

Camet	Nombre	Rol	Proyecto	Editar	Borrar
L11111	Lider SGI	LIDER_PROYECTO	SF		
E44444	Claudio Torrez	ANALISTA	SGD		
E55555	Alvaro Marciales	ANALISTA	SGD		
L00000	Lider SGD	LIDER_PROYECTO	SBE		
E66666	Analista 1	ANALISTA	PDP2		
L22222	Lider SBD	LIDER_PROYECTO	PA		

Figura 63. Gestión de Usuarios

- Gestión de Categorías



Figura 64. Gestión de Categorías

- o Gestión de Motivos de Rechazo



Figura 65. Gestión de Motivos de Rechazo

Todos los usuarios pueden ver sus datos de cuenta personal a través del vínculo de la imagen superior derecha (Figura 65):



Figura 66. Información de la Cuenta

Adicionalmente la aplicación posee en el menú un Combo Box de opciones de proyectos (Figura 66) para filtrar los datos de las tablas y también está la opción de búsqueda al lado del Combo Box de opciones de proyectos:

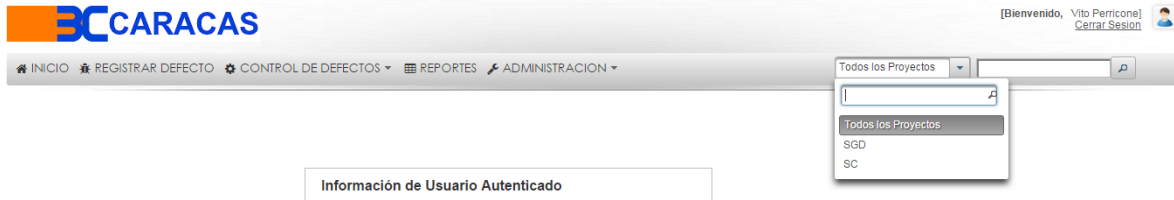


Figura 67. Filtro de Proyectos.

Y por último está la opción “Cerrar Sesión” para salir del sistema (Figura 67).



Figura 68. Cerrar Sesión.

- **Líder de Proyecto:** A diferencia del usuario Administrador, el Líder no puede validar ni reportar defectos.
- **Analista:** El Analista sólo puede resolver efectos.
- **Usuario Funcional:** El Usuario Funcional sólo reporta y valida defectos.
- **Gerente o Visualizador:** El gerente sólo visualiza los reportes relacionado a los defectos en general.