

**UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS**



**DESARROLLO DE UNA APLICACIÓN WEB BAJO SOFTWARE LIBRE
PARA LA AUTOMATIZACIÓN DE LA GESTIÓN DE LAS ALARMAS
GENERADAS POR UNA HERRAMIENTA DE MONITOREO DE EQUIPOS
PARA LA SUPERINTENDENCIA DE MANTENIMIENTO DE LA
PLATAFORMA DE LA GERENCIA DE AUTOMATIZACIÓN
INFORMÁTICA Y TELECOMUNICACIONES DE PDVSA PUERTO LA
CRUZ**

Realizado por:

Cobos Ramírez, Esther María

**Trabajo de Grado presentado como requisito parcial para optar al título de:
INGENIERO EN COMPUTACIÓN**

Puerto La Cruz, Diciembre 2010

UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



**DESARROLLO DE UNA APLICACIÓN WEB BAJO SOFTWARE LIBRE
PARA LA AUTOMATIZACIÓN DE LA GESTIÓN DE LAS ALARMAS
GENERADAS POR UNA HERRAMIENTA DE MONITOREO DE EQUIPOS
PARA LA SUPERINTENDENCIA DE MANTENIMIENTO DE LA
PLATAFORMA DE LA GERENCIA DE AUTOMATIZACIÓN
INFORMÁTICA Y TELECOMUNICACIONES DE PDVSA PUERTO LA
CRUZ**

Revisado y Aprobado por:

Ing. Gabriela Veracierta
Tutor Académico

Ing. Carlimar Pinto
Tutor Industrial

Puerto La Cruz, Diciembre 2010

UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



**DESARROLLO DE UNA APLICACIÓN WEB BAJO SOFTWARE LIBRE
PARA LA AUTOMATIZACIÓN DE LA GESTIÓN DE LAS ALARMAS
GENERADAS POR UNA HERRAMIENTA DE MONITOREO DE EQUIPOS
PARA LA SUPERINTENDENCIA DE MANTENIMIENTO DE LA
PLATAFORMA DE LA GERENCIA DE AUTOMATIZACIÓN
INFORMÁTICA Y TELECOMUNICACIONES DE PDVSA PUERTO LA
CRUZ**

Jurado Calificador:

Ing. Gabriela Veracierta

Tutor Académico

Ing. Claudio Cortinez

Jurado Principal

Ing. Rhonald Rodríguez

Jurado Principal

Puerto La Cruz, Diciembre 2010

RESOLUCIÓN

De acuerdo al Artículo 41 del reglamento de Trabajo de Grado:

“Los Trabajos de Grado son de exclusiva propiedad de la Universidad de Oriente y sólo podrán ser utilizados a otros fines con el consentimiento del Consejo de Núcleo respectivo, quién debiera participarlos al Consejo Universitario para su autorización”.

DEDICATORIA

Primeramente a Dios y a mi hermano mayor Jesucristo, por mostrarme el camino que debo seguir.

A mis padres María Ramírez y José Camacho, que ya no están presentes pero sus consejos, esfuerzo y aliento me ayudaron a saltar obstáculos, y gracias a eso soy lo que soy. Para ustedes mis eternos agradecimientos. “Los amo”.

A mis hermanos Crismar, Josué, Joel, Cristal y Cinthia que a pesar de los problemas siempre hemos estado unidos y su apoyo ha sido un gran impulso para mi: “Los quiero hermanitos”.

A mi esposo bello y gran amor, Deivis, gracias por estar en todo momento a mi lado, por apoyarme y ayudarme, a su familia porque a pesar de no pertenecer a ella me han tratado como a una hija.

A mi papa Criswet Cobos, que a pesar de todo de una forma u otra me apoyo durante mi carrera, a mi abuela Carmela Calles que a pesar de la distancia siempre ha estado apoyándome incondicionalmente.

A la familia payares, en especial a mi tía Oslinda, gracias por recordarme siempre “que no estoy sola que a pesar de que mi madre no esta, estas tu”, gracias tía que Dios te bendiga.

A todos mis hermanos en cristo, por recordarme cada día que Dios me ama y que pase lo que pase soy más que vencedora en cualquier situación.

A mis “compañeras de clases” Carolina, Luissane y Joselin, gracias por la ayuda y todo el apoyo que me han brindado, “Sigán adelante muchachas”.

A Carlimar Pinto, Nubia Rodríguez, Cesar Mata y Carolina Rojas, excelentes asesores y compañeros de trabajo, gracias por su apoyo incondicional y por preocuparse siempre por mí, sin ustedes este proyecto no sería lo que es.

AGRADECIMIENTOS

Agradezco a mi padre, Dios todo poderoso por darme sabiduría, entendimiento e inteligencia y darme toda la fuerza necesaria para seguir adelante, gracias por tu apoyo padre amado y por todas las bendiciones y las riquezas que tengo a tu lado.

A mis padres por enseñarme todo lo que sé, porque siempre confiaron en mi, y dieron todo cuanto estuvo a su alcance.

A la familia que labora en Petróleos de Venezuela S.A (PDVSA), por darme la oportunidad de realizar este proyecto en sus instalaciones.

A el Centro Integral de Monitoreo Oriente (CIMOR) por facilitarme toda la información necesaria para el desarrollo del proyecto.

Al Ingeniero Luís Gonzáles quien mostró gran interés para el desarrollo de este proyecto y brindo información esencial para su desarrollo.

A la señora Miosotis Millán, por confiar en mi y ofrecerme cuanto estaba a su alcance.

Al profesor José Luís Bastardo por ayudarme en los momentos que lo necesite, invaluable su ayuda y total disposición.

A mi asesora Gabriela Veracierta, por todo el tiempo dedicado a corregir el proyecto y aclarar mis dudas.

A todas aquellas personas que no nombro pero que de una u otra forma contribuyeron en mis logros.

RESUMEN

La Superintendencia de Mantenimiento de la Plataforma (MAP), se encarga de establecer políticas y lineamientos de mantenimiento para la gerencia de Automatización, Informática y Telecomunicaciones (AIT), y poder lograr de ésta manera la prestación de un mejor servicio. El Centro Integral de Monitoreo Oriente (CIMOR) se encarga de garantizar la calidad y disponibilidad de la Plataforma de AIT, mediante el análisis y comprobación de los activos, esto se realiza a través de herramientas que se encargan de monitorear los equipos de servicios comunes oriente. Actualmente estas herramientas no permiten a los analistas realizar la respectiva documentación de las fallas reportadas, lo que impide generar acciones predictivas y correctivas de manera remota y asegurar la localización y notificación oportuna de eventos al personal de soporte de mantenimiento. Por ello la Superintendencia de Mantenimiento de la Plataforma se vio en la necesidad de solicitar a la gerencia de AIT el desarrollo de un software para la automatización de las alarmas generadas por una herramienta de monitoreo, para obtener beneficios de seguridad de la información, disminución de esfuerzo y tiempo, y rapidez en cuanto a la toma de decisiones. El diseño y desarrollo de la aplicación fue realizado según las fases del Proceso Unificado Racional (RUP). Entre las técnicas y tecnologías usadas para el diseño se encuentran el Lenguaje Unificado de Modelado (UML) y su extensión WEBML. Para la implementación se utilizaron las herramientas de PHP y PostgreSQL cumpliendo con las normas de la empresa, en cuanto al desarrollo de aplicaciones con software libre de acuerdo al decreto presidencial 3.390.

ÍNDICE DE CONTENIDO

	Pág.
RESOLUCIÓN	iv
DEDICATORIA	v
AGRADECIMIENTOS	vii
RESUMEN.....	ix
ÍNDICE DE FIGURAS.....	xvii
ÍNDICE DE TABLAS	xxii
CAPÍTULO I. INTRODUCCIÓN	25
1.1. Planteamiento del Problema.....	25
1.2. Objetivos	29
1.2.1. Objetivo General	29
1.2.2. Objetivos Específicos.....	29
CAPÍTULO II. MARCO TEÓRICO	31
2.1. Antecedentes de la Investigación.....	31
2.2. Fundamentos Teóricos	33
2.2.1. Sistema de gestión de casos	33
2.2.2. Nagios	33
2.2.2.1. Características de Nagios	33
2.2.2.2. Requerimientos	34
2.2.2.3. Que Monitorea Nagios	34
2.2.3. Falla.....	35
2.2.5. Mantenimiento Preventivo	35
2.2.5.1. Mantenimiento Preventivo Por Condición (MPC).....	35
2.2.5.2. Mantenimiento Preventivo Por Planificación (MPP).....	35
2.2.6. Mantenimiento Correctivo	36
2.2.7. Alarma.....	36
2.2.8. Evento	36

2.2.9. Software libre	37
2.2.10. Software Libre en Venezuela	39
2.2.14. Base de Datos	41
2.2.14.1. Elementos de almacenamiento de una Base de Datos	41
2.2.14.2. Tipos de Base de Datos	41
2.2.14.4. Modelo Entidad-Relación	42
2.2.14.5. Sistema Manejador de Base de Datos (DBMS)	43
2.2.15. Lenguaje de Programación PHP	45
2.2.17. Lenguaje CSS	46
2.2.18. Lenguaje JavaScript	46
2.2.19. Ajax	47
2.2.20. Servidor Web Apache	48
2.2.21. PostgreSQL	48
2.3 Fundamentos Metodologicos	48
2.3.1. Ingeniería de Software	48
2.3.2. Proceso Unificado Racional (RUP)	49
2.3.2.1. Dirigido por Casos de Uso	50
2.3.2.2. Centrado en la Arquitectura	51
2.3.2.3. Iterativo e Incremental	52
2.3.3. Ciclo de Vida RUP	52
2.3.3.1. Fase de Inicio	54
2.3.3.2. Fase de Elaboración	54
2.3.3.3. Fase de Construcción	55
2.3.3.4. Fase de Transición	55
2.3.4. Flujos de trabajo del Proceso Unificado	56
2.3.4.1. Requisitos	56
2.3.4.2. Análisis	57
2.3.4.3. Diseño	57
2.3.4.4. Implementación	57

2.3.4.5. Pruebas	58
2.3.5. Lenguaje de Modelado Unificado (UML)	58
2.3.6. Tipos de Modelo	59
2.3.7. WebML	61
2.3.7.1. Modelo de Datos	64
2.3.7.2. Modelo de Hipertexto	64
2.3.7.3. Elementos del modelo Hipertexto WebML	65
2.3.7.4. Modelo de Presentación	66
2.3.8. Programación Orientada a Objetos (POO).....	66
2.3.8.1. Ventajas de la POO	72
CAPÍTULO III. FASE DE INÍCIO	74
3.1. Introducción	74
3.2. Flujo de Trabajo de Requisitos	74
3.2.1. Estudio del contexto del sistema.....	75
3.2.1.1 Modelo de Dominio	76
3.2.2. Requerimientos esenciales del sistema	78
3.2.3. Requisitos Funcionales del Sistema.....	79
3.2.4. Requisitos No Funcionales del Sistema	79
3.2.5. Identificación de los Riesgos	80
3.2.6. Modelos de Casos de Uso	81
3.2.6.1. Actores del Sistema.....	81
3.2.6.2. Casos de Uso del Sistema	85
3.2.7. Descripción de los Casos de Uso	85
3.2.7.1. Caso de Uso Administrar Usuarios	85
3.2.7.2. Caso de Uso Gestionar Alarmas	88
3.2.7.3. Caso de Uso Gestionar Eventos	90
3.2.7.4. Caso de Uso Gestionar Servicios	92
3.2.7.5. Caso de Uso Generar Reportes	93
3.3. Flujo de Trabajo de Análisis	94

3.3.1. Especificación de los grupos de usuarios formalmente descritos	95
3.3.2. Diagrama de Clases de Análisis	95
3.3.2.1. Descripción del Diagrama de Clase de Análisis para el caso de uso Administrar Usuarios	102
3.3.2.2. Descripción del Diagrama de Clase de Análisis para el caso de uso Gestionar Alarmas.....	102
3.3.2.3. Descripción del Diagrama de Clase de Análisis para el caso de uso Gestionar Eventos	107
3.3.2.4. Descripción del Diagrama de Clase de Análisis para el caso de uso Gestionar Servicios	110
3.3.2.5. Descripción del Diagrama de Clase de Análisis para el caso de uso Generar Reportes.....	111
3.4. Flujo de Trabajo Diseño.....	112
3.4.1. Paquete de Analisis del sistema	112
3.4.2. Prototipo Interfaz candidata	113
3.5. Evaluación de la Fase de Inicio.....	113
CAPÍTULO IV. FASE DE ELABORACIÓN	115
4.1. Introducción	115
4.2. Flujo de Trabajo de Análisis	116
4.2.1. Diagramas de Colaboración	116
4.2.1.1. Descripción del Diagrama de Colaboración para el caso de uso Administrar Usuarios	117
4.2.1.2. Descripción del Diagrama de Colaboración para el caso de uso Gestionar Alarmas: Consultar Alarmas.	118
4.2.1.3. Descripción del Diagrama de Colaboración para el caso de uso Gestionar Alarmas: Consultar Alarmas Cerradas	121
4.2.1.4. Descripción del Diagrama de Colaboración para el caso de uso Gestionar Alarmas: Gestionar Configuración.....	124

4.2.1.5. Descripción del Diagrama de Colaboración para el caso de uso Gestionar Eventos: Ingresar Eventos	128
4.2.1.6. Descripción del Diagrama de Colaboración para el caso de uso Gestionar Eventos: Consultar Eventos.....	128
4.2.1.7. Descripción del Diagrama de Colaboración para el caso de uso Gestionar Eventos: Consultar Eventos Cerrados	133
4.2.1.8. Descripción del Diagrama de Colaboración para el caso de uso Gestionar Servicios	135
4.2.1.9. Descripción del Diagrama de Colaboración para el caso de uso Generar Reportes.....	137
4.3. Flujo de Trabajo de Diseño	138
4.3.1. Diagramas de Clase de Diseño.....	140
4.3.1.1. Diagrama de Clase de Diseño Administrar Usuarios.....	141
4.3.1.2. Diagrama de Clase de Diseño Gestionar Alarmas: Consultar Alarmas	142
4.3.1.3. Diagrama de Clase de Diseño Gestionar Alarmas: Consultar Alarmas Cerradas	143
4.3.1.4. Diagrama de Clase de Diseño Gestionar Alarmas: Gestionar Configuración.....	144
4.3.1.5. Diagrama de Clase de Diseño Gestionar Eventos: Ingresar Eventos	145
4.3.1.6. Diagrama de Clase de Diseño Gestionar Eventos: Consultar Eventos	146
4.3.1.7. Diagrama de Clase de Diseño Gestionar Eventos: Consultar Eventos Cerrados	147
4.3.1.8. Diagrama de Clase de Diseño Gestionar Servicios.....	148
4.3.1.9. Diagrama de Clase de Diseño Generar Reportes	149
4.3.2. Diseño de la arquitectura.....	150
4.3.3. Diseño de la base de datos	150

4.3.3.1. Modelo De Datos	152
4.3.3.2. Estructura de la Base de Datos.....	154
4.3.4. Modelo De Hipertexto y Modelo De Presentación.....	161
4.3.4.1. Vista Administrador del Sistema	162
4.3.4.2. Vista Analista de Monitoreo	184
4.3.4.3. Vista Analista de Control de Gestión.....	186
4.3.4.4. Vista Superintendente	187
4.3.5. Modelo de navegación	188
4.4. Flujo de Trabajo de Implementación	189
4.4.1. Identificación de Componentes de la Arquitectura.....	189
4.4.2. Implementación de la Arquitectura.....	190
4.4.3. Conclusión de la Fase de Elaboración	190
CAPÍTULO V. FASE DE CONSTRUCCIÓN.....	193
5.1. Introducción	193
5.2. Flujo de Trabajo de Implementación	193
5.2.1. Área Alarmas	194
5.2.1.1. Código Fuente Consultar Alarmas	195
5.2.1.2. Código Fuente Gestionar Configuración	208
5.2.2. Área Eventos	214
5.2.2.1. Código Fuente Consultar Eventos.....	215
5.2.2.2. Código Fuente Consultar Eventos Cerrados por Código	223
5.2.3. Área Reportes.....	229
5.2.3.1. Código Fuente Reporte Estándar	229
5.2.3.2. Código Fuente Reporte por Impacto	236
5.3. Flujo de Trabajo de Pruebas.....	244
5.3.1. Modelo de Pruebas.....	245
5.3.2. Casos de Pruebas.....	245
5.3.2.2. Caso de prueba del caso de uso Administrar Usuarios	246
5.3.2.3. Caso de prueba del caso de uso Gestionar Servicios	246

5.3.2.4. Caso de prueba del caso de uso Gestionar Eventos	248
5.3.2.5. Caso de prueba del caso de uso Gestionar alarmas.....	249
5.3.3. Pruebas de Integración	250
CAPÍTULO VI. FASE DE TRANSICIÓN	252
6.1. Introducción	252
6.2. Flujo de Trabajo de Implementación	272
6.2.1. Preparación de la versión beta.....	272
6.2.2. Instalación de la versión beta	272
6.3. Flujo de Trabajo de Pruebas.....	272
6.3.1. Pruebas a partir de la versión beta.....	272
6.3.1.1. Reacción a los resultados de las pruebas.....	273
CONCLUSIONES	274
RECOMENDACIONES	276
BIBLIOGRAFÍA	277
ANEXOS	280

ÍNDICE DE FIGURAS

	Pág.
Figura 2.1. Ciclo de vida RUP.	53
Figura 2.2. Flujos de trabajo: requisitos, análisis, diseño, implementación y prueba / Fases: inicio, elaboración, construcción y transición.....	56
Figura 3.1. Modelo de Dominio.....	78
Figura 3.2. Diagrama de Caso de Uso general del sistema.....	86
Figura 3.3. Especificación de Grupos formalmente descritos.....	96
Figura 3.4. Diagrama de Clase de Análisis para el caso de uso Administrar Usuarios.	103
Figura 3.5. Diagrama de Clase de Análisis para el caso de uso Gestionar Alarmas: Consultar Alarmas.....	105
Figura 3.6. Diagrama de Clase de Análisis para el caso de uso Gestionar Alarmas: Consultar Alarmas Cerradas.	106
Figura 3.7. Diagrama de Clase de Análisis para el caso de uso Gestionar Alarmas: Gestionar Configuración.	107
Figura 3.8. Diagrama de Clase de Análisis para el caso de uso Gestionar Eventos: Ingresar Eventos.....	108
Figura 3.9. Diagrama de Clase de Análisis para el caso de uso Gestionar Eventos: Consultar Eventos.	109
Figura 3.10. Diagrama de Clase de Análisis para el caso de uso Gestionar Eventos: Consultar Eventos Cerrados.....	110
Figura 3.11. Diagrama de Clase de Análisis para el caso de uso Gestionar Servicios.	111
Figura 3.12. Diagrama de Clase de Análisis para el caso de uso Generar Reportes.	112
Figura 3.13. Diagrama de Paquete de Analisis del Sistema.....	114
Figura 3.14. Prototipo de Interfaz candidata.....	114
Figura 4.1. Diagrama de Colaboración para el caso de uso Administrar Usuarios. .	119

Figura 4.2. Diagrama de Colaboración para el caso de uso Gestionar Alarmas: Consultar Alarmas.....	122
Figura 4.3. Diagrama de Colaboración para el caso de uso Gestionar Alarmas: Consultar Alarmas Cerradas.	125
Figura 4.4. Diagrama de Colaboración para el caso de uso Gestionar Alarmas: Gestionar Configuración.	127
Figura 4.5. Diagrama de Colaboración para el caso de uso Gestionar Eventos: Ingresar Eventos.....	129
Figura 4.6. Diagrama de Colaboración para el caso de uso Gestionar Eventos: Consultar Eventos.	131
Figura 4.7. Diagrama de Colaboración para el caso de uso Gestionar Eventos: Consultar Eventos Cerrados.....	134
Figura 4.8. Diagrama de Colaboración para el caso de uso Gestionar Servicios.....	137
Figura 4.9. Diagrama de Colaboración para el caso de uso Generar Reportes.....	139
Figura 4.10. Diagrama de Clase de Diseño Administrar Usuarios	141
Figura 4.11. Diagrama de Clase de Diseño Gestionar Alarmas: Consultar Alarmas	142
Figura 4.12. Diagrama de Clase de Diseño Gestionar Alarmas: Consultar Alarmas Cerradas	143
Figura 4.13. Diagrama de Clase de Diseño Gestionar Alarmas: Gestionar Configuración.....	144
Figura 4.14. Diagrama de Clase de Diseño Gestionar Eventos: Ingresar Eventos ...	145
Figura 4.15. Diagrama de Clase de Diseño Gestionar Eventos: Consultar Eventos.	146
Figura 4.16. Diagrama de Clase de Diseño Gestionar Eventos: Consultar Eventos Cerrados	147
Figura 4.17. Diagrama de Clase de Diseño Gestionar Servicios	148
Figura 4.18. Diagrama de Clase de Diseño Generar Reportes.....	149
Figura 4.19. Vista lógica de la arquitectura.	151
Figura 4.20. Relación de tablas de la base de datos.	1513

Figura 4.21. Estructura de la Tabla Equipo.	154
Figura 4.22. Estructura de la Tabla Elemento.....	154
Figura 4.23. Estructura de la Tabla Servicio Elemento.	155
Figura 4.24. Estructura de la Tabla Servicio.....	155
Figura 4.25. Estructura de la Tabla Criticidad Servicio.....	155
Figura 4.26. Estructura de la Tabla Grupo Solucionador.....	156
Figura 4.27. Estructura de la Tabla Localidad.	156
Figura 4.28. Estructura de la Tabla Tipo Equipo.	156
Figura 4.29. Estructura de la Tabla Tipo Elemento.	157
Figura 4.30. Estructura de la Tabla Evento.....	157
Figura 4.31. Estructura de la Tabla Evento Estado Evento.....	157
Figura 4.32. Estructura de la Tabla Estado Evento.....	158
Figura 4.33. Estructura de la Tabla Tipo Estado Evento.	158
Figura 4.34. Estructura de la Tabla Tipo Evento.	158
Figura 4.35. Estructura de la Tabla Mantenimiento.....	159
Figura 4.36. Estructura de la Tabla Impacto Evento.....	159
Figura 4.37. Estructura de la Tabla Documentación.....	159
Figura 4.38. Estructura de la Tabla Tipo Documentación.	160
Figura 4.39. Estructura de la Tabla Usuario.	160
Figura 4.40. Estructura de la Tabla Nivel Usuario.....	160
Figura 4.41. Estructura de la Tabla Estatus Usuario.....	161
Figura 4.42. Estructura de la Tabla Tipo Validación.	161
Figura 4.43 Vista de Administrador del Sistema.	163
Figura 4.44. Página Ingresar Usuario.....	164
Figura 4.45. Página Consultar Usuario.	164
Figura 4.46. Página Modificar Usuario.....	165
Figura 4.47. Página Eliminar Usuario.....	165
Figura 4.48. Página Listar Usuarios.....	166
Figura 4.49. Página Consultar Alarmas.	167

Figura 4.50. Página Alarmas Pendientes.....	168
Figura 4.51. Página Búsqueda Avanzada.....	169
Figura 4.52. Página Búsqueda por Código.	170
Figura 4.53. Página Ingresar.	171
Figura 4.54. Página Editar.....	172
Figura 4.55. Página Eliminar.	173
Figura 4.56. Página Ingresar Eventos.	174
Figura 4.57. Página Consultar Eventos.	175
Figura 4.58. Página Eventos Pendientes.	176
Figura 4.59. Página Búsqueda Avanzada.....	177
Figura 4.60. Página Búsqueda por Código.	178
Figura 4.61. Página Ingresar Servicio.....	179
Figura 4.62. Página Consultar Servicio.	179
Figura 4.63. Página Modificar Servicio.....	179
Figura 4.64. Página Eliminar Servicio.....	180
Figura 4.65. Página Reporte Estándar.....	181
Figura 4.66. Página Reporte por Impacto.	181
Figura 4.67. Página Reporte por Mantenimiento.....	182
Figura 4.68. Página Reporte por Equipo.....	183
Figura 4.69. Página Reporte de Búsqueda Rápida.....	183
Figura 4.70. Vista Analista de Monitoreo.....	185
Figura 4.71. Vista Analista de Control de Gestión.	187
Figura 4.72. Vista Analista Superintendente.....	189
Figura 4.73. Modelo de Navegación.....	18991
Figura 4.74. Diagrama de Despliegue del Sistema Integrado de Notificación y Comunicación de Alarmas (SIDENCA).....	192
Figura 5.1. Pantalla Consultar Alarmas: Área Alarmas.....	195
Figura 5.2. Pantalla Gestionar Configuración: Área Alarmas.	208

Figura 5.3. Pantalla Consultar Eventos: Área Eventos.	215
Figura 5.4. Pantalla Consultar Eventos Cerrados por Código: Área Eventos.....	223
Figura 5.7 Pantalla Reporte Estándar: Área Reportes.....	229
Figura 5.8. Pantalla Reporte por Impacto: Área Reportes.	235

ÍNDICE DE TABLAS

	Pág.
Tabla 2.1. Elementos del Modelo de Hipertexto WebML.....	67
Tabla 3.1. Requisitos adicionales del sistema.....	80
Tabla 3.2. Riesgos del Sistema.	82
Tabla 3.3. Identificación de los actores del sistema.....	84
Tabla 3.4. Descripción de los casos de uso del sistema con sus actores.....	87
Tabla 3.5. Identificación de las Clases Interfaz.	97
Tabla 3.6. Identificación de las Clases de Control.....	100
Tabla 3.7. Identificación de las Clases Entidad.	101
Tabla 4.1. Leyenda para el Diagrama de Colaboración para el caso de uso Administrar Usuarios.	120
Tabla 4.2. Leyenda para el Diagrama de Colaboración para el caso de uso Gestionar Alarmas: Consultar Alarmas.	123
Tabla 4.3. Leyenda para el Diagrama de Colaboración para el caso de uso Gestionar Alarmas: Consultar Alarmas Cerradas.....	126
Tabla 4.4. Leyenda para el Diagrama de Colaboración para el caso de uso Gestionar Alarmas: Gestionar Configuración.	127
Tabla 4.5. Leyenda para el Diagrama de Colaboración para el caso de uso Gestionar Eventos: Ingresar Eventos.....	130
Tabla 4.6. Leyenda para el Diagrama de Colaboración para el caso de uso Gestionar Eventos: Consultar Eventos.	132
Tabla 4.7. Leyenda para el Diagrama de Colaboración para el caso de uso Gestionar Eventos: Consultar Eventos Cerrados.....	135
Tabla 4.8. Leyenda para el Diagrama de Colaboración para el caso de uso Gestionar Servicios.....	138

Tabla 4.9. Leyenda para el Diagrama de Colaboración para el caso de uso Generar Reportes.	140
Tabla 4.10. Descripción de la vista administrador del sistema.	162
Tabla 4.11. Descripción de la vista analista de monitoreo.....	184
Tabla 4.12. Descripción de la vista analista de control de gestión.	186
Tabla 4.13. Descripción de la vista superintendente.....	188
Tabla 5.1. Clase de Equivalencia para Operación “Ingresar Usuario”.	247
Tabla 5.2. Clase de Equivalencia para Operación “Ingresar Servicio”.....	248
Tabla 5.3. Clase de Equivalencia para Operación “Ingresar Servicio”.....	249
Tabla 5.4. Clase de Equivalencia para Operación “Buscar Alarma Cerrada por Código”.	250

CAPÍTULO I

INTRODUCCIÓN

1.1. Planteamiento del Problema

La integración de aplicaciones es actualmente uno de los grandes retos de la Ingeniería del Software. Hoy en día no es difícil encontrar empresas que estén ejecutando una cantidad muy grande y variada de aplicaciones en un entorno distribuido para llevar a cabo su negocio. Tal heterogeneidad hace, muchas veces, que unos procesos de negocio de la empresa tengan que utilizar dos o más aplicaciones. En las empresas también es muy frecuente la necesidad de añadir funcionalidades nuevas a las aplicaciones ya existentes, lo que en muchos casos puede resultar prohibitivo.

La integración es una tarea cada vez más frecuente en las empresas, y que además de tener un alto coste suele consumir muchos recursos. Para tales empresas la necesidad de integración es, en gran parte, consecuencia de la evolución natural de su negocio.

Petróleos de Venezuela, Sociedad Anónima (PDVSA) es una empresa estatal venezolana, que se dedica a la explotación, producción, refinación, mercadeo y transporte del petróleo venezolano. Dicha corporación está dividida de la siguiente manera:

- Negocios que se divide en: Exploración y Producción, Refinación, Comercio y Suministro.

- Filiales que está dividido en: Bariven, Corporación Venezolana de Petróleo (CVP), Deltaven, Intevep, Palmaven, PDV Marina, PDVSA América y PDVSA Gas.
- Funciones que a su vez está dividida en: Gerencia Corporativa de Ambiente e Higiene Ocupacional (AHO), Automatización Informática y Telecomunicaciones (AIT), Apoyo Presidencial, Auditoría Fiscal, Centro de Arte la Estancia, Club Recreacional, Consultoría Jurídica, Finanzas, Logística, Prevención y Control de Perdidas (PCP), Planificación, Recursos Humanos y Seguridad Industrial.

La Gerencia de AIT es una Organización que rige, provee y mantiene los servicios y soluciones integrales de tecnologías de automatización, información y comunicaciones de la Corporación; contribuyendo a mantener su continuidad operativa y a ejecutar sus planes; renuevan y actúan como agentes de transformación en PDVSA y en la sociedad venezolana con corresponsabilidad social, económica y ambiental; potenciando un ecosistema tecnológico que impulsa los poderes creadores del pueblo, el conocimiento libre, el desarrollo endógeno sustentable y la economía social productiva para lograr la soberanía tecnológica.

La Gerencia de AIT Servicios comunes Oriente es la responsable por los servicios comunes en la región Oriente. Reporta administrativamente a la Dirección de AIT. Provee una visión integrada de la Continuidad Operativa de AIT en cada región, provee servicios a todos los negocios que conviven en la región a través de los Centros de Atención, centros de Cómputo, Comunicaciones, atención especializada en sitio, dicta lineamientos de operación al personal AIT de las refinerías y distritos para la ejecución del soporte en sitio de servicios comunes. La Gerencia de AIT Servicios Comunes Oriente se divide en Superintendencias dentro de las cuales se encuentra la Superintendencia de Mantenimiento de la Plataforma (MAP).

Esta se encarga de la Planificación, Programación y Ejecución de las actividades de mantenimiento, dichas actividades generan a su vez etapas, herramientas y procesos, en los que esta soportada la Filosofía del Proceso de Mantenimiento de la Plataforma de AIT.

El objetivo fundamental de MAP es Preservar la función de la plataforma AIT mediante el Mantenimiento detectivo, predictivo, preventivo y correctivo oportuno conforme a los acuerdos de calidad establecidos, a fin de dar continuidad operativa al negocio, minimizando las interrupciones e incrementando la disponibilidad de la plataforma, para evitar pérdida de producción directa o indirecta que pueda ocasionar incrementos en el costo de los productos generados y comercializados por PDVSA.

La Superintendencia MAP está dividida en disciplinas, dentro de las cuales se encuentra el Centro Integral de Monitoreo Oriente (CIMOR).

Este se encarga de garantizar la calidad y disponibilidad de la plataforma de AIT de PDVSA mediante el análisis y comprobación de las características funcionales de los sistemas que permitan generar acciones predictivas y correctivas de manera remota y asegurar la localización y notificación oportuna de eventos al personal de soporte y mantenimiento. Dentro del monitoreo de la plataforma se ejecutan las actividades que por su naturaleza requieren ver el estado, evidenciando algún proceso de deterioro o que dependen de detectar a tiempo un incidente que involucre pérdida total de función, posteriormente las acciones correctivas pasan a “Ejecutar mantenimiento en sitio” o “Ejecutar mantenimiento remotamente”.

Actualmente el departamento no cuenta con un procedimiento eficiente que les permita generar un reporte de cada una de las alarmas generadas por una herramienta de monitoreo de equipos llamada Nagios, la cual se encarga de la monitorización de red y sistemas, debido a que este proceso no arroja una

documentación de cada caso, la información es muy escasa y no se puede definir con certeza.

A fin de solventar esta problemática, se propuso en la Superintendencia MAP de AIT - Servicios Comunes Oriente el Desarrollo de una Aplicación Web bajo Software Libre para Automatizar la Gestión de las alarmas generada por Nagios, que permitirá obtener una documentación real y detallada, y estar al tanto de todas las fallas arrojadas por los equipos.

Para cumplir con los requisitos propuestos, se diseñó un módulo que permitiera al usuario poder ingresar los diferentes servicios que ofrece la Gerencia de AIT, y seleccionar los equipos que lo componen, esto para visualizar el servicio que ha sido afectado por la falla.

Para la realización de la aplicación se seleccionó toda la información referente, a la herramienta de monitoreo Nagios, para así establecer la comunicación entre las dos Aplicaciones. Es de suma importancia que la Aplicación cuente con la función de que una vez cargada la información se genere una plantilla que indicará el equipo con su respectiva falla. Las personas encargadas de llenar los casos de fallas en la aplicación deberán ser los analistas de monitoreo, ya que su función es verificar si dicha alarma es real o no.

Se requiere que la aplicación, con la intervención del analista de monitoreo, registre todos los datos referentes a las alarmas generadas, y de corresponder a una falla, permita generar los casos de manera directa y automatizada, siendo almacenada esta información en la base de datos, con el propósito de que esta pueda ser revisada por los analistas solucionadores y así ejecutar las acciones y finalmente documentar y cerrar el caso.

El principal propósito de este proyecto será el desarrollo de una aplicación que permita dar las siguientes soluciones:

- Proporcionar una información detallada a los analistas debido a que éstos podrán acceder de una manera más rápida y eficiente a la consulta de la información.
- Proporcionar una interfaz que le permita al personal del departamento agregar, consultar, eliminar y generar datos.
- Garantizar una información real y sencilla para que cada analista pueda realizar el mantenimiento que se requiere a cada equipo según el tipo de falla que este pueda tener.
- Fácil generación de indicadores de la plataforma.
- Rapidez en cuanto a la toma de decisiones.
- Actualización constante de la base de datos.

Para el desarrollo de la aplicación se utilizó la metodología del Proceso Unificado Racional (RUP), ya que junto con el lenguaje unificado de modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. El RUP divide el ciclo de desarrollo en fases, de las cuales serán contempladas para el avance de la aplicación las siguientes: fase de inicio, fase de elaboración, fase de construcción y fase de transición.

La importancia de este proyecto radica, en que el departamento tenga una mejor distribución de la información técnica, a través de una moderna herramienta, como lo es el desarrollo de una Aplicación Web para automatizar la información.

1.2. Objetivos

1.2.1. Objetivo General

Desarrollar una Aplicación Web bajo Software Libre para la Automatización de la Gestión de las alarmas generadas por una herramienta de monitoreo de equipos, para la Superintendencia de Mantenimiento de la Plataforma de la Gerencia de Automatización informática y Telecomunicaciones de PDVSA Puerto La Cruz.

1.2.2. Objetivos Específicos

- Describir las herramientas de monitoreo y Gestión de Casos utilizadas por la Superintendencia de Mantenimiento de la Plataforma (MAP).
- Diseñar la estructura y arquitectura que debe tener la aplicación.
- Diseñar el Modelo de la Base de Datos relacional mediante la cual trabajará la aplicación y los módulos por los que estará compuesta.
- Codificar la aplicación en función de los estándares requeridos por PDVSA.
- Realizar pruebas, necesarias para la verificación del cumplimiento de los requerimientos de la aplicación.

CAPÍTULO II

MARCO TEÓRICO

2.1. Antecedentes de la Investigación

Para el desarrollo de la aplicación se tomo como base la revisión bibliográfica de diferentes fuentes de información esenciales para el desarrollo de todo proyecto las cuales se utilizaron como apoyo, en cuanto a las metodologías y técnicas manejadas así como en los resultados arrojados por dichos estudios.

A continuación se describen algunos de los trabajos que han servido de precedente y sustento para la realización de la aplicación:

- Tenias, J. (2007) presentó un Trabajo de Grado titulado **“Desarrollo de un software Basado en Aplicaciones Web para el Monitoreo de dispositivos de la Plataforma de Telecomunicaciones de PDVSA-GAS”**. El cual fue presentado en la Universidad de Oriente Núcleo Anzoátegui. Este software permite tener un seguimiento controlado de todos los dispositivos que se encuentran en la Plataforma de Telecomunicaciones PDVSA-GAS. Se utilizó la metodología del Proceso Unificado y UML para el modelado de la estructura del software.

- Medina, J. (2007) presentó un Trabajo de Grado titulado **“Desarrollo de un Sistema basado en Aplicaciones Web para la Automatización del Control de pedidos asociados al Proceso de Ventas de una Empresa Cafetalera”**. El cual fue presentado en la Universidad de Oriente Núcleo Anzoátegui. Este sistema permite tener un control automatizado de todas las ventas realizadas por la empresa. Se utilizó la metodología del Proceso Unificado y UML para el modelado de la estructura del software.

- Núñez, L. (2007) presentó un Trabajo de Grado titulado **“Desarrollo de una Aplicación Web para la visualización en tiempo real de los parámetros operacionales de producción de la empresa PDVSA”**. El cual fue presentado en la Universidad de Oriente Núcleo Anzoátegui. Este software permite visualizar de forma automatizada los parámetros operacionales de producción de la empresa, basado en los principios del Lenguaje Modelado Unificado (UML).
- Ordaz, A. (2007) presentó un Trabajo de Grado titulado **“Desarrollo de un Software para la Automatización y seguimiento de las actividades que se realizan en el almacén de un taller de Mantenimiento Aeronáutico”**. El cual fue presentado en la Universidad de Oriente Núcleo Anzoátegui. Esta herramienta permite gestionar de forma automatizada las actividades realizadas dentro del almacén. En este proyecto el software diseñado utilizó como herramienta el Lenguaje Unificado de Modelado (UML).
- Salazar, P. (2007) presentó un Trabajo de Grado titulado **“Desarrollo de un Software para la Automatización de reportes y consultas de archivos históricos del tráfico de conexiones de red realizadas por la Superintendencia de Seguridad Lógica de una empresa petrolera utilizando tecnología Web”**. El cual fue presentado en la Universidad de Oriente Núcleo Anzoátegui. Este software permite llevar un registro automatizado de todos los archivos históricos del tráfico de conexiones de red de una empresa petrolera. Para el desarrollo de este sistema se utilizó la metodología de Proceso Unificado de Desarrollo de Software cuya arquitectura fue construida bajo el patrón cliente-servidor.
- Abad, J. (2008) presentó un Trabajo de Grado titulado **“Desarrollo de un Sistema de Monitoreo que permita la detección de fallas del sistema de información de tiempo real Petrocedeño”**. El cual fue presentado en la Universidad de Oriente Núcleo Anzoátegui. Este sistema permite detectar las

fallas de los equipos de manera controlada, para facilitar las tareas de monitoreo, basado en los principios del Lenguaje de Modelado Unificado (UML).

2.2. Fundamentos Teóricos

2.2.1. Sistema de gestión de casos

Un sistema de gestión de casos es un modelo organizativo cuyos objetivos son registrar y documentar, cada una de las solicitudes en forma de requerimientos, incidentes y/o problemas, que se generan en las determinadas gerencias de servicios que conforman una empresa. (Garcia, G Villarroel, J y Pernía, M 2008).

2.2.2. Nagios (Barrientos, L y Beites, J 2006)

Es un sistema de monitorización de red, equipos y servicios, el cual informa proactivamente los problemas cuando suceden en tiempo real. Su funcionamiento consiste en una arquitectura cliente-servidor mediante ejecución de polling periódico de chequeos de recursos y servicios sobre sistemas. Cuando se detecta un error es capaz de notificar en diferentes modos, informando el estado del servicio que ha provocado el error, incluyendo informes de estado, de logs e históricos Web. Su arquitectura es abierta la cual permite una rápida y fácil adaptación de sus funciones según las necesidades que se tenga.

Es muy flexible y adaptable a cualquier necesidad. Está escrito en C y es Software Libre (Licencia GPL versión 1.3).

2.2.2.1. Características de Nagios

- Monitoreo de Servicios de Red: SMTP, POP3, HTTP, SSH, DNS, entre otros.
- Monitoreo de Recursos: Carga de procesador, espacio libre en filesystems, uso de la memoria, entre otros.

- Capacidad de desarrollar plugins de forma sencilla que permite programar sus propias actividades.
- Capacidad de definir una topología o jerarquía de red que permita distinguir entre servicios caídos o inaccesibles.
- Envío de notificaciones mediante múltiples métodos (vía email, SMS, alertas sonoras).
- Consola web para visualizar el estado de los servicios, generación de estadísticas, historial de alarmas, entre otros.
- Soporte de bases de datos para el almacenamiento de datos externos (utilizándose ficheros de textos).

2.2.2.2. Requerimientos

- Sistema Operativo Debian o Ubuntu GNU/Linux.
- Compilador C, Apache, PHP, Postgresql, SNMP, Nagios.
- Librerías gráficas GD, JPEG y PNG.

2.2.2.3. Que Monitorea Nagios

- **Servidores UNIX/Linux:** Ping, FileSystems, PGSQL, Memoria, CPU, entre otros.
- **Servidores Windows:** Ping, DNS, LDAP, DHCP, Servicios Activos, Discos, Memoria Virtual, CPU, entre otros.
- **Routers:** Ping, Disponibilidad, Interfaces Serial, ATM, Ethernet, CPU, Memoria, entre otros.
- **Switches:** Ping, Disponibilidad, Puertos, Memoria, entre otros.
- **Otros:** Sensores de Temperatura y Humedad y básicamente cualquier otro elemento que sea detectado por protocolo SNMP.

2.2.3. Falla

Es el efecto que se origina cuando un componente, equipo sistema o proceso deja de cumplir con la función que de ellos se espera. (Medina, N Nucette, G Seneco, K Yañes, M y Gomez, H 2007).

2.2.5. Mantenimiento Preventivo (MP) (Medina, N Nucette, G Seneco, K Yañes, M y Gomes, H 2007)

Es el mantenimiento que se ejecuta a intervalos predeterminados y/o de acuerdo con criterios prescritos, utilizando todos los medios para determinar frecuencia de inspecciones, revisiones sustitución de piezas, probabilidad de aparición de fallas, vida útil etc....con el objetivo de reducir, predecir y/o prevenir fallas o detectarlas en su fase incipiente evitando así la degradación o deterioro del ISED (Infraestructura, Sistemas, Equipos, y Dispositivos), y sus consecuencias negativas para el proceso productivo.

2.2.5.1. Mantenimiento Preventivo Por Condición (MPC)

Consiste en determinar en todo instante la condición técnica (mecánica y/o eléctrica) real del ISED (Infraestructura, Sistemas, Equipos y Dispositivos) examinado, mientras éste se encuentre en pleno funcionamiento, para ello se hace uso de inspecciones y de un programa sistemático de mediciones de los parámetros más importantes del equipo. Tiene como objetivo disminuir las paradas por mantenimientos preventivos, y de esta manera minimizar los costos por mantenimiento y por no producción.

2.2.5.2. Mantenimiento Preventivo Por Planificación (MPP)

Cubre los diferentes niveles de Mantenimiento Preventivo a que son sometidos los equipos, de acuerdo a Planes de Mantenimiento elaborados y

calendarizados. Motivo por el cual, esta Clase de Actividad será empleada por el Planificador/Programador, para programar la realización de estas actividades.

2.2.6. Mantenimiento Correctivo (MC) (Medina, N Nucette, G Seneco, K Yañes, M y Gomes, H 2007)

Es el mantenimiento que se ejecuta después de la aparición de una falla para restablecer un ISED (Infraestructura, Sistemas, Equipos y Dispositivos) a una condición en la cual pueda cumplir la función requerida.

Este mantenimiento también es denominado "mantenimiento reactivo", tiene lugar luego que ocurre una falla o avería, es decir, solo actuará cuando se presenta un error en el sistema. En este caso si no se produce ninguna falla, el mantenimiento será nulo, por lo que se tendrá que esperar hasta que se presente el desperfecto para recién tomar medidas de corrección de errores.

2.2.7. Alarma

Una alarma es aquella que se genera de manera automática por una herramienta utilizada para monitorear equipos. Un aviso de alarma es una señal por medio del cual se informa al personal autorizado para que sigan instrucciones específicas de emergencia debido a la presencia real o inminente de una amenaza. (Barrientos, L y Beites, J 2006).

2.2.8. Evento

Se puede decir que un evento es una alarma no generada, ya sea por desconocimiento de la herramienta acerca del equipo afectado o bien a solicitud del personal encargado del buen funcionamiento del equipo afectado. (Barrientos, L y Beites, J 2006).

2.2.9. Software libre (Stallman, R 2004)

En 1998 un grupo se movió para reemplazar el término “Free Software” por “Open Source”. Los cuales esencialmente se refieren a lo mismo.

El código abierto no solo significa acceso al código fuente. Los términos de distribución del código abierto deben cumplir con los siguientes criterios:

- **Libre distribución:** Las licencias no deberán restringir ninguna parte de vender o entregar el software como un componente de una distribución agregada de software.
- **Código Fuente:** El programa debe incluir el código fuente, y debe permitir la distribución en código fuente así como en su forma compilada. Donde algún producto no sea distribuido con su código fuente, debe existir una forma bien publicada de cómo obtener el código fuente por no más de un costo razonable preferencialmente, descargable de Internet libre de cargos. El código fuente debe ser la forma preferida en la que un programador modificaría el programa. No son permitidos los códigos fuente ofuscados. Formas intermedias tales Como la salida de un pre-procesador o traductor no están permitidas.
- **Trabajos Derivados:** La licencia debe permitir modificaciones y trabajos derivados, y debe permitirles ser distribuidos bajo los mismos términos tal como la licencia del software original.
- **Integridad del código fuente del Autor:** La licencia podría restringir el código fuente de ser distribuido en una forma modificada solo si la licencia permite la distribución de “parches” con el código fuente para el propósito de ser modificada el programa en el momento de construir. La licencia debe permitir explícitamente la distribución de software construido desde código fuente modificado. La

licencia podría requerir trabajos derivados para llevar un nombre o número de versión distinto del software original.

- **Sin discriminación contra personas o grupos:** La licencia no debe discriminar algún tipo de persona o grupo de personas.
- **Sin discriminación contra áreas de esfuerzo:** La licencia no debe restringir a nadie de hacer uso del programa en un área específica de esfuerzo. Por ejemplo, no podría restringir el programa de ser usado en un negocio, o de ser usado para investigación genética.
- **Distribución de la Licencia:** Los derechos adjuntados al programa deben aplicar a todos a quienes el programa les sea redistribuido sin la necesidad de ejecutar una licencia adicional por estas partes.
- **La licencia no debe ser específica a un producto:** Los derechos adjuntados al programa no deben depender de la distribución de software particular de la que sea parte. Si el programa es extraído de esa distribución y usado o distribuido dentro de los términos de su licencia, todas las partes a quienes el programa es redistribuido deben tener los mismos derechos que aquellos que son obtenidos en conjunto con la distribución original del software.
- **La licencia no debe restringir otro software:** La licencia no debe poner restricciones a otro software que sea distribuido junto con el software de la licencia. Por ejemplo, la licencia no debe insistir en que todos los otros programas distribuidos en el mismo medio sean software de código abierto.
- **La licencia debe ser tecnológicamente neutral:** Ninguna provisión de la licencia podría ser predicada en ninguna tecnología individual o estilo de interfaz.

2.2.10. Software Libre en Venezuela (Gobierno Bolivariano de Venezuela 2007)

El Gobierno Bolivariano reconoce la importancia del Software Libre desarrollado bajo Estándares Abiertos, para fortalecer la Industria Nacional de Software, con lo cual, en diciembre del 2004, se publica en Gaceta Oficial el Decreto N° 3.390, que establece: “La Administración Pública Nacional empleará prioritariamente Software Libre desarrollado con Estándares Abiertos, en sus sistemas, proyectos y servicios informáticos. A tales fines, todos los órganos y entes de la Administración Pública nacional iniciarán los procesos de migración gradual y progresiva de éstos hacia el Software Libre”.

Así mismo define el Software Libre como: Programa de computación cuya licencia garantiza al usuario acceso al código fuente del programa y lo autoriza a ejecutarlo con cualquier propósito, modificarlo y redistribuir tanto el programa original como sus modificaciones en las mismas condiciones de licenciamiento acordadas al programa original, sin tener que pagar regalías a los desarrolladores previos.

Además considera: Que el Software Libre desarrollado con Estándares Abiertos, permite mayor participación de los usuarios en el mantenimiento de los niveles de seguridad e interoperabilidad. Por otro lado, los programas informáticos libres son aquellos que cumplen con las siguientes cuatro libertades básicas:

- **Libertad Cero:** “Usar el programa con cualquier propósito”. Es decir, el ejercicio de esta libertad implica que se puede utilizar con cualquier fin, ya sea educativo, cultural, comercial, político, social, etc. Esta libertad se deriva de que hay ciertas licencias que restringen el uso del software a un determinado propósito, o que prohíben su uso para determinadas actividades.

- **Libertad Uno:** “Estudiar cómo funciona el programa, y adaptarlo a sus necesidades”. Significa que se puede estudiar su funcionamiento (al tener acceso al código fuente) lo que va a permitir, entre otras cosas: descubrir funciones ocultas, indagar como se realiza determinada tarea, descubrir que otras posibilidades se tiene, que adaptación se puede realizar para determinada necesidad, etc. El adaptar el programa a determinadas necesidades implica que se puede suprimir partes que no sean del interés del usuario, agregar partes que se consideren importantes, copiar una parte del código que realiza una tarea y adicionarla a otro programa, etc.

- **Libertad Dos:** “Distribuir copias”. Esta significa que existe la libertad de redistribuir el programa, ya sea gratis o con algún costo, por e-mail, FTP o en CD, a una persona, a varias o a una persona que vive en otro país, etc.

- **Libertad Tres:** “Mejorar el programa, y liberar las mejoras al público”. Es la libertad de hacer mejor el programa, es decir, que se pueden hacer que los requerimientos del hardware sean menores para funcionar, que tenga mayores prestaciones, que ocupe menos espacio, que tenga menos errores, entre otras modificaciones. El poder liberar las mejoras al público quiere decir que si se realiza una mejora que permita un requerimiento menor de hardware, o que haga que ocupe menos espacio, entonces se puede redistribuir ese programa mejorado de forma libre, o simplemente proponer la mejora en un lugar público (un foro de noticias, una lista de correo, un sitio Web, un FTP, un canal de chat).

En la Administración Pública Venezolana se debe usar los programas informáticos que cumplan con estas cuatro libertades, las cuales se pueden reconocer por el modelo de licenciamiento de los mismos.

2.2.14. Base de Datos (Elmasri y Navathe 1997)

De forma sencilla puede indicar que una base de datos no es más que un conjunto de información relacionada que se encuentra agrupada o estructurada.

2.2.14.1. Elementos de almacenamiento de una Base de Datos

Campo: Es la unidad básica de una base de datos.

Registro: Es el conjunto de información referida a una misma persona u objeto.

2.2.14.2. Tipos de Base de Datos

Las bases de datos se pueden dividir en cuatro tipos básicos:

- Bases de datos de fichero plano (o ficheros por bloques).
- Bases de datos relacionales.
- Bases de datos orientadas a objetos.
- Bases de datos híbridas.

Base de datos de fichero: Las bases de datos de fichero plano consisten en ficheros de texto divididos en filas y columnas. Estas bases de datos son las más primitivas y quizás ni tan siquiera merezcan considerarse como tales. Pueden ser útiles para aplicaciones muy simples, pero no para aplicaciones medianas o complejas, debido a sus grandes limitaciones.

Base de datos relacional: Las bases de datos relacionales son las más populares actualmente. Su nombre proviene de su gran ventaja sobre las bases de datos de fichero plano: la posibilidad de relacionar varias tablas de datos entre sí, compartiendo información y evitando la duplicidad y los problemas que ello conlleva

(espacio de almacenamiento y redundancia). Existen numerosas bases de datos relacionales para distintas plataformas (Access, Paradox, Oracle, Sybase,) y son ampliamente utilizadas. Sin embargo, tienen un punto débil: la mayoría de ellas no admite la incorporación de objetos multimedia tales como sonidos, imágenes o animaciones.

Base de datos orientada a objeto: Las bases de datos orientadas a objetos incorporan el paradigma de la Orientación a Objetos (OO) a las bases de datos. La base de datos está constituida por objetos, que pueden ser de muy diversos tipos, y sobre los cuales se encuentran definidas unas operaciones. Las bases de datos orientadas a objetos pueden manejar información binaria (como objetos multimedia) de una forma eficiente. Su limitación suele residir en su especialización, ya que suelen estar diseñadas para un tipo particular de objetos (por ejemplo, una base de datos para un programa de CAD).

Base de datos híbrida: Las bases de datos híbridas combinan características de las bases de datos relacionales y las bases de datos orientadas a objetos. Manejan datos textuales y datos binarios, a los cuales se extienden las posibilidades de consulta. Es una tecnología reciente y aún existen pocas en el mercado.

2.2.14.4. Modelo Entidad-Relación

Un diagrama o modelo entidad-relación (a veces denominado por su siglas, *E-REntity relationship*) es una herramienta para el modelado de datos de un sistema de información. Estos modelos expresan entidades relevantes para un sistema de información, sus inter-relaciones y propiedades.

El Modelo Entidad-Relación es un concepto de modelado para bases de datos, propuesto por Peter Chen en 1976, mediante el cual se pretende visualizar los objetos

que pertenecen a la Base de Datos como *entidades* (esto es similar al modelo de Programación Orientada a Objetos) las cuales tienen unos atributos y se vinculan mediante *relaciones*.

Es una representación conceptual de la información. Mediante una serie de procedimientos se puede pasar del modelo E-R a otros, como por ejemplo el modelo relacional.

El modelado entidad-relación es una técnica para el modelado de datos utilizando diagramas entidad relación. No es la única técnica pero sí la más utilizada. Brevemente consiste en los siguientes pasos:

- Se parte de una descripción textual del problema o sistema de información a automatizar (los requisitos).
- Se hace una lista de los sustantivos y verbos que aparecen.
- Los sustantivos son posibles entidades o atributos.
- Los verbos son posibles relaciones.
- Analizando las frases se determina la cardinalidad de las relaciones y otros detalles.
- Se elabora el diagrama (o diagramas) entidad-relación.
- Se completa el modelo con listas de atributos y una descripción de otras restricciones que no se pueden reflejar en el diagrama.

2.2.14.5. Sistema Manejador de Base de Datos (DBMS)

El DBMS es un conjunto de programas que se encargan de manejar la creación y todos los accesos a las Bases de Datos. Se compone de un lenguaje de definición de datos (DDL: *Data Definition Language*), de un lenguaje de

manipulación de datos (DML: *Data Manipulation Language*) y de un lenguaje de consulta (SQL: *Structured Query Language*).

El Lenguaje de Definición de Datos (DDL): Es utilizado para describir todas las estructuras de información y los programas que se usan para construir, actualizar e introducir la información que contiene una Base de Datos.

El Lenguaje de Manipulación de Datos (DML): Es utilizado para escribir programas que crean, actualizan y extraen información de las Bases de Datos.

El Lenguaje de Consulta (SQL): Es empleado por el usuario para extraer información de las Bases de Datos. El Lenguaje de consulta permite al usuario hacer requisiciones de datos sin tener que escribir un programa, usando instrucciones como el SELECT, el PROJECT y el JOIN.

La secuencia conceptual de operaciones que ocurren para acceder cierta información que contiene una base de datos es la siguiente:

- El usuario solicita información contenida en la Base de Datos.
- El DBMS intercepta este requerimiento y lo interpreta.
- El DBMS realiza las operaciones necesarias para acceder y/o actualizar la información solicitada.
- El administrador de la Base de Datos (DBA).
- El DBA es la persona encargada de definir y controlar las Bases de Datos corporativas, además proporciona asesoría a los usuarios y ejecutivos que la requieran.

2.2.15. Lenguaje de Programación PHP (Alonso, S 2006)

PHP es un lenguaje de programación que permite, entre otras cosas, la generación dinámica de contenidos en un servidor Web. Entre sus principales características, se pueden destacar su potencia, alto rendimiento y su facilidad de aprendizaje. PHP es una eficaz herramienta de desarrollo para los programadores web, ya que proporciona elementos que permiten generar de manera rápida y sencilla sitios web dinámicos.

Su interpretación y ejecución se da en el servidor, en el cual se encuentra almacenado el script, y el cliente sólo recibe el resultado de la ejecución. Cuando el cliente hace una petición al servidor para que le envíe una página web, generada por un script PHP, el servidor ejecuta el intérprete de PHP, el cual procesa el script solicitado que generará el contenido de manera dinámica, pudiendo modificar el contenido a enviar, y regresa el resultado al servidor, el cual se encarga de regresárselo al cliente. Este proceso se ve reflejado en la Figura 2.2.

Utilizando PHP también es posible generar archivos PDF, Flash, así como imágenes en diferentes formatos, entre otras cosas.

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, pero sin embargo para que sus páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP.

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX (y de ese tipo, como Linux), Windows y Mac OS X, y puede interactuar con los servidores de web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI.

El modelo PHP puede ser visto como una alternativa al sistema de Microsoft que utiliza ASP.NET/C#/VB.NET, a ColdFusion de la compañía Macromedia, a JSP/Java de Sun Microsystems, y al famoso CGI/Perl". Su creación y desarrollo se da en el ámbito de los sistemas libres, bajo la licencia GNU, sin embargo existen también compiladores comerciales.

2.2.17. Lenguaje CSS (Eguíluz, J 2008)

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML Y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

El trabajo del diseñador web siempre está limitado por las posibilidades de los navegadores que utilizan los usuarios para acceder a sus páginas. Por este motivo es imprescindible conocer el soporte de CSS en cada uno de los navegadores más utilizados del mercado.

Internamente los navegadores están divididos en varios componentes, la parte del navegador que se encarga de interpretar el código HTML y CSS para mostrar las páginas se denomina motor. Desde el punto de vista del diseñador CSS, la versión de un motor es mucho más importante que la versión del propio navegador.

2.2.18. Lenguaje JavaScript (Eguíluz, J 2008)

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas Web dinámicas.

JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos.

El lenguaje JavaScript surgió debido a que las aplicaciones Web eran cada vez más complejas y la velocidad de navegación era muy lenta.

Entre las funcionalidades que aporta Javascript al desarrollo de aplicaciones Web se encuentran:

- Javascript da a los diseñadores HTML una herramienta de programación.
- Normalmente, los diseñadores HTML no son programadores, sin embargo, Javascript es un lenguaje de scripts con una sintaxis muy simple.
- Javascript puede reaccionar a eventos.
- Javascript puede colocar texto dinámico dentro de una página HTML.
- Javascript puede leer y escribir elementos HTML.
- Javascript puede ser usado para validar datos.
- Javascript puede ser usado para crear cookies.

2.2.19. Ajax (Eguíluz, J 2008)

Ajax no es un lenguaje de programación, sino una nueva técnica para crear aplicaciones Web más rápidas, eficientes e interactivas.

Con Ajax, un script Javascript puede comunicarse directamente con el servidor, a través del objeto XMLHttpRequest. Con este objeto se puede intercambiar data con el servidor sin recargar la página.

Ajax está basado en estándares de Internet, tales como:

- Javascript.
- XML.
- HTML.
- CSS.

Las aplicaciones Ajax son independientes del navegador Web donde se ejecuten.

2.2.20. Servidor Web Apache (Apache Software Foundation 2009)

El servidor Web Apache es un esfuerzo por desarrollar y mantener un servidor HTTP de código abierto para sistemas operativos modernos. El objetivo del proyecto es proveer un servidor web seguro, eficiente y extensible que proporcione servicios HTTP en sincronización con los estándares HTTP actuales.

El servidor web Apache es un proyecto de la fundación de software Apache (Apache Software Foundation) y ha sido el servidor web más popular en Internet desde Abril de 1996.

2.2.21. PostgreSQL (Sitio Oficial de Postgresql 2009)

PostgreSQL es un poderoso sistema manejador de bases datos relacionales, de código abierto y capaz de ejecutarse en los principales sistemas operativos, incluyendo Linux, Unix (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows.

PostgreSQL provee total soporte a claves foráneas, vistas, uniones, triggers y procedimientos almacenados (en múltiples lenguajes) e incluye la mayoría de los tipos de datos SQL92 y SQL99, tales como: NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTEGER, INTERVAL, entre otros.

2.3 Fundamentos Metodologicos

2.3.1. Ingeniería de Software

El "software es la suma total de los programas de computadora, procedimientos, reglas, la documentación asociada y los datos que pertenecen a un

sistema de cómputo" y "un producto de software es un producto diseñado para un usuario". En este contexto, la Ingeniería de Software es un enfoque sistemático del desarrollo, operación, mantenimiento y retiro del software, por esto se le considera a la Ingeniería de Software "la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas (eficaces en costo o económicas) a los problemas de desarrollo de software", es decir, "permite elaborar consistentemente productos correctos, utilizables y costo-efectivos". (Schach, S 2005).

El proceso de ingeniería de software se define como "un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad". El proceso de desarrollo de software "es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo". (Binder, R 2000).

2.3.2. Proceso Unificado Racional (RUP) (Booch, G 1999)

El Proceso Unificado Racional (Rational Unified Process en inglés), habitualmente resumido como RUP es un proceso de desarrollo de software, es decir un conjunto de actividades necesarias para transformar los requisitos de un usuario", o de un grupo de éstos en un sistema de software. Se basa en componentes, es decir, el sistema "Software en Construcción está formado por componentes software interconectados a través de interfaces bien definidas". Además, el Proceso Unificado se fundamenta en una herramienta esencial: el Lenguaje Unificado de Modelado (Unified Modeling Language, UML), ésta es sumamente versátil y poderosa para expresar gráficamente todos los esquemas de un sistema software.

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

El Proceso Unificado posee tres paradigmas que lo identifican y particularizan de forma única, estos son:

- Dirigido por casos de uso.
- Centrado en la arquitectura.
- Iterativo e incremental.

2.3.2.1. Dirigido por Casos de Uso

Se define caso de uso como “un fragmento de funcionalidad del sistema que proporciona al usuario un resultado”. Así, se tiene que los casos de uso se emplean para especificar los requisitos funcionales de un sistema, además que guían su diseño, implementación y prueba, por tal, guían el proceso de desarrollo.

Dirigido por casos de uso significa que el proceso de desarrollo avanza a través de una serie de flujos de trabajo que se originan precisamente en los casos de de uso. Basándose en ellos, los desarrolladores crean una serie de modelos de diseño e implementación que llevan a cabo los casos de uso. Además, estos modelos se validan para que sean conformes a los casos de uso. Finalmente, dichos casos también sirven para realizar las pruebas sobre los componentes desarrollados.

Los casos de uso no se desarrollan independientemente, se desarrollan a la vez que la arquitectura del sistema. Es decir, los casos de uso guían la arquitectura del sistema y dicha arquitectura influye en la selección de los mismos. En consecuencia, tanto la arquitectura del sistema como los casos de uso maduran según avanza el ciclo de desarrollo.

2.3.2.2. Centrado en la Arquitectura

La arquitectura software plantea diferentes planos del mismo sistema en construcción con características y significación propias, esta idea es análoga a la arquitectura de un edificio, el cual antes de ser construido se contempla desde varios puntos de vista: estructura, conducciones eléctricas, fontanería, entre otras. Cada uno de estos aspectos está representado por un gráfico con su notación correspondiente (planos). De forma similar sucede con el software.

“La arquitectura Software incluye los aspectos estáticos y dinámicos más significativos del sistema”. Ahora la descripción de esta arquitectura se establece mediante un conjunto de vistas del sistema o vistas de los modelos del sistema (vista del modelo de casos de uso, vista del modelo de análisis, vista del modelo de diseño, vista del modelo de despliegue y vista del modelo de implementación). Por tal razón, la arquitectura no es más que una “vista general de las características más resaltantes del sistema”. También, es importante destacar que “la arquitectura debe diseñarse para permitir que el sistema evolucione, no sólo en su desarrollo inicial, sino también a lo largo de las futuras generaciones”.

La arquitectura persigue ciertos objetivos fundamentales como “la comprensibilidad, la capacidad de adaptación al cambio y la reutilización”. Para ello debe existir una interacción entre los casos de uso (la función) y la arquitectura (la forma). Para encontrar y establecer la forma, es decir, la arquitectura que permita que el sistema evolucione tanto en su etapa inicial como futura, es necesario trabajar en “la comprensión de las funciones claves del sistema” o lo que se denomina casos de uso claves. A decir verdad, “tanto la arquitectura como los casos de uso deben evolucionar en paralelo”.

2.3.2.3. Iterativo e Incremental

Todo sistema informático complejo supone un gran esfuerzo que puede durar desde varios meses hasta años. Por lo tanto, lo más práctico es dividir un proyecto en varias partes más pequeñas o mini proyectos conocidos como iteraciones, los cuales, satisfactoriamente ejecutados derivan en incrementos. Las iteraciones se refieren a los pasos en el flujo de trabajo (requisitos, análisis, diseño, implementación y prueba) y los incrementos, al crecimiento del sistema. Además, cada iteración parte de la anterior, incrementando o revisando la funcionalidad implementada.

Los tres paradigmas antes descritos tienen igual rango de importancia. La arquitectura es la estructura sobre la cual cabalgan las iteraciones, paralelamente los casos de uso establecen los objetivos y guían el trabajo en cada iteración. La supresión de alguno de estos conceptos reduce grandemente el valor del Proceso Unificado.

El Proceso Unificado es un modelo complejo con mucha terminología propia, pensado principalmente para el desarrollo de grandes proyectos. Es un proceso que puede adaptarse y extenderse en función de las necesidades de cada empresa.

2.3.3. Ciclo de Vida RUP (Schach, S 2005)

El ciclo de vida RUP es una implementación del Desarrollo en espiral. Fue creado ensamblando los elementos en secuencias semi-ordenadas. El ciclo de vida organiza las tareas en fases e iteraciones.

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. En la Figura 2.1 se muestra como varía el

esfuerzo asociado a las disciplinas según la fase en la que se encuentre el proyecto RUP.

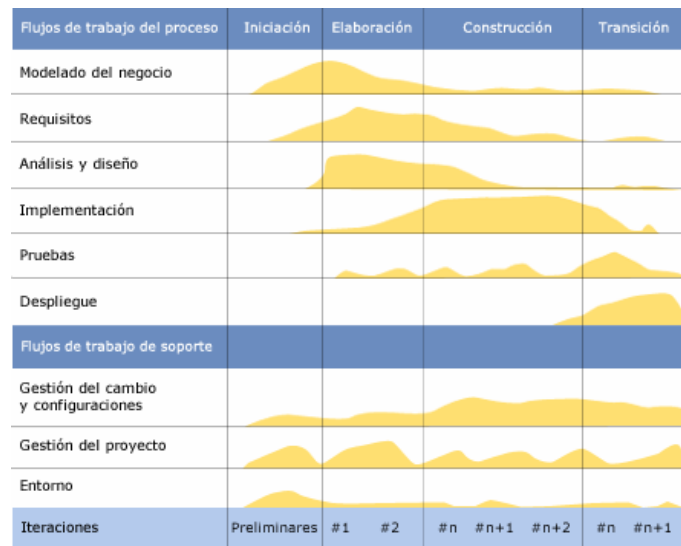


Figura 2.1. Ciclo de vida RUP.

Fuente: Schach, 2005.

El proceso unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo consta de cuatro fases (inicio, elaboración, construcción y transición). Cada fase luego se descompone en iteraciones con sus respectivos incrementos, y concluyen con la disponibilidad de un conjunto de modelos desarrollados hasta conseguir un estado predefinido, llamado hito, que permite la toma de decisiones importantes para así continuar la fase siguiente, permite además controlar el progreso del trabajo según pasa por cada fase. A continuación se describe brevemente cada fase.

2.3.3.1. Fase de Inicio

El objetivo general de esta fase es poner en marcha el proyecto de desarrollo, pero antes se debe desarrollar el análisis del negocio lo cual lleva a delimitar el alcance (ámbito) del sistema, el ámbito es necesario para comprender lo que debe cubrir la arquitectura y así ubicar y enfrentar tempranamente los riesgos dentro de los límites del sistema.

En esta fase se identifican los requisitos funcionales del sistema mediante un modelo de casos de uso que comprende los casos de uso más críticos, además se identifican y priorizan los riesgos más importantes. Adicionalmente se planifica en detalle la fase de elaboración.

2.3.3.2. Fase de Elaboración

Para esta fase se recopilan la mayor parte de los requisitos funcionales, aun pendientes, mediante los casos de uso y se diseña la arquitectura del sistema. Los objetivos fundamentales son:

- Recopilar la mayor parte de los requisitos definiéndolos como casos de uso.
- Establecer una arquitectura sólida para guiar las fases posteriores.
- Continuar la observación y control de los riesgos críticos.
- Completar el plan de proyecto.

Para su cumplimiento, es necesario tomar las decisiones considerando la comprensión global del sistema: ámbito, requisitos funcionales y no funcionales.

2.3.3.3. Fase de Construcción

El objetivo principal de esta fase es culminar y entregar un producto software en su versión operativa inicial conocida como “versión beta”. El producto debería poseer la calidad adecuada y la seguridad de cumplir los requisitos para así emprender su aplicación en la comunidad de usuarios. Para alcanzar todo lo anterior se deben detallar todos los casos de uso y escenarios restantes, modificar si es pertinente la descripción de la arquitectura, dejar cerrados los modelos de análisis, diseño e implementación continuados en los flujos de trabajo de las iteraciones adicionales. Además integrar los subsistemas y someterlos a prueba y hacer lo propio con todo el sistema.

Al finalizar esta fase, el producto contiene todos los casos de uso que la dirección y el cliente han acordado en desarrollar. Y es aquí donde se establece y aclara la expectativa acerca de sí el producto cubre suficientemente las necesidades de los usuarios, como para hacer entrega de la versión inicial.

2.3.3.4. Fase de Transición

Esta fase tiene como objetivos fundamentales:

- Cumplir con todos los requisitos establecidos en fases anteriores, hasta satisfacer a todos los usuarios.
- Gestionar y preparar todos los aspectos de implantación relativos a la operación del sistema en el contexto del usuario.
- La aplicación de pruebas de aceptación, que no son más, que la corrección de los posibles defectos acusados por los usuarios en la versión beta.

La fase de transición conlleva actividades como: la fabricación, formación del cliente, el proporcionar una línea de ayuda y asistencia, y la corrección de los defectos encontrados posterior a la entrega del software.

2.3.4. Flujos de trabajo del Proceso Unificado (Schach, S 2005)

El proceso unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo consta de cuatro fases: inicio, elaboración, construcción y transición. Cada fase se subdivide a su vez en iteraciones se muestra en la Figura 2.2.

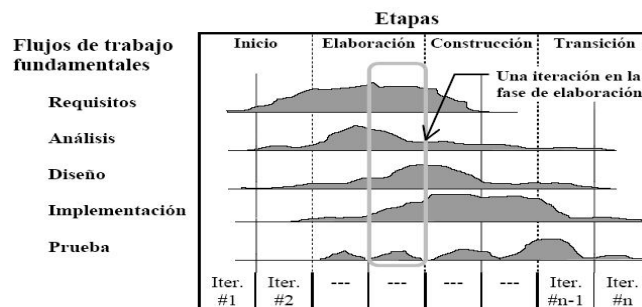


Figura 2.2. Flujos de trabajo: requisitos, análisis, diseño, implementación y prueba / Fases: inicio, elaboración, construcción y transición.

Fuente: Jacobson L (1997).

2.3.4.1. Requisitos

Para la Captura de Requisitos se debe crear un Modelo que represente el sistema a desarrollar, este modelo contiene los requisitos funcionales, es decir, los casos de uso, entonces se habla de Modelo de Casos de Uso. El objetivo fundamental de este flujo de trabajo es identificar, documentar y detallar los casos de uso representativos e identificar igualmente a los actores.

2.3.4.2. Análisis

El Modelo de análisis, tiene dos propósitos fundamentales: refinar los casos de uso con más detalle y establecer la asignación inicial de funcionalidad del sistema a un conjunto de objetos que proporcionan el comportamiento. También se debe señalar que el modelo de análisis hace abstracciones y evita resolver algunos problemas y tratar algunos requisitos, por tanto se posponen al diseño y la implementación.

2.3.4.3. Diseño

Es en este flujo en donde se moldea el sistema y se establece su forma, es decir, su arquitectura, ello con la intención de dar soporte a sus requisitos funcionales y no funcionales. Es importante señalar además, que el diseño se vale del modelo de análisis obtenido en el flujo de igual nombre, el cual proporciona una comprensión detallada de los requisitos, esto le es esencial para los propósitos establecidos por el diseño.

El diseño también se plantea la misión de establecer adecuadas entradas y correctos puntos de partida para el flujo de implementación mediante la captura de requisitos o subsistemas individuales, interfaces y clases.

2.3.4.4. Implementación

El objetivo principal de este flujo de trabajo es desarrollar la arquitectura y sistema como un todo. Aquí se implementa el sistema en componentes como código fuente, y ejecutables entre otros.

2.3.4.5. Pruebas

Las pruebas verifican lo que resultó de la implementación y prueban las construcciones internas así como las versiones finales del sistema a entregarse.

Se planifican las pruebas de integración en cada iteración y las pruebas de sistema al final de cada iteración. Igualmente se diseñan e implementan los casos de prueba que indican que probar y se crean los procedimientos de prueba que señalan como hacer las pruebas. Finalmente se ejecutan las diferentes pruebas y se tratan los resultados de las construcciones, en caso de defectos se prueban nuevamente o son devueltas a un flujo de trabajo anterior.

2.3.5. Lenguaje de Modelado Unificado (UML) (Binder, R 2000)

El Lenguaje de Modelado Unificado es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; aún cuando todavía no es un estándar oficial, está apoyado en gran manera por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software.

UML es una especificación de notación orientada a objetos. Se basa en las anteriores especificaciones Booch, Rumbaugh y Coad-Yourdon. Divide cada proyecto en un número de diagramas que representan las diferentes vistas del proyecto. Estos diagramas juntos son los que representa la arquitectura del proyecto. Su utilización es independiente del lenguaje de programación y de las características de los proyectos, ya que UML ha sido diseñado para modelar cualquier tipo de proyectos, tanto informáticos como de arquitectura, o de cualquier otro ramo.

Este lenguaje permite especificar, construir, visualizar y documentar los objetos de un sistema programado. Puede ser utilizado por cualquier metodología de

análisis y diseño orientada por objetos para expresar las relaciones estáticas o dinámicas que existen entre ellos. Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- **Visualizar:** UML permite expresar de una forma gráfica un sistema.
- **Especificar:** UML permite especificar cuáles son las características de un sistema antes de su construcción.
- **Construir:** A partir de los modelos especificados se pueden construir los sistemas diseñados.
- **Documentar:** Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Aunque UML está pensado para modelar sistemas complejos con gran cantidad de Software, el lenguaje es lo suficientemente expresivo como para modelar sistemas que no son informáticos, como flujos de trabajo (workflow) en una empresa, diseño de la estructura de una organización y por supuesto, en el diseño de hardware. Un modelo UML está compuesto por tres clases de bloques de construcción:

- **Elementos:** Son abstracciones de cosas reales o ficticias (objetos, acciones).
- **Relaciones:** Relacionan los elementos entre sí.
- **Diagramas:** Son colecciones de elementos con sus relaciones.

2.3.6. Tipos de Modelo (Binder, R 2000)

- **Modelo Funcional:** Muestra la funcionalidad del sistema desde el punto de vista del usuario, incluye:
 - **Diagramas de caso de uso:** Describe lo que hace un sistema desde el punto de vista de un observador externo, debido a esto, un diagrama de

este tipo generalmente es de los más sencillos de interpretar en UML, ya que su razón de ser se concentra en un Qué hace el sistema, a diferencia de otros diagramas UML que intentan dar respuesta a un Como logra su comportamiento el sistema.

- **Modelo de Objetos:** Muestra la estructura y la subestructura del sistema usando objetos, atributos, operaciones y asociaciones, incluye:
- **Diagramas de clase:** Representa las clases que serán utilizadas dentro del sistema y las relaciones que existen entre ellas. Los diagramas de Clases por definición son estáticos, esto es, representan que partes interactúan entre sí.
- **Modelo Dinámico:** Muestra el comportamiento interno del sistema, incluye:
 - **Diagramas de secuencia y colaboración:** Muestran a los diferentes objetos y las relaciones que pueden tener entre ellos, los mensajes que se envían entre ellos. Son dos diagramas diferentes, que se puede pasar de uno a otro sin pérdida de información, pero que nos dan puntos de vista diferentes del sistema. En resumen, cualquiera de los dos es un Diagrama de Interacción.
 - **Diagramas de actividades:** Demuestra la serie de actividades que deben ser realizadas en un caso de uso, así como las distintas rutas que pueden irse desencadenando en dicho caso y así auxiliar a los miembros del equipo de desarrollo a entender como es utilizado el sistema y cómo reacciona en determinados eventos.

- **Diagramas de estado:** Muestra los estados, eventos, transiciones y actividades por los cuales pasa un objeto durante su vida en una aplicación, junto con los cambios que permiten pasar de un estado a otro. Son útiles en sistemas que reaccionen a eventos.

2.3.7. WebML (Politecnico Di Milano, Webml Site 2003)

Hasta el día de hoy, lo más importante en el desarrollo de aplicaciones Web han sido las herramientas, pero muy poco se ha dicho y escrito sobre el proceso de desarrollo. La fácil creación de páginas HTML y en general de sitios Web, usando herramientas simples, ha hecho que el desarrollo de éste tipo de aplicaciones se haga sin un trabajo serio de análisis y diseño. Cualquier sistema de complejidad no trivial, necesita ser analizado y modelado. Las aplicaciones Web, al igual que otras aplicaciones, necesitan métodos y técnicas formales de análisis y diseño.

Todas las metodologías propuestas para las aplicaciones Web desde mediados de los años noventa presentan su propia notación para casi todos sus diagramas. Una recopilación excelente de varios de estos métodos fue presentada en el primer taller internacional sobre tecnologías de software orientadas a Web. En ella se describen cada una de estas propuestas en base a un mismo caso de estudio.

Jim Conallen, a finales de los noventa, presentó una extensión de UML basada en las técnicas de implementación actuales de ese tiempo. Esta extensión propuesta por Conallen es bastante útil como técnica de diseño para cuando la implementación de la aplicación que está siendo modelada se encuentra próxima a realizarse o se está realizando, pero no cuando se trata de modelar una aplicación como parte de la planificación de un proyecto. Estas metodologías, y otras muchas que se han propuesto, contribuyen con ideas importantes para el diseño de software orientado a Web.

El método Hipermedia Orientado a Objetos (Object-oriented Hypermedia, OOH) proporciona un conjunto de nuevas vistas que extienden el UML para modelar únicamente la interfaz de aplicaciones Web.

En este contexto, la Ingeniería Web Basada en UML (UML-Based Web Engineering, UWE) es una propuesta de metodología que recupera muchos de los conceptos descritos y utilizados en otras metodologías e incorpora nuevos elementos que permiten describir de mejor manera una aplicación Web.

Otra propuesta interesante además de ser motivo de estudio e implementación en este trabajo es el lenguaje de modelado Web (Web Modeling Language, WebML). El WebML es una notación visual para el diseño de aplicaciones Web complejas y con gran volumen de datos, ya que provee especificaciones gráficas formales envueltas en un completo proceso de diseño que puede ser asistido por herramientas visuales de diseño; a su vez permite la descripción de los sitios Web desde distintos puntos de vista entre los que podemos mencionar el conceptual, el navegacional y el de presentación.

WebML apunta a proveer un acercamiento en la estructura del diseño intensivo de datos que se manejan en un sitio Web. Un conjunto de modelos integrados ayudan a los diseñadores en la producción del sitio de alta calidad ya que todas las facetas del diseño deben ser y son dirigidas. El uso de metodologías viejas se vuelve despreciado.

Los principales objetivos del proceso de diseño WebML son:

- Expresar la estructura de una aplicación Web con un alto nivel de descripción que pueda ser usado para consultas, evolución y mantenimiento futuro de la aplicación.

- Provee múltiples vistas del mismo contenido.
- Descompone la información en páginas, navegación y presentación, que pueden ser definidas de manera independiente.
- Permite almacenar la meta - data recolectada durante el diseño para emplearlos durante toda la vida de la aplicación para generar páginas Web de forma dinámica.
- Sitios Web que manejen gran cantidad de datos haciendo uso de interfaces dirigidas al público en general.

El desarrollo de aplicaciones Web con WebML consiste en diferentes fases que deben ser aplicadas de manera iterativa e incremental. El proceso involucra varios ciclos, cada uno de los cuales produce un prototipo o versión parcial de la aplicación, lo que permite realizar evaluaciones y pruebas desde las fases iniciales de desarrollo.

El proceso de desarrollo comienza con el Modelo de Datos del sistema en el que mediante algún lenguaje de modelado como UML (WebML no exige ninguno en concreto), se representa la estructura estática del mismo. Tras esto, se realiza el Modelo de Hipertexto donde se describen uno o más hipertextos que pueden ser publicados en el sitio Web. Cada uno de estos hipertextos define una vista del sitio. La descripción de los hipertextos se realiza mediante dos modelos: el Modelo de Composición, que define las páginas que componen el sistema, y el Modelo de Navegación, que describe cómo se podrá navegar a través de ellas. Y por último, se describe el Modelo de Presentación que define la apariencia física de las páginas.

Los principales modelos incluidos en la metodología de diseño WebML son:

- Modelo de Datos.
- Modelo de Hipertexto.

- Modelo de Presentación.

2.3.7.1. Modelo de Datos

La meta del modelado de datos es permitir la especificación de los datos usados por la aplicación, en una manera formal e intuitiva. El resultado del modelado de datos es un esquema conceptual, el cual transmite en una forma simple y legible el conocimiento disponible sobre los datos de la aplicación. Diseñar tal esquema es preliminar al diseño de las funciones de negocio que operan los datos, y a la implementación de las estructuras físicas que mantienen el almacenamiento, actualización y consulta de los datos.

2.3.7.2. Modelo de Hipertexto

La meta del modelado de hipertexto es especificar la organización de las interfaces de la aplicación web. Para ser efectiva, tal especificación debe ser capaz de transmitir en una manera simple e intuitiva tales aspectos de la división lógica de la aplicación en módulos de alto nivel, cada uno representando un conjunto de funciones coherentes apuntando a una clase específica de usuarios, la partición de los módulos de alto nivel en submódulos, para una mejor organización de grandes aplicaciones, y la actual topología de hipertexto de cada modulo, en términos de páginas, hechas de elementos de contenido, y enlazadas a apoyar la navegación de los usuarios y la interacción.

El modelo de hipertexto debería estar al nivel correcto de abstracción; la especificación del hipertexto debe ser mantenida a un nivel conceptual, el cual significa que no debe comprometer mucho el diseño y la implementación de los detalles, tales como la distribución actual de la funcionalidad entre las varias capas de la aplicación web.

2.3.7.3. Elementos del modelo Hipertexto WebML

Esencialmente un diagrama en WebML cubre la información básica: Organiza la aplicación en subsistemas, expresados en hipertextos y en áreas, su correspondiente descomposición en áreas y páginas, los componentes que constituyen una página, y las asociaciones (links) entre páginas y unidades.

Esta información está representada en un nivel abstracto, correspondiendo así la terminología orientada a objeto de UML con el nivel analítico-descriptivo del sistema. De hecho, un diagrama WebML ignora a propósito la manera en cómo las funciones del hipertexto están particionadas entre los componentes del lado de servidor y los objetos del lado del cliente que son ejecutados en el buscador, pero concentrándose en una vista lógica de la aplicación en sí.

Estas consideraciones sugieren que el uso de los diagramas de clase de UML constituye la herramienta más apropiada para representar la información de un diagrama en WebML.

La conversión de un esquema WebML a un diagrama de clases de UML procede acorde con las siguientes consideraciones:

- Los hipertextos y las áreas son expresados en paquetes anidados, representando la descomposición de la aplicación. Las páginas son denotadas por clasificadores con el estereotipo <<page>>.
- Una página es considerada como un objeto complejo, conformado por subcomponentes y a la vez asociados con otros objetos. Las páginas conjuntivas (AND) son simplemente denotadas con una página anidada. Las páginas disyuntivas (OR) son denotadas con una página anidada y el estereotipo <<alternativa>>.

- Las unidades son representadas como clases asociadas como partes de una asociación con la página que las involucra, representadas como unidades anidadas dentro de la misma.
- Los links entre páginas o unidades son representados como asociaciones orientadas, con el estereotipo <<link>>. Los parámetros de los links están expresados con cada uno de los valores asociados a los mismos.
- Los datos que se pasan entre páginas entre son representados mediante una flecha punteada, la cual contiene la etiqueta expresada con cada uno de los valores asociados a los mismos.

La tabla 2.1 muestra los estereotipos en WebML, su respectiva descripción y el elemento UML asociado.


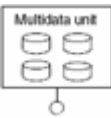


2.3.7.4. Modelo de Presentación

Define como lucirá la vista del sitio. WebML incluye un modelo simple de presentación que permite colocar contenidos dinámicos en la página además de aplicar estilos distintos para cada uno.

2.3.8. Programación Orientada a Objetos (POO) (Schach, S 2005)


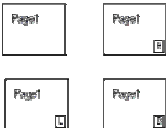
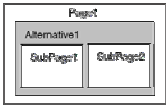
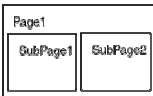
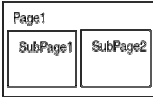
La Programación Orientada a Objetos (POO) es una forma especial de programar, promete mejoras de amplio alcance en la forma de diseño, desarrollo y mantenimiento del software, ofrece una solución a largo plazo a los problemas y preocupaciones que han existido desde el comienzo en el desarrollo de software: la falta de portabilidad del código y reusabilidad, código que es difícil de modificar, ciclos de desarrollo largos y técnicas de codificación no intuitivas.

Tabla 2.1. Elementos del Modelo de Hipertexto WebML (1/5)

Elemento Web ML	Descripción	Atributos
<p data-bbox="386 405 596 474">Data Unit (Unidad de Datos)</p> 	<p data-bbox="678 405 1013 520">La unidad de datos publica un solo objeto obtenido de una entidad determinada.</p>	<p data-bbox="1039 405 1263 657">Nombre. Entidad Fuente. Contenido: Subpáginas anidadas Selector (opcional). Atributos Incluidos.</p>
<p data-bbox="337 726 647 795">Multidata Unit (Unidad de Datos Múltiple)</p> 	<p data-bbox="678 726 1013 884">Presenta múltiples objetos de una entidad juntos, repitiendo la presentación de muchas unidades de dato</p>	<p data-bbox="1039 726 1360 936">Nombre. Entidad Fuente. Selector (opcional). Atributos Incluidos. Cláusula de Orden (opcional).</p>
<p data-bbox="332 1014 649 1083">Scroller-Unit (Unidad de desplazamiento)</p> 	<p data-bbox="678 1014 1013 1171">Una unidad de desplazamiento provee comandos para desplazar los objetos en un escenario.</p>	<p data-bbox="1039 1014 1360 1224">Nombre. Entidad Fuente. Selector (opcional). Bloque de factores. Cláusula de Orden (opcional).</p>
<p data-bbox="337 1323 649 1438">Multi-choice index Unit (Unidad índice de múltiples elecciones)</p> 	<p data-bbox="678 1323 1013 1570">Es una variante de la unidad de índice, donde cada elemento de la lista está asociado con un checkbox; permitiendo a los usuarios seleccionar múltiples objetos</p>	<p data-bbox="1039 1323 1360 1533">Nombre. Entidad Fuente. Selector (opcional). Atributos Incluidos. Cláusula de Orden (opcional).</p>





Fuente: "Politécnico de Milano". Webml Training.

Tabla 2.1. Elementos del Modelo de Hipertexto WebML (2/5)

Elemento Web ML	Descripción	Atributos
<p>Index Unit (Unidad Índice)</p> 	Una unidad índice presenta objetos múltiples de una entidad como una lista.	Nombre. Entidad Fuente. Selector (opcional). Atributos Incluidos. Cláusula de Orden (opcional).
<p>Page (Pagina)</p> 	Representan la actual interfaz buscada por el usuario. Esta contiene sub-paginas AND/OR.	Nombre Punto de referencia Contenido: Unidades o subpágina
<p>OR sub-pages (Sub-Páginas OR (alternativo))</p> 	Especifican que hay partes de la página pueden tener contenidos alternativos	Nombre Punto de referencia Contenido: Subpáginas alternativas
<p>AND Sub-Pages (Sub-Páginas AND)</p> 	Son usadas para dividir el contenido de la página en una pantalla en partes. Las sub-páginas AND están contenidas en una página o en una sub-página.	Anidación de Páginas.
<p>AND Sub-Pages (Sub-Páginas AND)</p> 	Son usadas para dividir el contenido de la página en una pantalla en partes. Las sub-páginas AND están contenidas en una página o en una sub-página.	Anidación de Páginas.


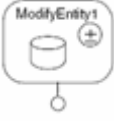



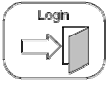
Fuente: "Politécnico de Milano". Webml Training.

Tabla 2.1. Elementos del Modelo de Hipertexto WebML (3/5)

Elemento Web ML	Descripción	Atributos
<p>Vista del Sitio (site view)</p> 	<p>Representa una vista del hipertexto.</p>	<p>Nombre. Contenido: páginas, áreas. Página de inicio.</p>
<p>Área</p> 	<p>Es un contenedor de páginas o, recursivamente, otras sub-áreas, las cuales pueden ser usadas para darle una organización jerárquica al hipertexto</p>	<p>Nombre. Marca Contenido: páginas, sub-áreas. Página por defecto o subárea.</p>
<p>Entry Unit (Unidad de entrada)</p> 	<p>Soporta entrada de datos basada en formulario</p>	<p>Nombre. Para cada campo: Nombre. Tipo. Valor inicial. Predicado de validez.</p>
<p>Create Unit (Unidad de creación)</p> 	<p>Permite la creación de un nuevo ejemplo de entidad (instancias).</p>	<p>Nombre. Entidad Fuente. Conjunto de valores asignados.</p>






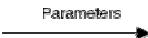

Fuente: "Politécnico de Milano". Webml Training.

Tabla 2.1. Elementos del Modelo de Hipertexto WebML (4/5).

Elemento Web ML	Descripción	Atributos
Eliminar Entidad 	Elimina uno o más objetos de una entidad dada.	Nombre Entidad de origen Selector.
Modificar Entidad 	Modifica o actualiza uno o más objetos de una entidad dada	Nombre Entidad de origen Selector Conjunto de asignaciones de valores
Conectar 	Crea nueva instancia de una relación	Nombre Rol de la relación Selector de la entidad de origen Selector de la entidad de destino
Desconectar 	Elimina una instancia de una relación	Nombre Rol de la relación Selector de la entidad de origen Selector de la entidad de destino
Link de Transporte 	Link empleado sólo para el transporte de datos entre unidades, el cual no está asociado con ningún ancla	Nombre Elemento origen Elemento destino Parámetros de enlace
Iniciar Sesión 	Verifica la entidad de un usuario accediendo al sitio.	Parámetro: Nombre de usuario. Clave.

Fuente: "Politécnico de Milano". Webml Training.

Tabla 2.1. Elementos del Modelo de Hipertexto WebML (5/5)

Elemento Web ML	Descripción	Atributos
Cerrar Sesión 	Lleva al usuario a la página principal sin control de acceso	Ninguno
Enviar E-mail 	Provee la capacidad de enviar mensajes de email	Remitente Recipiente Asunto Cuerpo Adjuntos
Unidad de operación 	Define una operación genérica: los parámetros de entrada y salida deben estar definido por el diseñador	Definidos por el diseñador
OK Link / KO Link  	OK Link: se ejecuta en caso de éxito en la operación. KO Link: se ejecuta en caso de que la operación falle.	Nombre Elemento origen Elemento destino Parámetros de enlace
Link Directo 	Establece el flujo de datos entre unidades	Nombre Elemento origen Elemento destino Parámetros de enlace
Link Automático 	Se ejecutan sin necesidad de la intervención del usuario.	Nombre Elemento origen Elemento destino Parámetros de enlace

Fuente: "Politécnico de Milano". Webml Training.

El elemento fundamental de la POO es, como su nombre lo indica, el objeto. Un objeto es un conjunto complejo de datos y programas que poseen estructura y forman parte de una organización.

Esta definición especifica varias propiedades importantes de los objetos. En primer lugar, un objeto no es un dato simple, sino que contiene en su interior cierto número de componentes bien estructurados. En segundo lugar, cada objeto no es un ente aislado, sino que forma parte de una organización jerárquica o de otro tipo.

2.3.8.1. Ventajas de la POO

- **Uniformidad.** Ya que la representación de los objetos lleva o implica tanto el análisis como el diseño y la codificación de los mismos.
- **Comprensión.** Tanto los datos que componen los objetos, como los procedimientos que los manipulan, están agrupados en clases, que se corresponden con las estructuras de información que el programa trata.
- **Flexibilidad.** Al tener relacionados los procedimientos que manipulan los datos con los datos a tratar, cualquier cambio que se realice sobre ellos quedará reflejado automáticamente en cualquier lugar donde estos datos aparezcan.
- **Estabilidad.** Dado que permite un tratamiento diferenciado de aquellos objetos que permanecen constantes en el tiempo sobre aquellos que cambian con frecuencia, permite aislar las partes del programa que permanecen inalterables en el tiempo.
- **Reusabilidad.** La noción de objeto permite que programas que traten las mismas estructuras de información reutilicen las definiciones de objetos empleadas en otros programas e incluso los procedimientos que los manipulan. De esta forma, el desarrollo de un programa puede llegar a ser una

simple combinación de objetos ya definidos donde estos están relacionados de una manera particular.

CAPÍTULO III

FASE DE INICIO

3.1. Introducción

En La fase de inicio se busca captar la mayor cantidad de casos de uso, identificación de actores, descripción de casos de uso y requerimientos adicionales necesarios para fundamentar el análisis inicial. Este análisis se divide en cuatro pasos:

- Delimitar el ámbito del sistema propuesto, es decir, definir los límites del sistema, definir los requerimientos e identificar las interfaces con sistemas relacionados que están fuera de los límites.
- Describir una propuesta de la arquitectura del sistema (en especial en aquellas partes que son nuevas, arriesgadas o difíciles).
- Identificar riesgos críticos (los que afectan la capacidad de construir el sistema) y determinar si podemos encontrar una forma de mitigarlos, quizás en una etapa posterior.
- Demostrar a usuarios o clientes potenciales que el sistema propuesto es capaz de solventar sus problemas o de mejorar sus objetivos de negocio construyendo un prototipo.

3.2. Flujo de Trabajo de Requisitos

La captura de requisitos es el proceso de investigar, normalmente en circunstancias difíciles, lo que se debe construir. El flujo de trabajo requisitos tiene como propósito guiar el desarrollo del proyecto hacia el sistema correcto hasta

obtener una arquitectura sólida. Se basa en la comprensión del contexto del sistema usando el diagrama del modelo de dominio, estudio del contexto del sistema, requerimientos funcionales, lista de riesgos, modelos de casos de uso, identificación de actores, descripción de casos de uso y requerimientos adicionales.

Los requisitos están comprendidos en dos grupos. Los requisitos funcionales que representa la funcionalidad primaria del sistema, se modelan mediante diagramas de casos de uso. Los requisitos no funcionales que representan aquellos atributos que debe exhibir el sistema, pero que no son una funcionalidad específica.

3.2.1. Estudio del contexto del sistema

Para comprender el contexto del sistema fue necesario conocer, estudiar y analizar las herramientas utilizadas por la Superintendencia de Mantenimiento de la Plataforma (MAP) para monitorear los equipos y generar casos, de igual forma se estudio las actividades realizadas por la gerencia. Para ello es necesario lo siguiente:

- Realizar entrevistas y visitas a los funcionarios que participan en dicha gerencia.
- Revisar la documentación de NAGIOS.

Todo esto con el fin de conocer el proceso que se lleva a cabo e identificar las debilidades que existen, de manera que éstas puedan ser corregidas a futuro con el sistema. Una vez estructuradas las informaciones necesarias se consiguió una aproximación del contexto del sistema representado mediante el modelo de dominio, describiendo los conceptos importantes del contexto como objeto del dominio y enlazando estos objetos entre sí.

3.2.1.1 Modelo de Dominio

El modelo de dominio se encarga de capturar los objetos más importantes vinculados al contexto del sistema. Estos objetos representan las “cosas” que existen actualmente o los eventos que tienen lugar en el entorno donde se desenvuelve el sistema.

Un modelo de dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. También se les denomina modelos conceptuales, modelo de objetos del dominio y modelos de objetos de análisis.

Para la descripción del modelo de dominio debe realizarse diagramas de clases basado en Lenguaje de Modelado UML.

El proceso manual se realiza de la siguiente manera: NAGIOS monitorea los equipos que conforman Servicios Comunes Oriente. Estos equipos presentan fallas las cuales generan alarmas, dichas alarmas son consultadas por los analistas del Centro Integral de Monitoreo Oriente (CIMOR), estos realizan las documentaciones correspondientes mediante planillas físicas, y generan un caso.

Cuando se genera una falla que corresponde a un evento, el analista de monitoreo debe llenar una planilla con todos los datos asociados al evento, y registrar en la misma la documentación del caso.

El analista de CIMOR debe realizar informes semanales de todas las alarmas generadas por NAGIOS y los eventos documentados, para así generar graficas en una hoja de cálculo sobre las fallas con mayor impacto.

El analista de Control de Gestión (CDG), solicita semanal y mensualmente la información de los casos generados para enviar a los superintendentes de las disciplinas que conforman el Departamento de Mantenimiento de la Plataforma, un reporte de todas las actividades de mantenimiento, realizadas a los diferentes casos generados durante ese tiempo.

Así concluye el proceso manual que conlleva a retardos en los tiempos de respuesta del personal que necesitan las gerencias.

Las clases que se presentan en el modelo de domino son las siguientes:

- **NAGIOS:** Herramienta encargada de monitorear los equipos pertenecientes a la plataforma de AIT (Automatización Informática y Telecomunicaciones) Servicios Comunes Oriente.
- **Alarma:** Fallas generadas por la herramienta de monitoreo, en este caso NAGIOS.
- **Caso:** Falla después que el analista documenta.
- **Evento:** Fallas que NAGIOS no genera debido a que los equipos no están registrados en la herramienta de monitoreo.
- **Servicio:** Servicios de red utilizados por la Superintendencia de Mantenimiento de la Plataforma.
- **Informe:** Soporte que muestra las alarmas y los eventos generados.
- **Analista de CIMOR:** Persona encargada de la visualización, documentación, localización y notificación oportuna de eventos al personal de soporte y mantenimiento
- **Analista CDG:** Persona encargada de solicitar y revisar los informes sobre las fallas generadas.

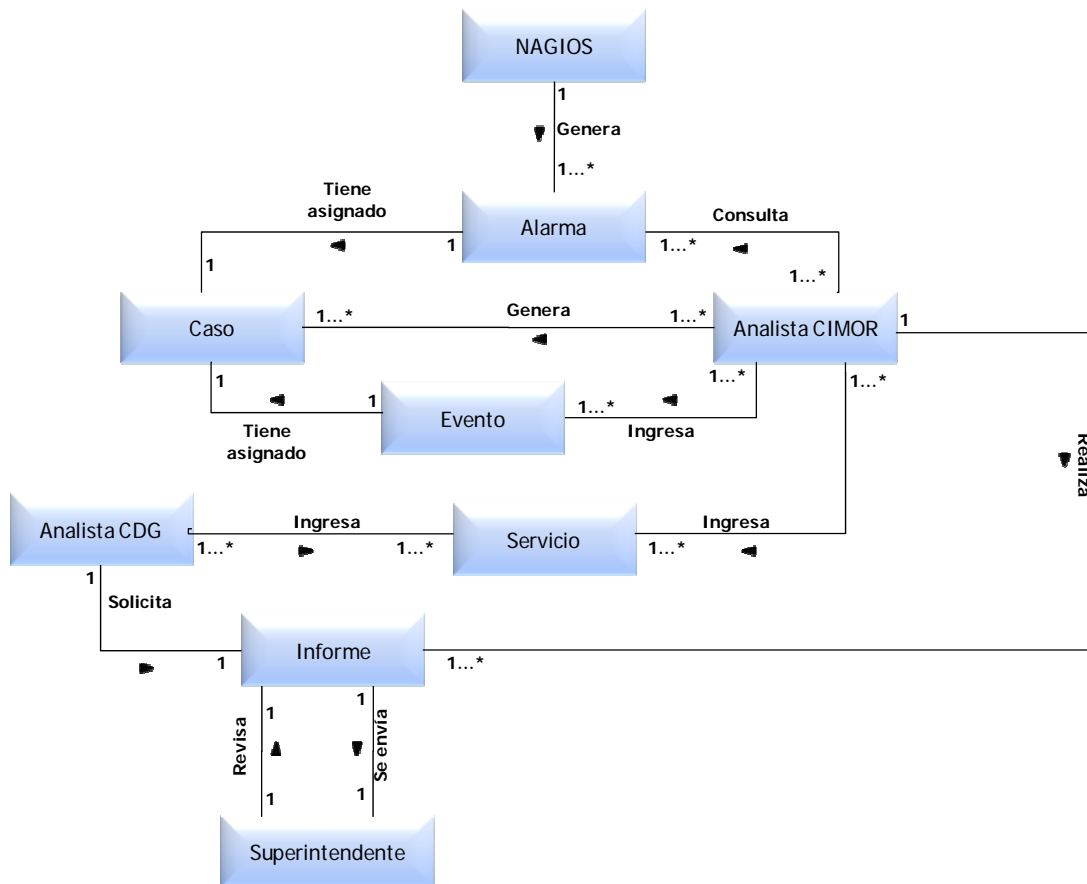


Figura 3.1. Modelo de Dominio.

Fuente: Elaboración Propia.

3.2.2. Requerimientos esenciales del sistema

- Automatizar el proceso de gestión de las alarmas generadas por Nagios.
- Guardar en la base de datos toda la información referente a las alarmas, la documentación correspondiente a la misma, y el personal asociado a ella para así poder realizar reportes automatizados de cualquier índole, reduciendo y eliminando el almacenamiento de información en archivos, y la duplicidad de información.

- Realizar una interfaz visual, fácil y cómoda que permita la interacción entre los usuarios y el sistema.
- Establecer claves de acceso al personal asociado, para mantener la seguridad del sistema.

3.2.3. Requisitos Funcionales del Sistema

- Registro, consulta y actualización de usuarios.
- Registro, consulta actualización y documentación de las alarmas generadas por NAGIOS.
- Registro, consulta y actualización de opciones avanzadas del sistema como: tipo de equipos, tipo de elementos, disciplina y localidad.
- Registro consulta y actualización de eventos.
- Registro consulta y actualización de los servicios utilizados por la Gerencia de Automatización Informática y Telecomunicaciones (AIT) Servicios Comunes Oriente.
- Generar casos de las alarmas.
- Generar Reportes de las alarmas de acuerdo a su estado.
- Generar Reportes de los indicadores operativos utilizados por la Superintendencia MAP.

3.2.4. Requisitos No Funcionales del Sistema

Los requisitos no funcionales son los requisitos que imponen restricciones al diseño o funcionamiento del sistema (tal como requisitos de funcionamiento, estándares de calidad, o requisitos del diseño). Estos requisitos especifican propiedades del sistema, como restricciones del entorno o de implementación, rendimiento, dependencias de la plataforma, mantenibilidad, extensibilidad o fiabilidad y responden esencialmente a limitaciones arquitectónicas y de rendimiento. Ver tabla 3.1.

Tabla 3.1. Requisitos adicionales del sistema.

Requisitos	Descripción
Requisitos del Software	El sistema debe estar desarrollado utilizando tecnologías de software libre en cumplimiento al decreto presidencial 3390.
Plataforma Hardware	El software debe ser implementado en un Servidor.
Rendimiento	El tiempo de respuesta de la aplicación depende de las características físicas de los equipos y de la velocidad de procesamiento de los computadores.
Usabilidad	Interfaz amigable para las personas autorizadas que usarán la aplicación, la cual debe generar comodidad durante la navegación, siempre enfocado a las actividades del negocio empresarial.
Disponibilidad	La aplicación estará disponible en todo momento para los usuarios autorizados.
Seguridad	Se implementan claves de acceso para el control de ingreso al sistema logrando así la protección de la información.
Mantenimiento	El administrador del sistema con su clave de acceso deberá poder insertar, modificar, eliminar y probar módulos y procesos del sistema lo cual generará un correcto mantenimiento del mismo.

Fuente: Elaboración Propia.

3.2.5. Identificación de los Riesgos

Todo desarrollo de software tiende a presentar una serie de riesgos que de no ser solucionados a tiempo, trae como consecuencia problemas posteriores que con el

transcurrir del tiempo son más difíciles de solucionar, razón por la cual es necesario tomar en cuenta dichos riesgos, desde los inicios del desarrollo del sistema.

Cada riesgo puede ser solventado totalmente o en gran medida por medio de la realización de los casos de uso del sistema para dichos riesgos, utilizando los requisitos funcionales y no funcionales ya encontrados y especificados.

La tabla 3.2 muestra los posibles riesgos del sistema donde se hace una breve descripción del riesgo, luego se indica el nivel y la parte del sistema afectada, además se señala el responsable de tratarlo y la ejecución necesaria para solventar dichos riesgos.

3.2.6. Modelos de Casos de Uso

El modelo de casos de uso permite capturar aquellos requisitos que tienen un valor agregado para cada usuario o para cada sistema externo, proporcionando la entrada fundamental para el análisis, el diseño y las pruebas del sistema. Cada usuario necesita de varios casos de uso, los cuales representarán las diferentes formas de utilizar el sistema, por esto, cada caso de uso denota una manera de darle soporte al usuario durante los distintos procesos involucrados.

3.2.6.1. Actores del Sistema

Los actores son personas, sistemas o hardware externo que interactúan con el sistema. Pueden proveer funcionalidades al sistema, así como utilizar las ya provistas por éste, por tanto pueden ingresar o capturar datos al sistema.

Cada actor tiene asignado un rol dentro de la aplicación, lo que determina las acciones que este puede realizar en el sistema. En la tabla 3.3 se especifican los usuarios del sistema y cada una de las acciones que pueden realizar dentro del mismo.

Tabla 3.2. Riesgos del Sistema (1/2).

Módulo	Descripción	Nivel	Responsable	Ejecución
Hardware	Bajo rendimiento del PC por insuficiencia de memoria RAM y/o poca capacidad de procesador	Crítico	PDVSA	Instalar memorias RAM y procesador de más capacidad.
	Tarjeta de red no funciona o no está instalada.	Crítico		Instalar nueva tarjeta de red.
	Poco espacio en disco duro	Crítico		Instalar disco duro de mayor capacidad.
	Insuficiente capacidad para almacenar datos	Crítico		Verificar espacio en HDD Probar otros manejadores de B.D.
Software	Selección de herramientas de desarrollo inadecuadas	Crítico	Desarrollador	Estudiar las capacidades de las herramientas seleccionadas y sus posibles reemplazos.
	Fallas en acceso a la base de datos	Muy Crítico		Establecer un correcto diseño de la Base de Datos. Verificar bases de datos.

Fuente: Elaboración Propia.

Tabla 3.2. Riesgos del Sistema (2/2)

Módulo	Descripción	Nivel	Responsable	Ejecución
Interfaz	Incompatibilidad con navegadores Web.	Crítico	Desarrollador	Probar la interfaz en distintos navegadores y buscar la mejor configuración que se adapte a ellos.
	No lograr una interfaz adecuada para las necesidades del sistema	Muy Crítico		Profundizar el análisis del flujo de actividades con apoyo del futuro usuario
	Menú de ayuda incomprensible.	No crítico		Realizar ayuda con interfaz amigable y fácil de entender para el usuario
Servidor	Inexistencia de servicios de Internet.	Crítico	PDVSA	Comprobar si existe o funciona el servicio de Internet., en caso contrario contratarlo.

Fuente: Elaboración Propia.

Tabla 3.3. Identificación de los actores del sistema.

Actores	Descripción
Administrador del Sistema	Encargado del control del sistema, quien bajo el rol de administrador tiene acceso a todos los módulos del sistema, lo que incluye la administración de usuarios (creación de cuentas, registro, actualización y retiro de usuarios, etc.).
Superintendente	Personal autorizado por la empresa para generar reportes, y visualizar ayuda. Cuentan con claves de acceso y nombre de usuario
Analista de Control de Gestión	Personal encargado de gestionar los servicios utilizados por la gerencia de AIT, además de tener acceso a reportes y visualizar ayuda. Cuentan con claves de acceso y nombre de usuario
Analista de Monitoreo	Personal encargado de gestionar toda la información referente a las alarmas, los eventos, servicios y generar reportes. Cuentan con claves de acceso y nombre de usuario
NAGIOS	Representa la herramienta de monitoreo encargada de generar las alarmas, ingresadas en el sistema.

Fuente: Elaboración Propia.

3.2.6.2. Casos de Uso del Sistema

Luego de las entrevistas con el personal en busca de los requisitos funcionales, y de la selección de los casos de usos que representaban mejor la funcionalidad del sistema, se realizó el diagrama de caso del sistema integrado de comunicación y notificación de alarmas (SIDENCA). Ver figura 3.2.

3.2.7. Descripción de los Casos de Uso

El modelo de caso de uso general del sistema es un diagrama que muestra las acciones principales que ejecutará el sistema que se está diseñando.

Se aborda cada caso de uso de forma detallada, de tal manera que se puedan descomponer las distintas acciones que el sistema puede llevar a cabo al interactuar con sus actores.

Se denota un flujo de sucesos para cada caso de uso que especifica la secuencia de acciones del mismo. La tabla 3.4 muestra la descripción y el flujo de sucesos de los casos de uso detallado del sistema.

3.2.7.1. Caso de Uso Administrar Usuarios

Actores: Administrador del sistema.

Descripción: Permite listar, ingresar, consultar, modificar o eliminar algún usuario.

Pre-condición: El administrador del sistema debe ingresar al menú usuarios y seleccionar la opción que desee realizar.

Flujo de eventos:

Flujo Principal:

1. El administrador del sistema invoca al caso de uso administrar usuarios, y selecciona la opción a efectuar.

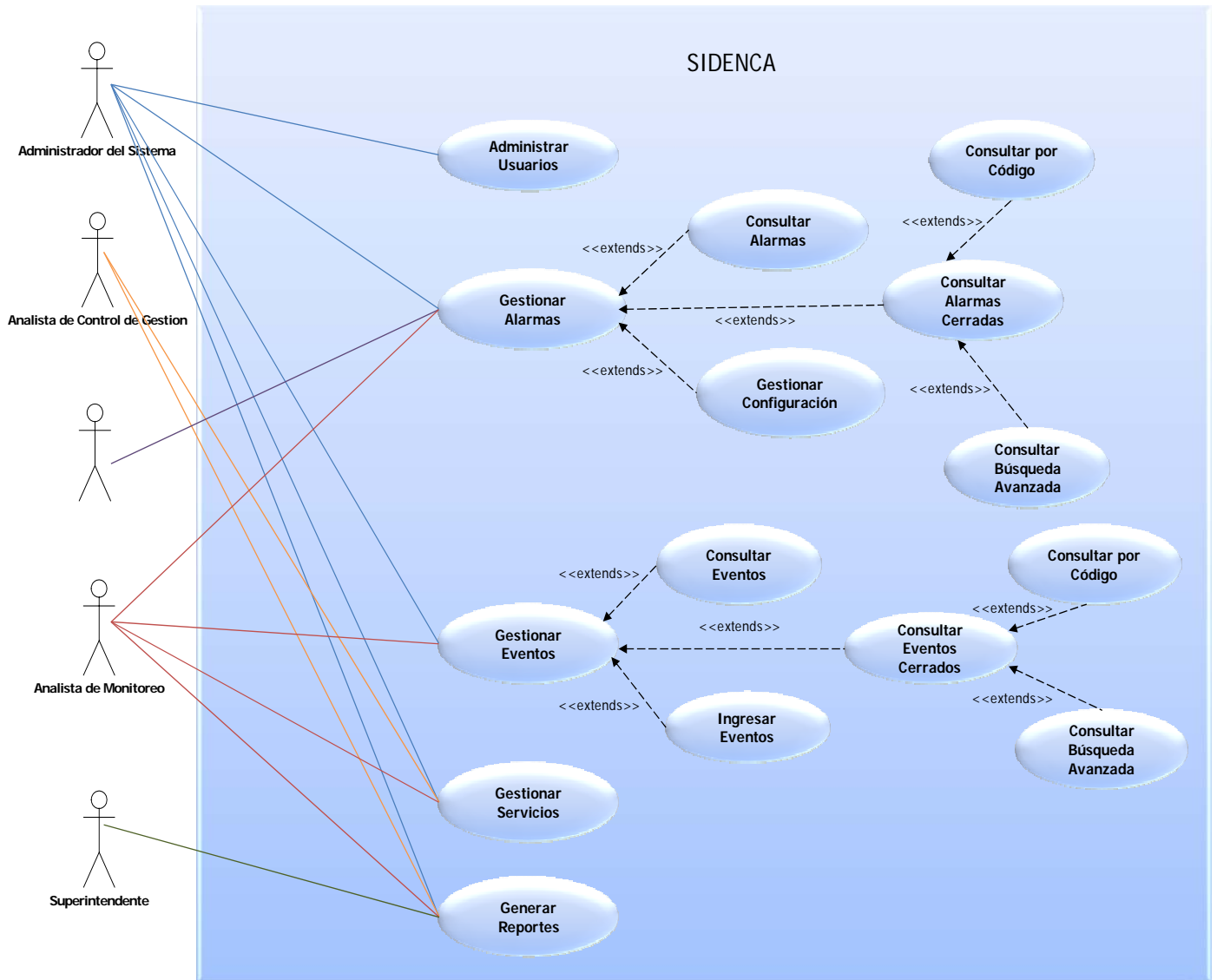


Figura 3.2. Diagrama de Caso de Uso general del sistema

Fuente: Elaboración Propia

NAGIOS

Tabla 3.4. Descripción de los casos de uso del sistema con sus actores.

Caso de Uso	Actores	Descripción
Administrar Usuarios	Administrador del Sistema	Presenta las actividades necesarias para el control de acceso al sistema, garantizando que sólo usuarios registrados puedan iniciar sesión.
Gestionar Alarmas	Administrador del Sistema, Analista de Monitoreo, NAGIOS	Se maneja toda la información relacionada con las alarmas provenientes de NAGIOS, las alarmas que han sido cerradas, así como la gestión de las opciones de configuración.
Gestionar Eventos	Administrador del Sistema, Analista de Monitoreo	Permite a los actores involucrados, ingresar y consultar las fallas que NAGIOS no reporta, así como manejar toda la información relacionada con los eventos cerrados.
Gestionar Servicios	Administrador del Sistema, Analista de Monitoreo, Analista de Control de Gestión.	Permite al analista manejar toda la información relacionada con los servicios que son utilizados por la gerencia.
Generar Reportes	Administrador del Sistema, Analista de Monitoreo, Analista de Control de Gestión, Superintendente	Cuenta con opciones de generar gráficas y reportes de las alarmas y eventos, para un mejor manejo de los indicadores de la plataforma.

Fuente: Elaboración Propia.

2. El sistema despliega un formulario dependiendo de la opción seleccionada, en el cual el administrador del sistema debe ingresar los datos bien sea de búsqueda, registro, actualización o retiro.
3. El usuario solicita, tomando en cuenta la opción elegida, almacenar los datos, actualizarlos, eliminar un usuario, o mostrar los datos consultados.
4. El sistema guarda los cambios realizados.
5. Finaliza el caso de uso.

Flujo Alternativo:

Paso 3. El usuario cancela la operación y vuelve a la pantalla anterior.

Paso 4. Se muestra un mensaje de error, debido a que no hay coincidencias con la búsqueda, existen caracteres no válidos, campos en blanco o bien el usuario ingresado ya está registrado en el sistema.

3.2.7.2. Caso de Uso Gestionar Alarmas

Caso de Uso Consultar Alarmas

Actores: Administrador del sistema, Analista de Monitoreo y NAGIOS.

Descripción: Permite visualizar, editar, documentar y eliminar las fallas generadas por NAGIOS que aun no han sido cerradas.

Pre-condición: El actor del sistema debe ingresar al menú alarmas y seleccionar la opción consultar alarmas entre las diferentes opciones de movimientos posibles.

Flujo de eventos:

Flujo Principal:

1. El actor invoca al caso de uso consultar alarmas.
2. Se consulta la información en la base de datos y despliega por interfaz.

3. El sistema muestra todas las alarmas generadas por NAGIOS hasta ese momento y las diferentes acciones que puede realizar el actor a cada falla.
4. Finaliza el caso de uso.

Flujo Alternativo:

Paso 3. Se muestra un mensaje indicando que no existen alarmas generadas para ese momento.

Caso de Uso Consultar Alarmas Cerradas

Actores: Administrador del sistema, Analista de Monitoreo.

Descripción: Permite visualizar, editar, documentar y eliminar las fallas generadas por NAGIOS que ya han sido cerradas previamente.

Pre-condición: El actor del sistema debe ingresar al menú alarmas y seleccionar la opción consultar alarmas cerradas entre las diferentes opciones de movimientos posibles.

Flujo de eventos:**Flujo Principal:**

1. El actor invoca al caso de uso consultar alarmas cerradas.
2. El sistema muestra las diferentes opciones de búsqueda
3. El actor selecciona la opción que desee.
4. El sistema despliega una interfaz con el resultado de la búsqueda y las diferentes acciones que el actor puede realizar.
5. Finaliza el caso de uso.

Flujo Alternativo:

Paso 4. Se muestra un mensaje de error debido a que no hay coincidencias con la búsqueda.

Caso de Uso Gestionar Configuración

Actores: Administrador del sistema, Analista de Monitoreo.

Descripción: Permite visualizar, ingresar, editar y eliminar las diferentes opciones asignadas a las alarmas como: localidad, disciplina, tipo equipo y tipo elemento.

Pre-condición: El actor del sistema debe ingresar al menú alarmas y seleccionar la opción configuración entre las diferentes opciones de movimientos posibles.

Flujo de eventos:

Flujo Principal:

1. El actor invoca al caso de uso gestionar configuración.
2. El sistema despliega las diferentes opciones a consultar, y las acciones que el actor puede realizar.
3. El actor selecciona de las diferentes opciones, la opción que desea gestionar, bien sea para insertar, modificar o eliminar.
4. Se muestran los datos correspondientes de la operación a realizar.
5. Finaliza el caso de uso.

Flujo Alternativo:

Se cancela la operación y se regresa a la interfaz anterior.

3.2.7.3. Caso de Uso Gestionar Eventos

Caso de Uso Ingresar Eventos

Actores: Administrador del sistema, Analista de Monitoreo.

Descripción: Permite ingresar las fallas que NAGIOS no reporta.

Pre-condición: El actor del sistema debe ingresar al menú eventos y seleccionar la opción ingresar eventos entre las diferentes opciones de movimientos posibles.

Flujo de eventos:**Flujo Principal:**

1. El actor invoca al caso de uso ingresar eventos.
2. El sistema muestra una interfaz con las opciones a seleccionar.
3. Dependiendo de la opción seleccionada por el actor el sistema muestra los campos que deben llenarse.
4. Se registran los datos del nuevo evento.
5. El usuario solicita guardar los datos.
6. Se almacenan permanentemente los datos.
7. Finaliza el caso de uso.

Flujo Alterno:

Paso 5. El usuario Cancela y vuelve a la pantalla anterior.

Paso 6. No se almacenan los datos debido a que existen campos en blanco, o caracteres inválidos.

Caso de Uso Consultar Eventos

Actores: Administrador del sistema, Analista de Monitoreo.

Descripción: Permite visualizar, editar, documentar y eliminar las fallas ingresadas (fallas no reportadas por NAGIOS), que aun no han sido cerradas.

Pre-condición: El actor del sistema debe ingresar al menú eventos y seleccionar la opción consultar eventos entre las diferentes opciones de movimientos posibles.

Flujo de eventos:**Flujo Principal:**

1. El actor invoca al caso de uso consultar eventos.
2. Se consulta la información en la base de datos y despliega por interfaz.
3. El sistema muestra una pantalla con todos los eventos ingresados hasta ese momento y las diferentes acciones que puede realizar el actor a cada falla.

4. Finaliza el caso de uso.

Flujo Alternativo:

Paso 3. Se muestra un mensaje indicando que no existen eventos ingresados para ese momento.

Caso de Uso Consultar Eventos Cerrados

Actores: Administrador del sistema, Analista de Monitoreo.

Descripción: Permite visualizar, editar, documentar y eliminar las fallas ingresadas, que ya han sido cerradas previamente.

Pre-condición: El actor del sistema debe ingresar al menú eventos y seleccionar la opción consultar eventos cerrados entre las diferentes opciones de movimientos posibles.

Flujo de eventos:

Flujo Principal:

1. El actor invoca al caso de uso consultar eventos cerrados.
2. El sistema muestra las opciones de búsqueda.
3. El actor selecciona la opción que desea consultar.
4. De acuerdo a la opción seleccionada el sistema despliega una interfaz con los datos correspondientes a la búsqueda y las diferentes acciones que puede realizar el actor.
5. Finaliza el caso de uso.

Flujo Alternativo:

Paso 4. Se muestra un mensaje de error debido a que no hay coincidencias con la búsqueda.

3.2.7.4. Caso de Uso Gestionar Servicios

Actores: Administrador del sistema, Analista de Monitoreo, Analista de Control de Gestión.

Descripción: Permite ingresar, consultar, modificar o eliminar los servicios utilizados por la gerencia de AIT Servicios Comunes Oriente.

Pre-condición: El actor del sistema debe ingresar al menú servicios y seleccionar la opción a realizar.

Flujo de eventos:

Flujo Principal:

1. El actor invoca al caso de uso gestionar servicios y elige la opción a efectuar.
2. El sistema muestra una pantalla con los campos correspondientes a la opción seleccionada, bien sea de búsqueda, registro, actualización o retiro de algún servicio.
3. El usuario solicita, tomando en cuenta la opción elegida, almacenar los datos, actualizarlos, eliminar un servicio, o mostrar los datos consultados.
4. El sistema guarda los cambios realizados.
5. Finaliza el caso de uso.

Flujo Alterno:

Paso 4. El usuario cancela la operación.

Paso 5. No se almacenan los datos debido a que existen campos en blanco, caracteres no validos, no hay coincidencias con la búsqueda, o bien el servicio ingresado ya se encuentra registrado.

3.2.7.5. Caso de Uso Generar Reportes

Actores: Administrador del sistema, Analista de Monitoreo, Analista de Control de Gestión, Superintendente.

Descripción: Permite generar informes y gráficas de las alarmas y los eventos.

Pre-condición: Acceso al caso de uso generar reportes.

Flujo de eventos:**Flujo Principal:**

1. El actor invoca al caso de uso generar reportes, y selecciona el tipo de reporte que desea generar.
2. El sistema despliega una interfaz con las diferentes opciones de consulta.
3. El actor escoge las especificaciones requeridas.
4. Se consulta la información en la base de datos tomando en cuenta las opciones requeridas por el usuario, y se muestran los datos como interfaz, incluyendo la(s) gráfica(s).
5. El actor selecciona el formato en el cual va a imprimir el reporte.
6. Finaliza el caso de uso.

Flujo Alternativo:

Paso 4. Se muestra un mensaje indicando que la fecha ingresada es incorrecta, o bien que no existen datos para esa fecha.

Paso 5. Se cancela la operación y se vuelve a la interfaz anterior.

3.3. Flujo de Trabajo de Análisis

Durante este flujo se analizan, refinan y estructuran los requerimientos que han sido identificados.

Luego de este proceso descrito se obtiene una primera impresión del modelo de diseño y así se logra estructurar el sistema entero incluyendo su arquitectura, lo que facilita una mejor comprensión, preparación, modificación, y en general, un mayor alcance de estos requerimientos.

3.3.1. Especificación de los grupos de usuarios formalmente descritos

En esta sub-etapa se identifica los usuarios que intervienen en cada uno de los procesos que forman los distintos casos de uso. La especificación en grupos de usuarios determina su ubicación dentro de la aplicación, permitiendo saber su nivel de acceso e importancia dentro de la misma, esto ayuda a disminuir los riesgos del sistema, así como también detalla que datos deben manejar cada uno de ellos y en que procesos están involucrados. La especificación de grupos formalmente descritos puede observarse en la figura 3.3.

3.3.2. Diagrama de Clases de Análisis

Una vez identificado los casos de uso del sistema y luego de puntualizar cada uno de ellos, se procedió a generar el Diagrama de Clases de Análisis del mismo; para ello, se analizó cada caso de uso representando al sistema como una estructura de clasificadores que revela como debería ser diseñada la estructura del sistema, mediante una abstracción de una o varias clases básicas, las cuales son de interfaz, de control o de entidad.

Las clases de análisis siempre encajan en uno de tres estereotipos básicos: De interfaz, para modelar la interacción entre el sistema y sus actores, de control, para representar coordinación, secuenciamiento, transacciones y control entre objetos o de entidad, para modelar información de larga vida o permanente.

Cada estereotipo implica una semántica específica, lo cual constituye un método potente y consistente de identificar y describir las clases de análisis.

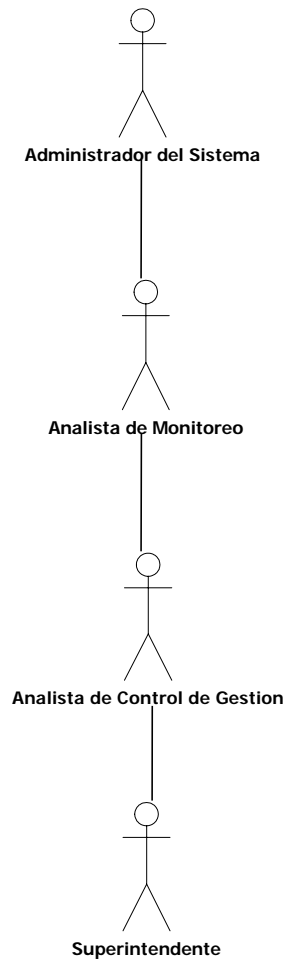


Figura 3.3. Especificación de Grupos formalmente descritos.

Fuente: Elaboración Propia.

En la tabla 3.5 se puede observar la identificación de cada una de las clases de interfaz con su respectiva descripción.

Tabla 3.5. Identificación de las Clases Interfaz (1/3).

Clases Interfaz	Descripción
IU Listar Usuarios	Permite a los actores involucrados visualizar los datos asociados a los usuarios registrados en el sistema.
IU Consultar Usuario	Permite consultar algún usuario registrado en el sistema.
IU Ingresar Usuario	Permite ingresar usuarios al sistema.
IU Modificar Usuario	Permite actualizar los datos de los usuarios registrados.
IU Eliminar Usuario	Permite eliminar algún usuario del sistema.
IU Mostrar Alarmas	Permite visualizar las alarmas generadas por NAGIOS.
IU Alarmas Pendientes	Permite visualizar las alarmas generadas por NAGIOS que tengan estado pendiente.
IU Editar Alarma	Permite al usuario modificar los datos de las alarmas.
IU Eliminar Alarma	Permite eliminar las alarmas que aun no se han documentado.
IU Documentar Alarma	Permite realizar la documentación de las alarmas generadas.
IU Buscar Alarma	Permite consultar las alarmas cerradas, ingresando número de falla.
IU Búsqueda Avanzada Alarma	Permite consultar las últimas 10 alarmas cerradas.
IU Eliminar Alarma Cerrada	Permite eliminar las alarmas cerradas.
IU Documentar Alarma Cerrada	Permite documentar las alarmas cerradas.
IU Ingresar Opciones	Permite al usuario seleccionar las opciones de configuración de las alarmas, y las acciones que desee realizar.

Fuente: Elaboración Propia.

Tabla 3.5. Identificación de las Clases Interfaz (2/3).

Clases Interfaz	Descripción
IU Editar	Permite modificar los datos de alguna opción dependiendo de la elección que realice el usuario.
IU Eliminar	Permite eliminar un registro de alguna opción de configuración del sistema.
IU Ingresar	Permite ingresar las diferentes opciones de configuración al sistema.
IU Ingresar Eventos	Permite seleccionar un tipo de evento.
IU Evento Manual	Permite ingresar un evento de tipo manual al sistema.
IU Evento Adicional	Permite ingresar un evento de tipo adicional al sistema.
IU Consultar Eventos	Permite consultar los diferentes eventos ingresados por los usuarios.
IU Eventos Pendientes	Permite visualizar los eventos que se encuentren en estado pendiente.
IU Editar Evento	Permite modificar los datos de un evento.
IU Eliminar Evento	Permite eliminar los eventos que aun no se han documentado.
IU Documentar Evento	Permite realizar la documentación de los eventos que no se han cerrado.
IU Buscar Evento	Permite consultar los eventos cerrados ingresando el número de caso del mismo.
IU Búsqueda Avanzada Evento	Permite visualizar los últimos 10 eventos cerrados hasta ese momento.
IU Eliminar Evento Cerrado	Permite eliminar los eventos cerrados.
IU Documentar Evento Cerrado	Permite documentar los eventos cerrados.

Fuente: Elaboración Propia.

Tabla 3.5. Identificación de las Clases Interfaz (3/3).

Clases Interfaz	Descripción
IU Ingresar Servicio	Permite ingresar los diferentes servicios que maneja la gerencia.
IU Buscar Servicio	Permite consultar los servicios registrados en el sistema.
IU Modificar Servicio	Permite actualizar los datos de algun servicio.
IU Eliminar Servicio	Permite eliminar del sistema algun servicio
IU Reporte Estandar	Permite al usuario generar reportes de todas las alarmas y los eventos generados por estado.
IU Reporte por Impacto	Permite al usuario generar reportes y gráficas de todas las alarmas y los eventos por impacto.
IU Reporte por Mantenimiento	Permite al usuario generar reportes y gráficas de todas las alarmas y los eventos por mantenimiento.
IU Reporte por Equipo	Permite al usuario generar reportes y gráficas de todas las alarmas y los eventos por equipo.
IU Reporte de Busqueda Rapida	Permite al usuario generar reportes de todas las alarmas y los eventos por IP y por equipo.

Fuente: Elaboración Propia.

La tabla 3.6 muestra la identificación de cada una de las clases de control y su descripción.

Tabla 3.6. Identificación de las Clases de Control.

Clases de Control	Descripción
Gestor Receptora Usuario	Se encarga de validar los datos ingresados, y realizar las operaciones necesarias, para enviar los datos correctos.
Gestor Procesar Alarmas	Se encarga de realizar la búsqueda de las alarmas nuevas, en progreso y cerradas.
Gestor Procesar Pendientes	Se encarga de realizar la búsqueda de las alarmas pendientes.
Gestor Receptora Alarmas	Se encarga de procesar los datos que seran enviados para ser almacenados en la base de datos.
Gestor Receptora Alarmas Cerradas	Se encarga de validar toda la informacion correspondiente a las alarmas cerradas.
Gestor Receptora Opciones	Se encarga de la gestión de consulta y de los procesos de edición y borrado de las diferentes categorías de configuración.
Gestor Procesar Eventos	Se encarga de desplegar las opciones, entre las cuales el usuario debe seleccionar.
Gestor Receptora Eventos	Se encarga de las operaciones de validación de los datos.
Gestor Procesar Eventos Pendientes	Se encargaga de los procesos de búsqueda y selección de los eventos pendientes.
Gestor Receptora Eventos Cerrados	Se encarga de administrar las diferentes opciones de consultar, modificar, eliminar y documentar.
Gestor Receptora Servicio	Se encarga de administrar las diferentes opciones de registrar nuevos datos, consultar, actualizar o retirar algun servicio.
Gestor Receptora Reportes	Se encarga del procesamiento y validación de los datos para reallizar las consultas de los reportes.

Fuente: Elaboración Propia

En la tabla 3.7 se puede apreciar la identificación de cada una de las clases entidad y la descripción de cada una.

Tabla 3.7. Identificación de las Clases Entidad.

Clases Entidad	Descripción
Usuario	Almacena todos los datos correspondientes a los usuarios ingresados al sistema.
Estado Evento	Almacena las diferentes opciones de estado para las alarmas.
Evento	Almacena todos los eventos ingresados y las alarmas generadas por NAGIOS.
Documentación	Registra todas las documentaciones realizadas a las alarmas y los eventos.
Localidad	Registra las diferentes localidades pertenecientes a servicios comunes oriente.
Disciplina	Registra los diferentes grupos solucionadores pertenecientes a servicios comunes oriente.
Tipo Equipo	Registra las diferentes categorías en la que se dividen los equipos de la gerencia.
Tipo Elemento	Registra las diferentes categorías en la que se dividen los elementos de los equipos.
Servicio	Registra los servicios ingresados.
Servicio Elemento	Registra los elementos asociados a los servicios.
Impacto	Registra los tipos de impactos que maneja la gerencia.
Mantenimiento	Registra los tipos de mantenimiento que maneja la gerencia.
Equipo	Registra los equipos pertenecientes a servicios comunes oriente.

Fuente: Elaboración Propia.

En este capítulo se presentarán los diagramas de clase de análisis para cada uno de los casos de uso del sistema en su totalidad, los diagramas de colaboración se mostraran en el siguiente capítulo.

3.3.2.1. Descripción del Diagrama de Clase de Análisis para el caso de uso Administrar Usuarios

A continuación en la figura 3.4 se muestra el diagrama del caso de uso Administrar Usuarios. Para realizar el análisis del caso de uso Administrar Usuarios se identificaron en primer lugar las diferentes clases de análisis como lo son: IU Ingresar Usuario, la cual permite rellenar un formato para ingresar un usuario al sistema, IU Consultar Usuario que permite buscar los datos de cualquier usuario registrado en el sistema, IU Modificar Usuario, la cual permite editar los datos de algún usuario, IU Eliminar Usuario, que permite eliminar del sistema algún usuario o cambiar el estatus del mismo en caso de este haber documentado alguna falla y IU Listar Usuarios, que muestra todos los usuarios almacenados en el sistema. Cada una de estas interfaces se conectan a la clase de control Gestor Receptora Usuario, la cual se encarga de manera general de validar los datos ingresados y realizar las operaciones necesarias para enviar los datos correctos a la clase entidad usuario encargada de realizar las consultas necesarias para almacenar los datos ingresados, buscar los datos solicitados, y eliminar datos. El actor identificado como administrador del sistema, representa al usuario que interactúa con este caso de uso.

3.3.2.2. Descripción del Diagrama de Clase de Análisis para el caso de uso Gestionar Alarmas

En el caso de uso Gestionar Alarmas se presenta a los actores con los privilegios correspondientes. Para este caso se pone a la disposición del actor un

menú con las opciones: Consultar Alarmas, Consultar Alarmas Cerradas y Configuración como se muestra en la figura 3.5.

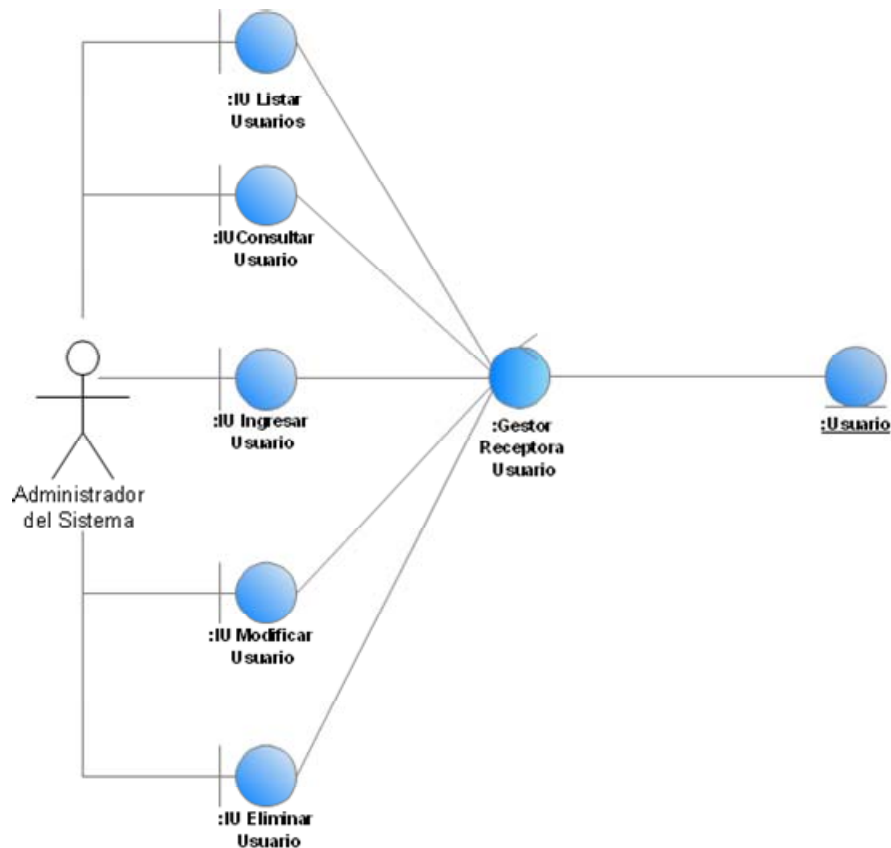


Figura 3.4. Diagrama de Clase de Análisis para el caso de uso Administrar Usuarios.

Fuente: Elaboración Propia.

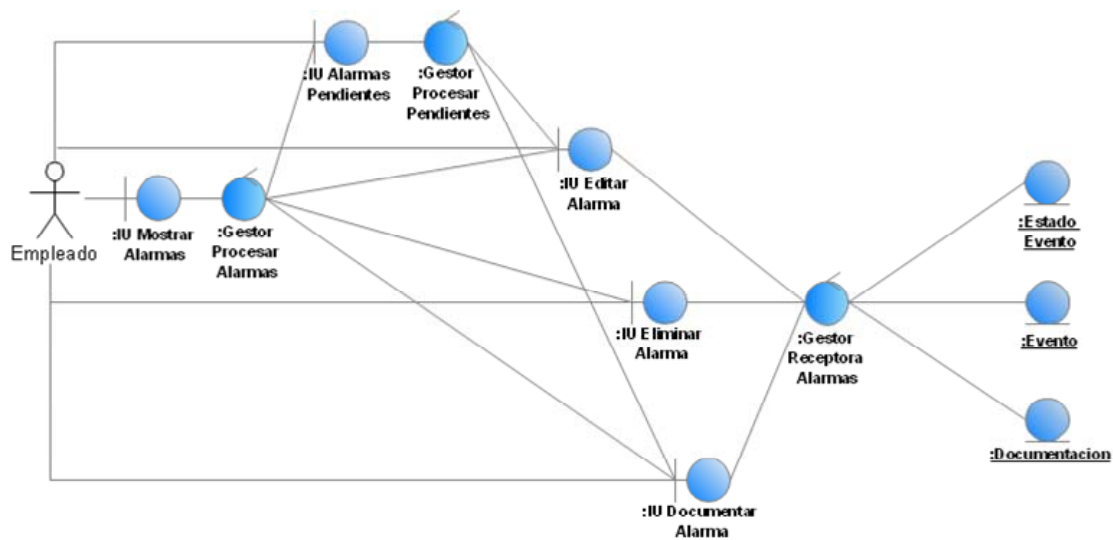
Para realizar el análisis del caso de uso Consultar Alarmas se identificaron en primer lugar las clases de análisis. IU Mostrar Alarmas, que es la interfaz principal del caso de uso, la cual muestran todas las alarmas generadas hasta ese momento y permitirá desplazarse para realizar las distintas acciones que contempla el caso de uso, IU Editar Alarma, donde se encuentra un formulario a rellenar con todos los datos correspondientes a la alarma seleccionada, lo cual permite modificar o cambiar algún dato, IU Eliminar Alarma, donde se selecciona una alarma para eliminarla, IU

Documentar Alarma, que permitirá ingresar toda la documentación referente a la falla, así como seleccionar ciertos atributos y consultar alarmas documentadas relacionadas con el equipo que ha originado la falla, IU Alarmas Pendientes que permitirá visualizar todas las alarmas que tengan asignado un estado pendiente, esta interfaz permitirá al actor desplazarse por las distintas acciones como son: IU Editar Alarma, donde se encuentra un formulario a rellenar con todos los datos correspondientes a la alarma pendiente, lo cual permitirá modificar o cambiar algún dato, IU Documentar Alarma, que permitirá seguir agregando documentación referente a la falla, así como seleccionar ciertos atributos y consultar alarmas documentadas relacionadas con el equipo que ha originado la falla. Se denotan las clases de control Gestor Procesar Alarmas y Gestor Procesar Pendientes, que se encargan de realizar la búsqueda de las alarmas correspondientes y mostrarlas al usuario con sus respectivas opciones a seleccionar. La clase de control Gestor Receptora Alarmas, se encarga de manera general del procesamiento y validación de los datos. Las clases entidades que corresponden a este caso de uso se denominan, Evento, Estado Evento y Documentación, encargadas de realizar las operaciones necesarias para almacenar, consultar, actualizar y eliminar los datos asociados a las alarmas en la base de datos.

El actor identificado como Empleado, representa a todos los actores que interactúan con este caso de uso, en este caso particular representa a los actores Administrador del Sistema y al Analista de Monitoreo.

En la figura 3.6 se visualiza el diagrama de clases de análisis para consultar alarmas cerradas, que cuenta con las clases interfaz IU Buscar Alarma, e IU Búsqueda Avanzada Alarma, estas interfaces interactuarán con la clase de control Gestor Procesar Alarmas que permitirá desplegar las interfaces referentes a este subproceso, IU Editar Alarma, IU Eliminar Alarma Cerrada e IU Documentar Alarma

Cerrada, mediante las cuales el actor podrá consultar, modificar, documentar y cambiar el estado de las alarmas que el usuario desee descartar.



**Figura 3.5. Diagrama de Clase de Análisis para el caso de uso Gestionar Alarmas:
Consultar Alarmas.**

Fuente: Elaboración Propia.

La clase de control Gestor Receptora Alarmas Cerradas es la encargada de realizar todas las operaciones concernientes a la validación y procesamiento de los datos. Este diagrama cuenta con las entidades Evento, Estado Evento y Documentación las cuales se encargan de las operaciones de consulta, actualización, y almacenamiento de todos los datos necesarios, asociados a las diferentes operaciones.

En la figura 3.7 se puede apreciar el diagrama de clase de análisis para el caso de uso gestionar configuración, en el que se identificaron en primer lugar las clases de interfaz: IU Ingresar Opciones, que es la interfaz principal del caso de uso y brindará la oportunidad de desplazarse por las diferentes opciones que éste contempla,

Ingresar, Editar y Eliminar en las cuales se interactuará con el usuario para realizar las respectivas acciones.

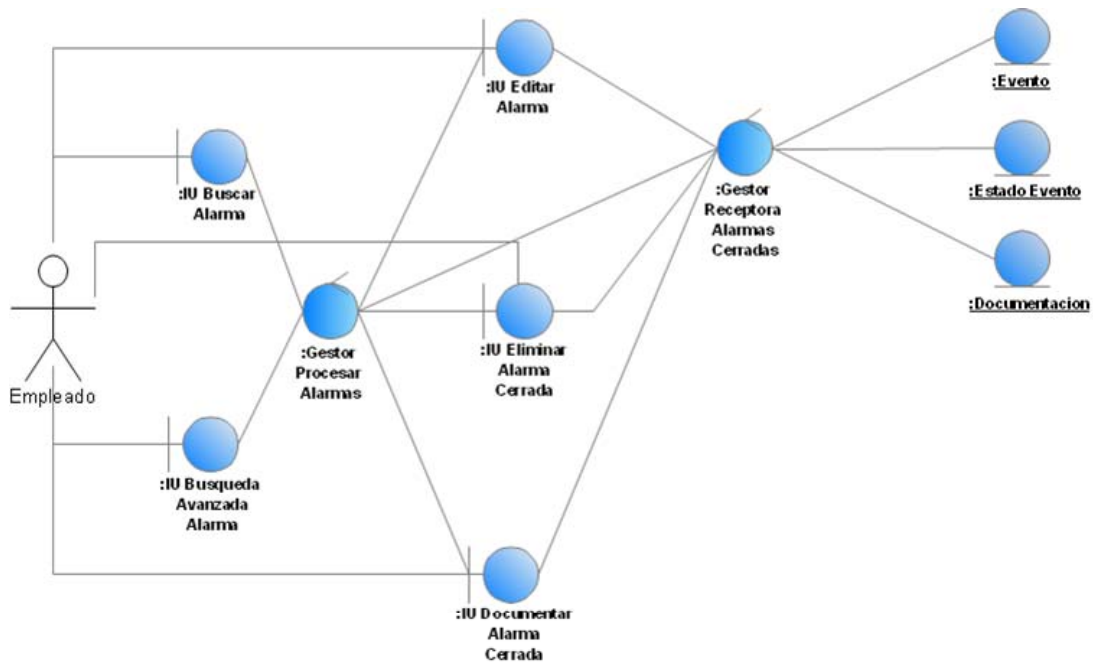


Figura 3.6. Diagrama de Clase de Análisis para el caso de uso Gestionar Alarmas: Consultar Alarmas Cerradas.

Fuente: Elaboración Propia.

Se denotan también en este diagrama las clases de control Gestor Procesar Alarmas, encargada de procesar la selección realizada por el usuario, Gestor Receptora Opciones, que permiten la gestión de consulta y de los procesos de edición y borrado de las diferentes categorías. Las clases entidad Localidad, Disciplina, Tipo Equipo y Tipo Elemento, sirven para almacenar la información de las opciones a gestionar en la base de datos.

3.3.2.3. Descripción del Diagrama de Clase de Análisis para el caso de uso Gestionar Eventos

En el caso de uso Gestionar Eventos se presenta a los actores con los privilegios correspondientes. Para este caso se pone a la disposición del actor un menú con las opciones: Ingresar Eventos, Consultar Eventos y Consultar Eventos Cerrados. La figura 3.8 muestra el diagrama de analisis para el caso de uso ingresar eventos.

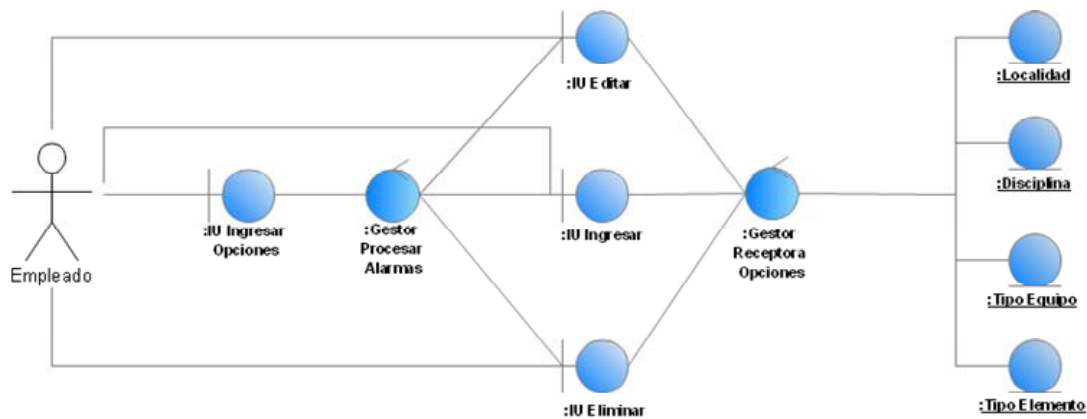


Figura 3.7. Diagrama de Clase de Análisis para el caso de uso Gestionar Alarmas: Gestionar Configuración.

Fuente: Elaboración Propia.

Para realizar el análisis del caso de uso Ingresar Eventos se identificaron en primer lugar las clases de análisis: IU Ingresar Eventos, que es la interfaz principal del caso de uso, permitirá al actor seleccionar entre diferentes opciones, la acción que desee ejecutar. IU Evento Manual, donde se encuentra un formulario a rellenar con todos los datos correspondientes para ingresar un evento de tipo manual, IU Evento Adicional, que permite ingresar todos los datos correspondientes a un evento adicional mediante un formulario. Se denotan las clases de control Gestor Procesar Eventos, que se encarga de desplegar las opciones entre las cuales el usuario debe

seleccionar, Gestor Receptora Eventos, encargada de las operaciones de validación. Las clases entidad identificadas son Evento, Estado Evento, Mantenimiento y Documentación encargada de la gestión de almacenamiento de los eventos a ingresar.

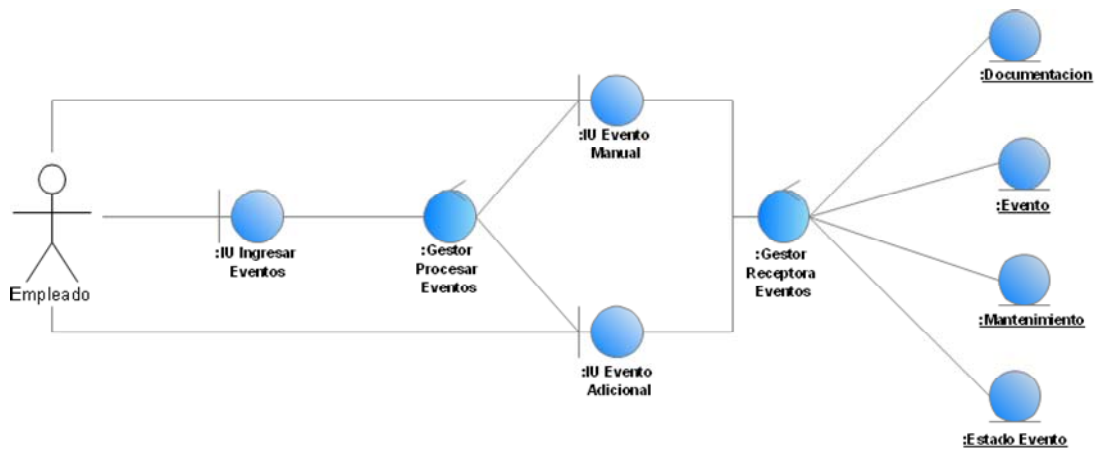
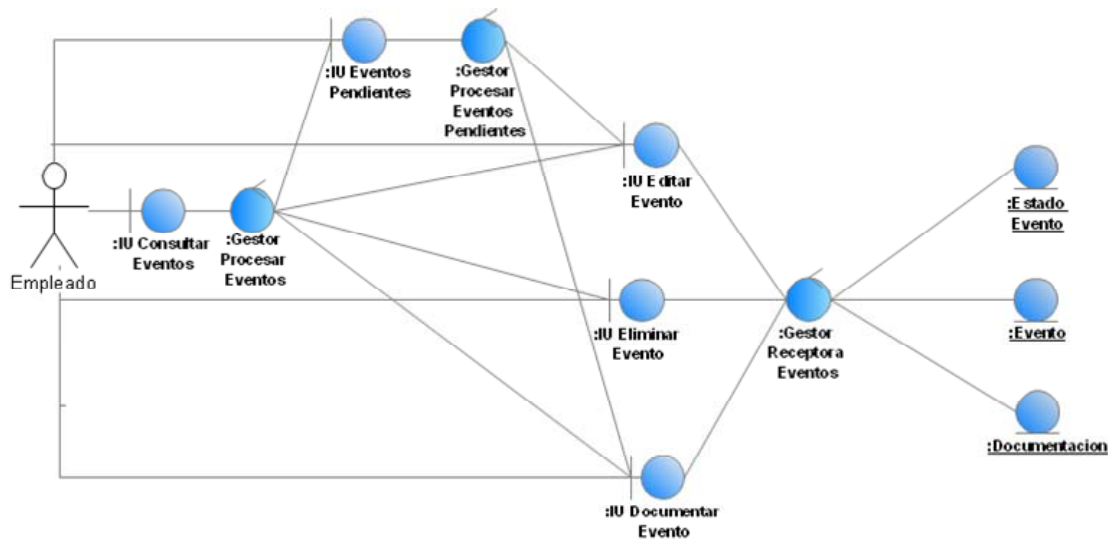


Figura 3.8. Diagrama de Clase de Análisis para el caso de uso Gestionar Eventos: Ingresar Eventos.

Fuente: Elaboración Propia

En la figura 3.9 se puede apreciar el diagrama de clase de análisis para el caso de uso Consultar Eventos, en el que se identificaron en primer lugar las clases de interfaz: IU Consultar Eventos, que es la interfaz principal del caso de uso y brindará la oportunidad de desplazarse por las diferentes opciones que éste contempla, IU Editar Evento, IU Eliminar Evento, IU Documentar Evento e IU Eventos Pendientes, en las cuales se interactuará con el usuario para realizar las respectivas acciones. Se denotan también en este diagrama las clases de control Gestor Procesar Eventos, que se encarga de manera general de los procesos relacionados con la búsqueda de los datos de todos los eventos ingresados. Gestor Procesar Eventos Pendientes encargada de los procesos de búsqueda y selección de los eventos pendientes. Gestor Receptora Eventos, encargada de los procesos de validación de los datos. Las clases entidad

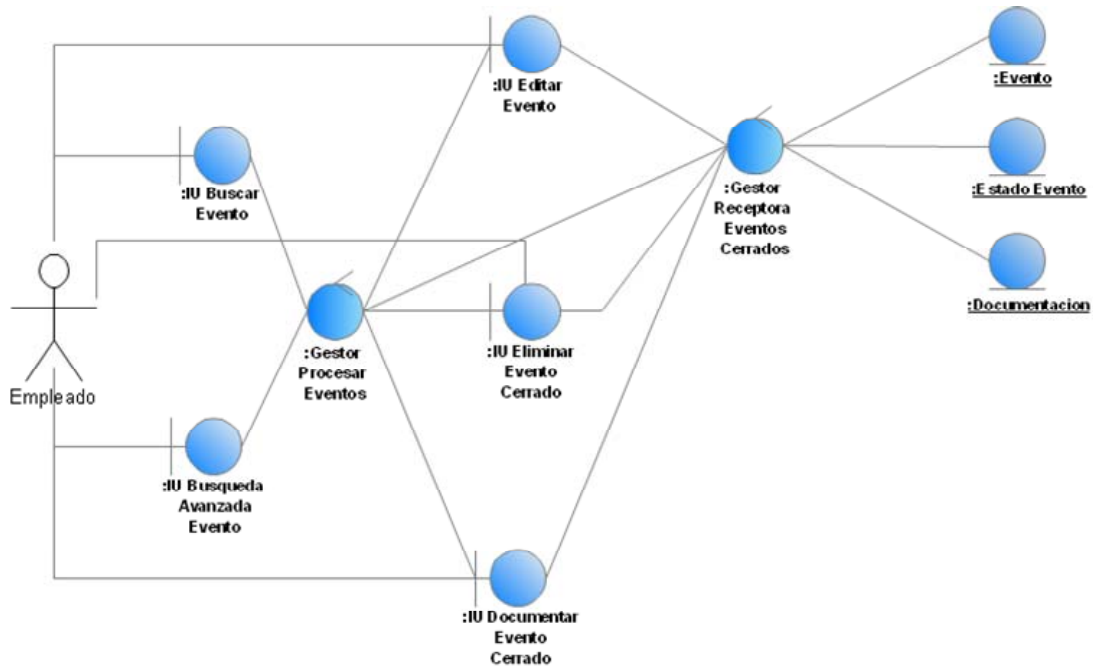
Evento, Estado Evento y Documentación, sirven para almacenar toda la información relacionada con los eventos, su documentación y estados en la base de datos.



**Figura 3.9. Diagrama de Clase de Análisis para el caso de uso Gestionar Eventos:
Consultar Eventos.**

Fuente: Elaboración Propia.

A continuación en la figura 3.10 se muestra el diagrama de clase de análisis para el caso de uso Consultar Eventos Cerrados. Entre las clases participantes se destacan la clase de interfaz denominada IU Buscar Evento e IU Búsqueda Avanzada Evento, que permiten al actor desplazarse por las distintas opciones que le brinda el caso de uso como son IU Editar Evento, IU Eliminar Evento Cerrado e IU Documentar Evento Cerrado. Se denotan las clases de control Gestor Procesar Eventos y Gestor Receptora Eventos Cerrados que permiten administrar las diferentes opciones de consultar, modificar, eliminar y documentar. Por último las clases de entidad Evento, Estado Evento y Documentación, que brindan la funcionalidad para el almacenamiento y consulta de datos.



**Figura 3.10. Diagrama de Clase de Análisis para el caso de uso Gestionar Eventos:
Consultar Eventos Cerrados.**

Fuente: Elaboración Propia.

3.3.2.4. Descripción del Diagrama de Clase de Análisis para el caso de uso Gestionar Servicios

A continuación en la figura 3.11 se muestra el diagrama de análisis para el caso de uso gestionar servicios. Entre las clases participantes al realizar el caso de uso Gestionar Servicios se destacan la clase de interfaz denominada IU Ingresar Servicio, que mediante un formulario permite ingresar un servicio y los elementos asociados a el mismo. IU Buscar Servicio, que muestra los datos que conforman un servicio en particular, IU Modificar Servicio, permite realizar algún cambio en la información mostrada y almacenar dichos cambios, y IU Eliminar Servicio que permite eliminar algún servicio del sistema. Se denota la clase de control, Gestor Receptora Servicio, que permite administrar las diferentes opciones de registrar nuevos datos, consultar,

actualizar o retirar algún servicio, la clase entidad Servicio se encarga de realizar las consultas necesarias que permiten obtener y almacenar los datos del sistema.

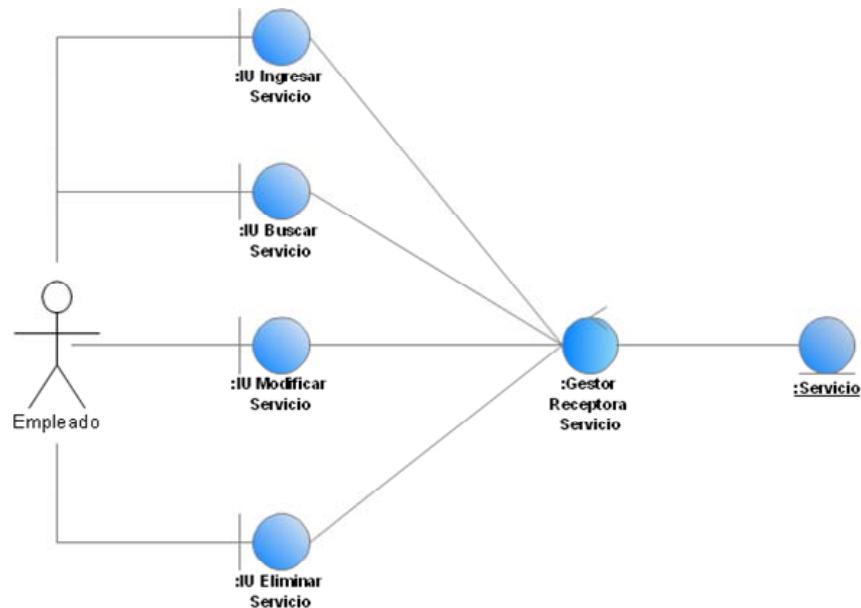


Figura 3.11. Diagrama de Clase de Análisis para el caso de uso Gestionar Servicios.

Fuente: Elaboración Propia.

3.3.2.5. Descripción del Diagrama de Clase de Análisis para el caso de uso Generar Reportes

En la figura 3.12 se detalla el diagrama de análisis para el caso de uso generar reportes. Para realizar el análisis del caso uso Generar Reportes se identificaron las clases de análisis IU Reporte Estándar, IU Reporte por Impacto, IU Reporte por Mantenimiento, IU Reporte por Equipo, IU Reporte de Búsqueda Rápida. En lo que respecta a las clases de control se cuenta con un Gestor Receptora Reportes, que funciona como procesador principal. Las clases entidad identificadas son Evento, Estado Evento, Mantenimiento, Equipo y Impacto encargadas de realizar las consultas necesarias que permiten el acceso a la información de la base de datos.

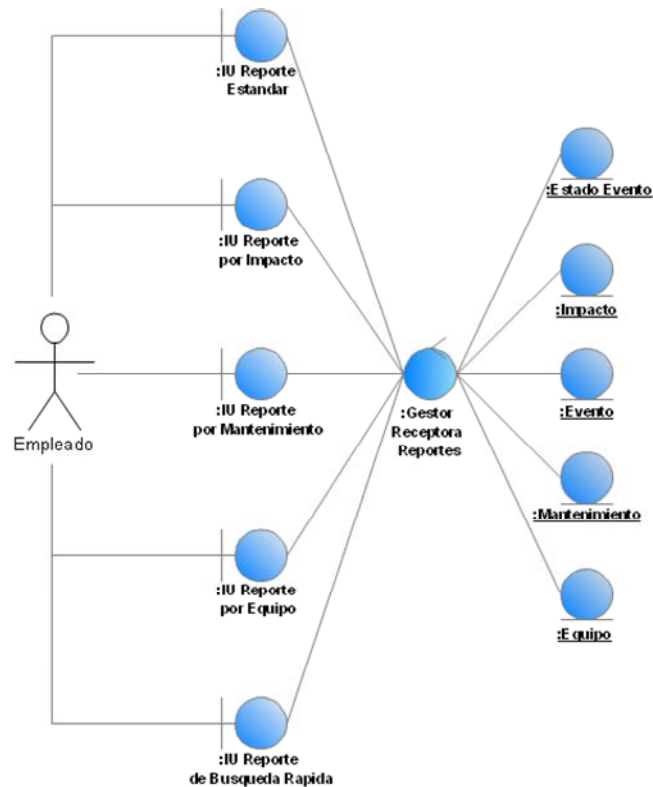


Figura 3.12. Diagrama de Clase de Análisis para el caso de uso Generar Reportes.

Fuente: Elaboración Propia.

3.4. Flujo de Trabajo Diseño

En el diseño se modela el sistema y se encuentra su forma (incluida la arquitectura) para que soporte todos los requisitos. Como esta es la fase de inicio, para este flujo de trabajo sólo se presentará un bosquejo de las interfaces principales que debería presentar el sistema.

3.4.1. Paquete de Análisis del sistema

La identificación de los paquetes de análisis del sistema permite ordenar el modelo de análisis a través de un entorno que permita visualizar el modelo en piezas más pequeñas y manejables. Se basa principalmente en los requerimientos del sistema que se han considerado, capturando estos requerimientos en la forma de los casos de

uso que dan soporte al funcionamiento del sistema mediante los procesos que son desarrollados en el mismo. El diagrama de paquetes de análisis del sistema puede observarse en la figura 3.13.

3.4.2. Prototipo Interfaz candidata

El prototipo de interfaz candidata, mostrada en la figura 3.14, es la primera interfaz que se muestra al usuario en el momento que accede al sistema.

Se puede observar un área asignada para el banner que tendrá el logo de la empresa, un area para mostrar la identificación del usuario que esta ingresando en ese momento, un área de opciones, un área para mostrar el logo de la aplicación, que en este caso es el Sistema Integrado de Notificación y Comunicación de Alarmas (SIDENCA) y un área ubicada en la parte inferior de la pantalla que muestra los derechos del autor.

3.5. Evaluación de la Fase de Inicio

Durante el desarrollo de esta fase se crearon modelos como los casos de uso y el modelo de dominio, que definieron de una manera clara lo que va a estar dentro o fuera del mismo. Esto se puede observar claramente en el modelo del dominio, cuya finalidad es precisamente aclarar el contexto del sistema. Se identificaron los actores principales, y en el modelo de análisis se explicaron las interfaces de éstos con el sistema, así como las tareas que cumplirían. En resumen, se cumplieron con los criterios relativos al ámbito del sistema, lo que nos ayudará a la base de nuestra arquitectura en el capítulo siguiente de fase de elaboración.

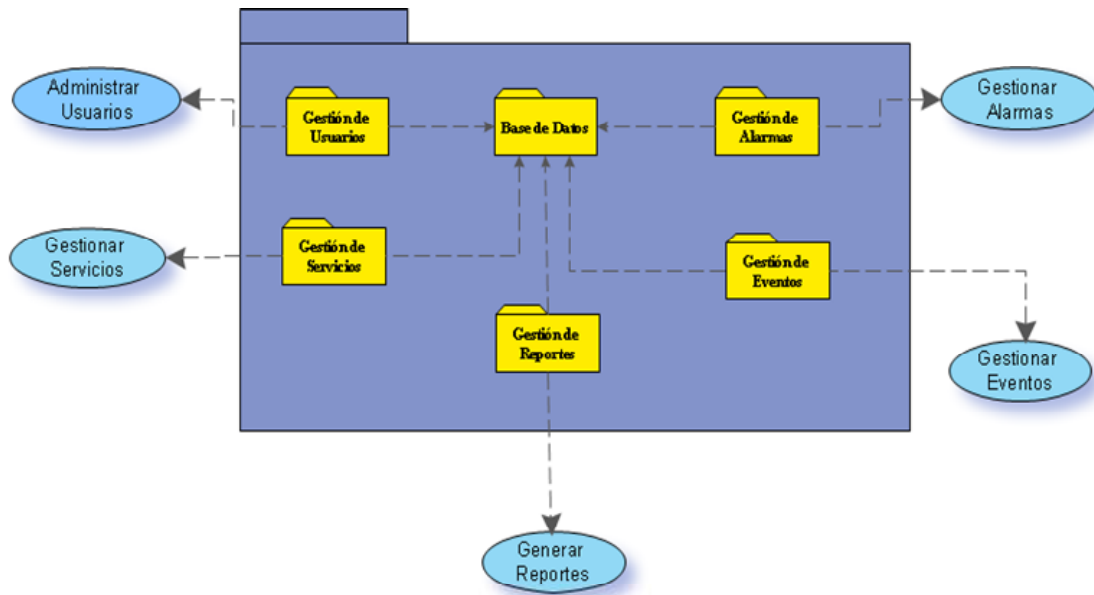


Figura 3.13. Diagrama de Paquete de Analisis del Sistema.

Fuente: Elaboración Propia.



Figura 3.14. Prototipo de Interfaz candidata.

Fuente: Elaboración Propia

CAPÍTULO IV

FASE DE ELABORACIÓN

4.1. Introducción

La fase de elaboración construye la línea base de la arquitectura del sistema. Los principales objetivos para esta fase:

- Recopilar la mayor parte de los requisitos que aún queden pendientes, formulando los requisitos funcionales como casos de uso.
- Establecer una base de la arquitectura sólida -la línea base de la arquitectura- para guiar el trabajo durante las fases de construcción y transición, así como en las posteriores generaciones del sistema.
- Continuar la observación y control de los riesgos críticos que aún queden, e identificar riesgos significativos hasta el punto de que se pueda estimar su impacto en el análisis negocio, y en particular en la apuesta económica.
- Completar los detalles del plan del proyecto.
- Para esta fase se debe desarrollar alrededor del 80 por ciento de los casos de uso y abordar los riesgos que interfieran en la consecución de este objetivo.
- Esta fase se fundamenta en el flujo de trabajo que se refiere al análisis, debido a que no se identificaron nuevos actores, casos de uso, ni requisitos que se pudiesen agregar a los establecidos en la fase de inicio.

4.2. Flujo de Trabajo de Análisis

En la fase de inicio se desarrolló el análisis de la arquitectura sólo hasta el punto de determinar que había una arquitectura del sistema factible. Ahora en esta fase se extiende el análisis hasta el punto de que pueda servir como base a la línea principal de la arquitectura ejecutable.

En la fase de análisis, se transforma el modelo de caso de usos en un modelo de análisis. El modelo de análisis de los requisitos permite lograr un método más concreto sobre los aspectos del sistema y una base más detallada de dichas exigencias obtenidas en el modelo de casos de uso para su mejor entendimiento; este diagrama de análisis modela la estructura del sistema basada en clases de análisis. La realización de casos de usos en este proyecto se describirá a través de diagramas de colaboración.

4.2.1. Diagramas de Colaboración

Los diagramas de clases de análisis permiten la apreciación de la estructura interna del mismo, la utilización de los diagramas de uso permitirán obtener la interacción con cada una de las entidades, gestores e interfaces del sistema.

En los diagramas de colaboración, la idea es ver los objetos en extenso, donde las interacciones muestran los flujos y los mensajes entre ellos, estos diagramas tienen como objetivo fundamental identificar los requisitos y responsabilidades sobre los objetos, mostrando las interacciones entre ellos creando enlaces entre objetos y añadiendo a esos enlaces.

4.2.1.1. Descripción del Diagrama de Colaboración para el caso de uso Administrar Usuarios

El Actor activa la interfaz ingresar usuario como se muestra en la figura 4.1, donde suministra los datos relacionados con el nuevo usuario (1) que serán validados por el Gestor Receptora Usuarios (2) y enviados a la entidad usuario que se encargará de almacenar los datos del nuevo usuario (3) y devolverá un mensaje con el estado del almacenamiento de los datos (4) el cual se enviara como respuesta a la interfaz para ser mostrado como parte del estado del registro al usuario (5).

Si el actor selecciona la opción consultar usuario se mostrará la interfaz consultar usuario donde el actor debe ingresar el atributo de la consulta (6) que será validado por el gestor receptora usuarios (7) y enviados a la clase entidad usuario encargado de solicitar los datos del usuario en la base de datos del sistema (8). Una vez obtenidos los datos necesarios el gestor receptora usuario será el encargado de transferir el resultado de la consulta a la interfaz correspondiente (9) que a su vez se encargará de mostrarlos al actor (10).

Si el administrador desea modificar los datos del usuario se desplegará la interfaz de modificar usuario donde se mostrarán los datos del usuario permitiendo modificarlos (11), una vez modificados se pasan los datos al gestor receptora usuario que validará los datos modificados (12) y los transferirá a la clase entidad usuario (13) para que ésta se encargue de almacenarlos, y a su vez enviara un mensaje de respuesta satisfactoria (14) el cual será enviado a la interfaz correspondiente y mostrado al actor (15).

En caso de desear eliminar algún usuario se seleccionará la opción correspondiente y se desplegará la interfaz eliminar usuario donde el actor ingresará el atributo del usuario a retirar (16) el cual será enviado al gestor receptora usuario

(17) que transmitirá la información a la entidad usuario (18) y cambiará el tipo de validación del usuario correspondiente y a su vez generará un mensaje de respuesta satisfactoria (19), el cual será enviado a la interfaz para ser mostrado al actor (20).

Si el administrador selecciona la opción de listar usuarios se mostrará la interfaz (21), el gestor receptora usuario (22) se encarga de enviar la solicitud a la entidad usuario (23) la cual se encarga de solicitar los datos de todos los usuarios registrados en el sistema. Una vez obtenidos los datos necesarios la clase entidad usuario será la encargada de transferir los resultados de la consulta al gestor receptora usuario (24) que a su vez se encargará de mostrarlos por la interfaz (25).

La tabla 4.1 muestra la leyenda para el diagrama de colaboración de la figura 4.1.

4.2.1.2. Descripción del Diagrama de Colaboración para el caso de uso Gestionar Alarmas: Consultar Alarmas.

El diagrama de colaboración para el caso de uso consultar alarmas se puede observar en la figura 4.2. El actor identificado como empleado representa a todos los actores que interactúan con este caso de uso, es decir al administrador del sistema, y el analista de monitoreo. Este caso de uso comienza cuando el actor selecciona la interfaz acorde al tipo de acción que va a realizar en la interfaz principal de mostrar alarmas (1), la cual permite visualizar todas las alarmas generadas por NAGIOS, el gestor procesar alarmas se encarga de procesar su elección (2) de acuerdo a la alarma seleccionada, en caso de seleccionar la opción editar, se activa la interfaz editar alarma (3), en la cual se desplegaran todos los datos correspondiente a la alarma seleccionada, y el usuario podrá ingresar los datos pertinentes (4), que serán validados por el gestor receptora alarmas (5) y transferidos a la clase entidad evento la cual se encargará de realizar las operaciones necesarias para actualizar los datos en

la base de datos del sistema (6), luego retorna el estado del registro (7), el cual será enviado como respuesta a la interfaz y se muestra un estado de conclusión satisfactoria o un error dependiendo del caso (8).

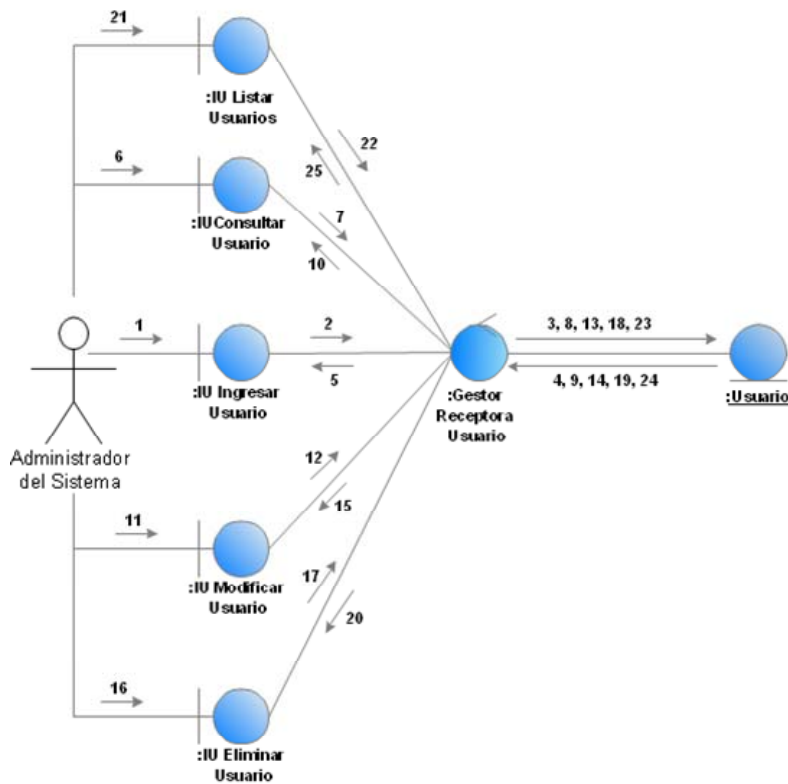


Figura 4.1. Diagrama de Colaboración para el caso de uso Administrar Usuarios.

Fuente: Elaboración Propia.

Tabla 4.1. Leyenda para el Diagrama de Colaboración para el caso de uso Administrar Usuarios.

1→ Activar interfaz ingresar	2→ Transferir datos de nuevo registro	3→ Almacenar datos usuario
4→ Transferir estado de almacenamiento	5→ Mostrar estado de registro al usuario	6→ Solicitar interfaz consulta
7→ Transferir datos consulta	8→ Solicitar datos usuario	9→ Transferir datos consulta
10→ Desplegar resultados consulta	11→ Activar interfaz modificar	12→ Transferir datos actualizados
13→ Gestionar datos actualizados	14→ Transferir Estado de almacenamiento	15→ Mostrar estado de registro al usuario
16→ → Activar interfaz eliminar	17→ Transferir datos de usuario a eliminar	18→ Retiro de usuario
19→ Devolver estado de retiro	20→ Recibo de mensaje	21→ Seleccionar listar
22→ Enviar solicitud	23→ Transferir consulta	24→ Enviar datos consultados
25→ Mostrar datos consultados		

Fuente: Elaboración Propia.

En caso de seleccionar la opción eliminar, se activa la interfaz para eliminar (9), y se selecciona el factor a eliminar (10) que será enviado al gestor receptora alarmas (11), en el cual serán validados para verificar si es posible o no cambiar el estado de la alarma. Luego los datos son transferidos a la entidad estado evento (12) que se encargará de cambiar el estado de la alarma, para luego retornar al gestor receptora alarmas (13) el estado de la operación y mostrarlo al usuario (14).

Si la opción seleccionada corresponde a documentar se activará la interfaz documentar alarma (15), donde el empleado ingresará los atributos correspondientes a la documentación de la falla (16), estos datos serán enviados al gestor receptora alarmas (17), donde serán validados y transferidos a la entidad documentacion que

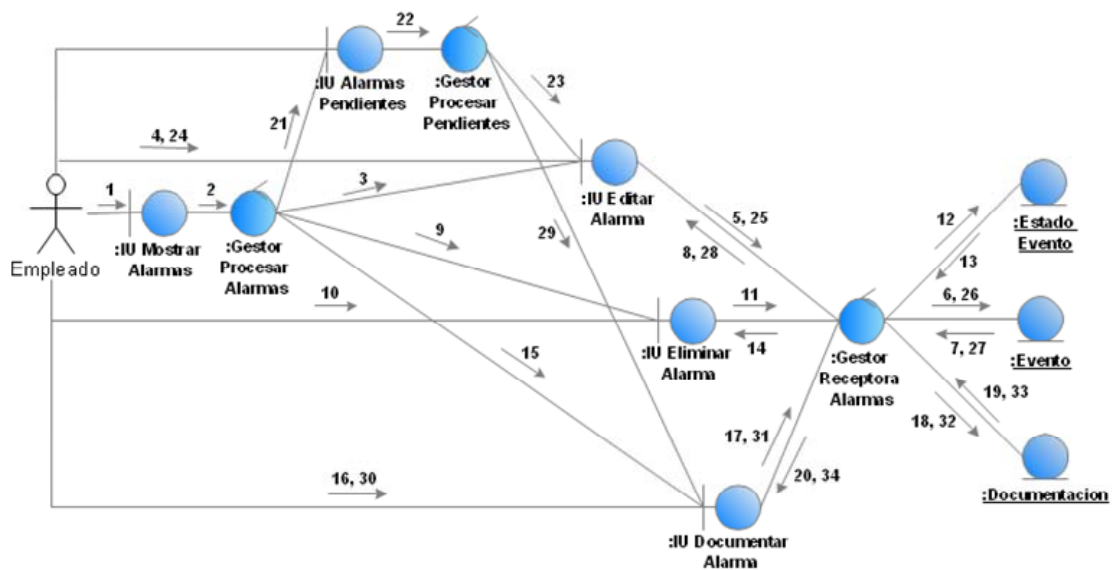
realizará las consultas pertinentes para almacenar los datos (18) dicho gestor retorna el estado del registro (19), que se transfiere al gestor receptora alarmas y a su vez es enviado a la interfaz para luego mostrar un mensaje de conclusión satisfactoria (20).

En caso de que el empleado desee consultar las alarmas pendientes se procede a desplegar dicha interfaz (21) donde el usuario podrá visualizar todas las alarmas pendientes hasta el momento. Desde la interfaz alarmas pendientes se puede proceder con otras acciones como es el caso de editar y documentar las alarmas mostradas, dichas opciones serán mostradas al momento que el gestor procesar pendientes (22) verifique cada una de las alarmas. Si se desea editar una alarma se activará la interfaz editar alarma (23), la cual mostrará un formulario con todos los datos de la alarma que se pueden modificar, el actor ingresará los datos correspondientes (24), y estos serán transferidos al gestor receptora alarmas (25) donde serán validados y enviados a la entidad evento para ser almacenados en la base de datos del sistema (26), una vez guardados los cambios se transfiere un mensaje de respuesta (27), el cual será mostrado en la interfaz al usuario (28). Si se desea documentar un registro se activa la interfaz para documentar (29), la cual mostrará un formulario con las documentaciones correspondientes a la alarma y los campos a llenar, el usuario ingresa los datos pertinentes (30), y estos son enviados al gestor correspondiente para ser validados (31) y transferidos a la entidad documentación (32) que se encargará de almacenar dichos datos. Se devuelve el estado de la operación (33) y se transfiere el mensaje de respuesta mostrando la conclusión de la operación (34).

4.2.1.3. Descripción del Diagrama de Colaboración para el caso de uso Gestionar Alarmas: Consultar Alarmas Cerradas

El caso de uso comienza cuando el actor selecciona la opción que desea ejecutar en la interfaz principal como se muestra en la figura 4.3, en caso de seleccionar buscar alarma, se activara la interfaz correspondiente (1, 26) y el gestor

procesar alarmas se encarga de procesar la selección realizada (2, 27) y enviarla al gestor receptora alarmas cerradas donde se validara el atributo ingresado (3, 28) y se transferirá a la entidad evento donde se realizaran las consultas de los datos de la(s) alarma(s) (4, 29), la entidad devuelve los datos consultados (5, 30) y estos son transferidos al gestor procesar alarmas (6, 31) para ser mostrados al usuario en la interfaz (7, 32).



**Figura 4.2. Diagrama de Colaboración para el caso de uso Gestionar Alarmas:
Consultar Alarmas.**

Fuente: Elaboración Propia.

La tabla 4.2 muestra la leyenda para el diagrama de colaboración de la figura 4.2.

Tabla 4.2. Leyenda para el Diagrama de Colaboración para el caso de uso Gestionar Alarmas: Consultar Alarmas.

1→ Activar interfaz mostrar alarmas	2→ Enviar selección realizada	3→ Activar interfaz editar alarma
4→ Ingresar datos a actualizar	5→ Transferir datos actualizados	6→ Enviar datos validados
7→ Transferir estado de actualización	8→ Mostrar mensaje conclusión satisfactoria	9→ Mostrar interfaz eliminar
10→ Seleccionar factor	11→ Transferir datos a eliminar	12→ Enviar datos validados
13→ Mensaje de eliminación satisfactoria	14→ Recibo de mensaje	15→ Mostrar interfaz documentar
16→ Ingresar datos de la documentación	17→ Transferir datos documentación	18→ Enviar datos validados
19→ Transferir mensaje de conclusión satisfactoria	20→ Mostrar mensaje	21→ Activar interfaz alarmas pendientes
22→ Enviar selección realizada	23→ Activar interfaz editar alarma	24→ Ingresar datos a actualizar
25→ Transferir datos actualizados	26→ Gestionar datos a actualizar	27→ Transferir mensaje de respuesta
28→ Mostrar mensaje de respuesta	29→ Mostrar interfaz documentar alarma	30→ Ingresar datos documentación
31→ Transferir datos documentación	32→ Gestionar datos documentación	33→ Transferir mensaje de conclusión satisfactoria
34→ Mostrar mensaje de conclusión satisfactoria		

Fuente: Elaboración Propia.

Desde las interfaces buscar alarma y búsqueda avanzada alarma se puede proceder con otras acciones como es el caso de editar la alarma (8, 33) donde el actor procederá a registrar todos los datos correspondientes para la actualización de dicha alarma (9, 34) que serán validados y transferidos por el gestor receptora alarmas

cerradas (10, 35) a la entidad evento (11, 36) el cual se encargará de actualizar los datos; esta entidad retorna el estado del registro (12, 37) que se transfiere a la interfaz y se muestra un mensaje de conclusión satisfactoria (13, 38). Si la opción seleccionada corresponde a eliminar alarma se desplegará la interfaz correspondiente (14, 39), por medio de la cual el usuario seleccionará la opción a eliminar (15, 40), el número de caso de la alarma será enviado al gestor receptora alarmas cerradas (16, 41) donde será validado y enviado a la clase entidad estado evento (17, 42) el cual realizara las operaciones necesarias para cambiar el estado de la alarma, y generara un mensaje de conclusión satisfactoria (18, 43), el cual será mostrado al usuario por medio de la interfaz correspondiente (19, 44). Si se desea documentar la alarma entonces se desplegará la interfaz documentar (20, 45), a partir de la cual el usuario ingresara la información correspondiente (21, 46), que será transferida al gestor receptora alarmas para validar los datos ingresados (22, 47), y enviarlos a la entidad documentación (23, 48) donde se realizarán las operaciones necesarias para almacenar la información, una vez guardados los datos se transfiere un mensaje de conclusión satisfactoria (24, 49), se muestra al usuario en la interfaz correspondiente (25, 50).

La tabla 4.3 muestra la leyenda para el diagrama de colaboración de la figura 4.3.

4.2.1.4. Descripción del Diagrama de Colaboración para el caso de uso Gestionar Alarmas: Gestionar Configuración

La figura 4.4 muestra el diagrama de colaboración para el caso de uso gestionar configuración. El actor selecciona la interfaz acorde al tipo de operación que desee realizar en la interfaz principal (1), y el gestor procesar alarmas se encarga de procesar su elección (2), en el caso de seleccionar la opción de ingresar se activa esta interfaz (3), donde el actor ingresará los datos del nuevo registro (4), que serán

transferidos al gestor receptora opciones donde serán validados (5) y enviados a la entidad pertinente para ser almacenados en la base de datos del sistema (6), dicha entidad retorna el estado del registro (7) que se transfiere al gestor correspondiente y se muestra un mensaje de conclusión satisfactoria (8).

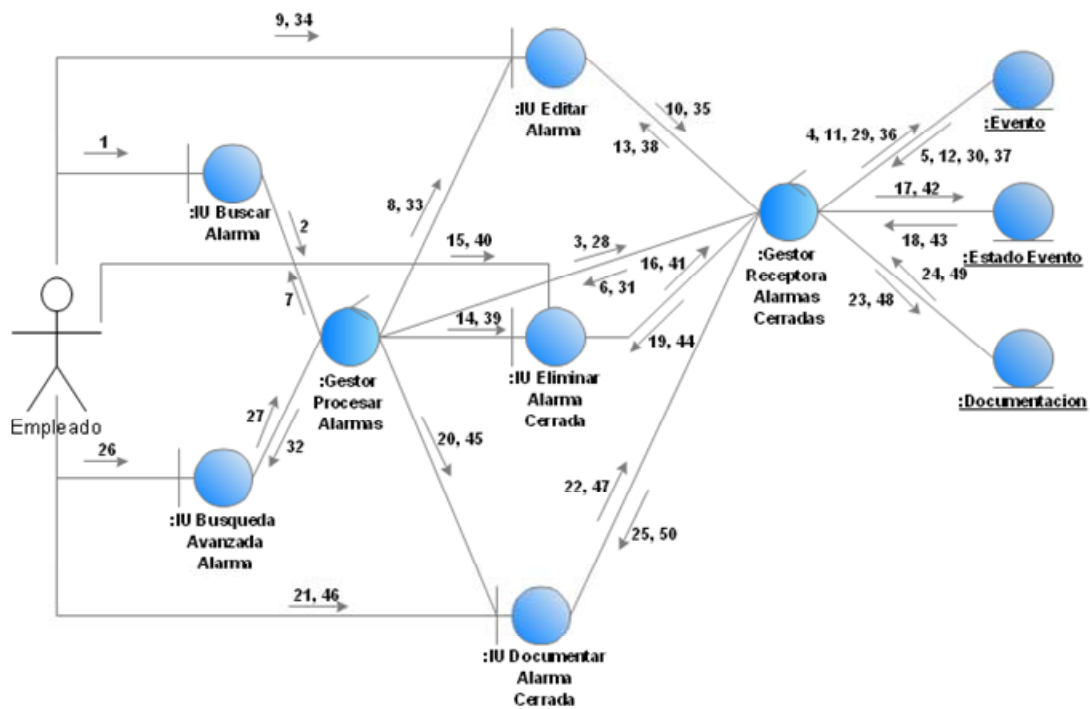


Figura 4.3. Diagrama de Colaboración para el caso de uso Gestionar Alarmas:

Consultar Alarmas Cerradas.

Fuente: Elaboración Propia.

Tabla 4.3. Leyenda para el Diagrama de Colaboración para el caso de uso Gestionar Alarmas: Consultar Alarmas Cerradas.

1,26→ Activar interfaz seleccionada.	2,27→ Enviar solicitud realizada	3,28→ Validar solicitud
4,29→ Consultar datos	5,30→Transferir datos	6,31→ Enviar datos
7,32→ Mostrar datos	8,33→ Activar interfaz editar	9,34→ Ingresar datos a modificar
10,35→ Transferir datos actualizados	11,36→ Realizar operaciones de consultas	12,37→ Devolver Estado de almacenamiento
13,38→ Mostrar Estado	14,39→ Activar interfaz eliminar	15,40→ Seleccionar opción
16,41→ Transferir número de caso	17,42→ Cambiar estado alarma	18,43→ Transferir mensaje de conclusión satisfactoria
19,44→ Mostrar mensaje	20,45→ Activar interfaz documentar	21,46→ Ingresar datos documentación
22,47→ Transferir datos	23,48→ Validar datos	24,49→ Transferir mensaje de conclusión satisfactoria
25,50→ Mostrar mensaje		

Fuente: Elaboración Propia.

Si la acción seleccionada corresponde a editar se desplegará la interfaz correspondiente (9) con los datos de la opción, a partir de los cuales el actor podrá actualizarlos (10), y serán validados (11), y transferidos a la entidad correspondiente según sea el caso (12), dicha entidad genera una respuesta de actualización (13) que será enviada y mostrada en la interfaz editar (14). En caso de seleccionar la acción eliminar se mostrará la interfaz correspondiente (15) a partir de la cual el usuario seleccionará la opción (16), y esta será validada por el gestor receptora opciones (17) el cual transferirá la selección a la entidad correspondiente (18) donde se realizarán las operaciones necesarias para eliminar dicho registro, y generar un estado de retiro (19), el cual será enviado como respuesta y mostrado en interfaz principal (20).

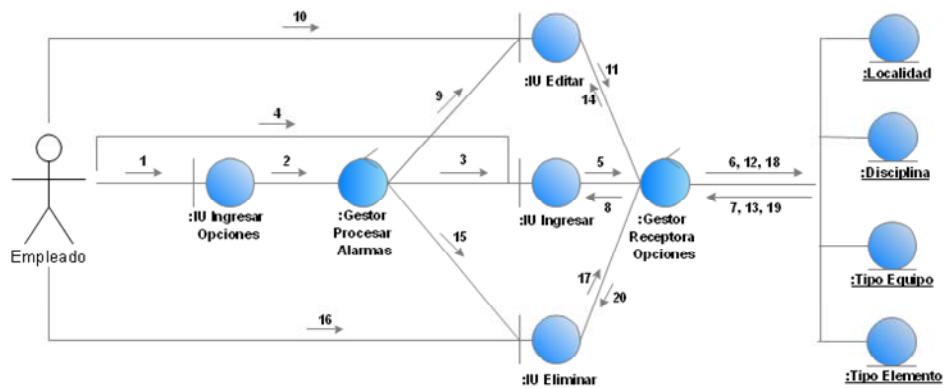


Figura 4.4. Diagrama de Colaboración para el caso de uso Gestionar Alarmas: Gestionar Configuración.

Fuente: Elaboración Propia.

La tabla 4.4 muestra la leyenda para el diagrama de colaboración de la figura 4.4.

Tabla 4.4. Leyenda para el Diagrama de Colaboración para el caso de uso Gestionar Alarmas: Gestionar Configuración.

1→ Seleccionar opción	2→ Transferir selección realizada	3→ Activar interfaz ingresar
4→ Ingresar datos nuevo registro	5→ Transferir datos nuevo registro	6→ Enviar datos para almacenarlos
7→ Devolver estado de registro	8→ Mostrar mensaje de conclusión satisfactoria	9→ Desplegar interfaz editar
10→ Modificar datos	11→ Transferir datos	12→ Enviar datos actualizados para ser almacenados
13→ Transferir respuesta de actualización	14→ Mostrar respuesta de actualización	15→ Activar interfaz eliminar
16→ Seleccionar opción	17→ Transferir selección	18→ Validar selección
19→ Devolver estado de retiro	20→ Mostrar respuesta	

Fuente: Elaboración Propia.

4.2.1.5. Descripción del Diagrama de Colaboración para el caso de uso Gestionar Eventos: Ingresar Eventos

La figura 4.5 muestra el diagrama de colaboración para el caso de uso ingresar eventos. El actor involucrado en el caso de uso selecciona la opción que desea ejecutar en la interfaz principal ingresar eventos (1) y el gestor procesar eventos se encarga de procesarla (2), en caso de que el actor seleccione la opción evento manual, se activará la interfaz para ingresar eventos de tipo manual (3) donde se desplegará un formulario que permitirá al usuario ingresar los datos del nuevo evento incluyendo su documentación (4) que serán transferidos al gestor receptora eventos (5) para validarlos y enviarlos a las clases entidad, evento (6, 7), documentación (8, 9), estado evento (10, 11) y mantenimiento (12, 13) para ser almacenados en la base de datos del sistema, se transferirá un mensaje de conclusión satisfactoria el cual será mostrado al usuario por la interfaz correspondiente (14).

Si la opción seleccionada corresponde a la interfaz evento adicional (15) se desplegará un formulario a partir del cual el usuario ingresará los datos del evento a registrar (16), estos serán enviados para ser validados (17) y transferidos en las entidades correspondientes (18 y 19, 20 y 21, 22 y 23, 24 y 25), el estado de registro será transferido y mostrado por la interfaz evento adicional (26).

La tabla 4.5 muestra la leyenda para el diagrama de colaboración de la figura 4.5.

4.2.1.6. Descripción del Diagrama de Colaboración para el caso de uso Gestionar Eventos: Consultar Eventos

El actor identificado como empleado representa a todos los actores que interactúan con este caso de uso, es decir, al administrador del sistema y el analista de monitoreo como se muestra en la figura 4.6.

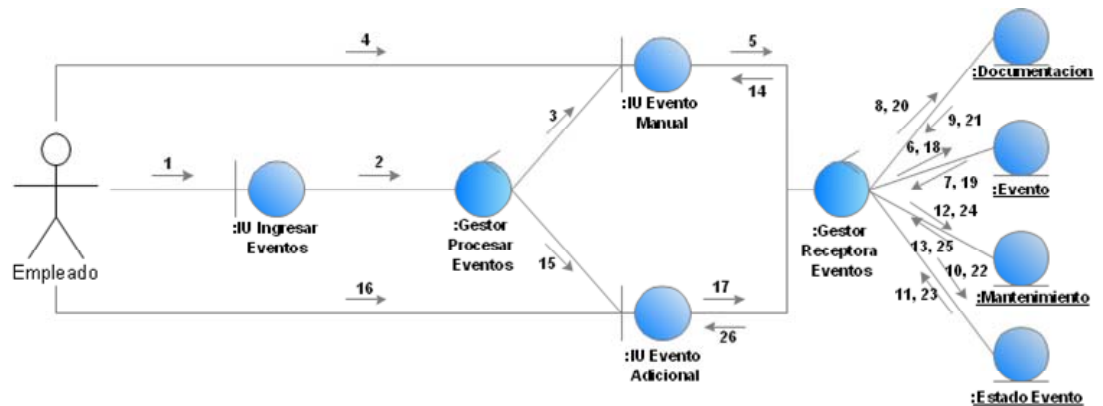


Figura 4.5. Diagrama de Colaboración para el caso de uso Gestionar Eventos: Ingresar Eventos.

Fuente: Elaboración Propia.

Este caso de uso comienza cuando el actor selecciona la interfaz acorde al tipo de acción que va a realizar en la interfaz principal de consultar eventos (1), la cual permite visualizar todos los eventos ingresados por los usuarios, el gestor procesar eventos se encarga de procesar su elección (2) de acuerdo al evento seleccionado, en caso de seleccionar la opción editar, se activa la interfaz editar evento (3), en la cual se desplegarán todos los datos correspondiente al evento seleccionado, y el usuario podrá ingresar los datos pertinentes (4), que serán validados por el gestor receptora eventos (5) y transferidos a la entidad evento (6) la cual se encarga de realizar las operaciones necesarias para actualizar los datos y retornar el estado del registro (7), el cual será enviado como respuesta a la interfaz, y se muestra un estado de conclusión satisfactoria o un error dependiendo del caso (8).

En caso de seleccionar la opción eliminar evento, se activará la interfaz correspondiente (9) permitiendo al usuario escoger una opción (10), los datos de dicho evento serán enviados al gestor receptora eventos (11), en el cual serán validados para verificar si es posible o no cambiar el estado del evento. Luego los datos son transferidos a la entidad estado evento (12) que se encargará de cambiar el

estado del evento, para luego retornar al gestor receptora eventos (13) el estado de la operación y mostrarlo al usuario (14).

Tabla 4.5. Leyenda para el Diagrama de Colaboración para el caso de uso Gestionar Eventos: Ingresar Eventos.

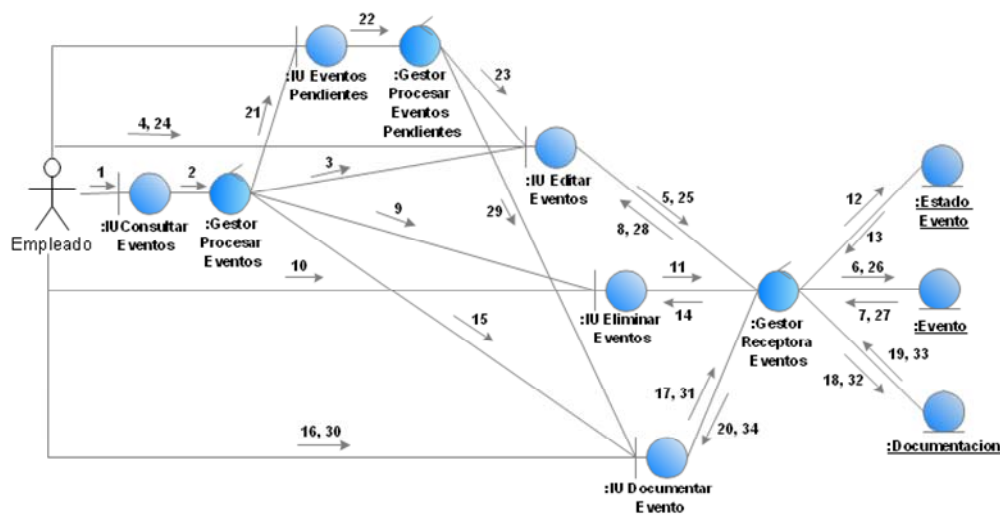
1→ Seleccionar opción	2→ Transferir selección	3→ Activar interfaz evento manual
4→ Ingresar datos nuevo evento	5→ Transferir datos	6,8,10,12→ Enviar datos procesados para almacenarlos
7,9,11,13→ Devolver estado de registro	14→ Mostrar estado de registro	15→ Desplegar interfaz evento adicional
16→ Ingresar datos nuevo evento	17→ Transferir datos	18,20,22,24→ Enviar datos procesados para almacenarlos
19,21,23,25→ Devolver estado de registro	16→ Mostrar estado de registro	

Fuente: Elaboración Propia.

Si la opción seleccionada corresponde a documentar evento se activará la interfaz documentar evento (15), donde el empleado ingresará los atributos correspondientes a la documentación de la falla (16), estos datos serán enviados al gestor receptora eventos (17), donde serán validados y transferidos a la entidad documentación (18), que realizará las consultas pertinentes para almacenar los datos en la base de datos del sistema, dicha entidad retorna el estado del registro, que se transfiere al gestor receptora eventos (19), y a su vez es enviado a la interfaz para luego mostrar un mensaje de conclusión satisfactoria (20).

En caso de que el empleado desee consultar los eventos pendientes se procede a desplegar dicha interfaz (21) donde podrá visualizar todos los eventos pendientes hasta el momento. Desde la interfaz eventos pendientes se puede proceder con otras

acciones como es el caso de editar y documentar los eventos mostrados, dichas opciones serán mostradas al momento que el gestor procesar eventos pendientes (22) verifique cada una de las fallas. Si se desea editar un evento se activará la interfaz editar evento (23), la cual mostrará un formulario con todos los datos del evento que se pueden modificar, el actor ingresará los datos correspondientes (24), y estos serán transferidos al gestor receptora eventos (25) donde serán validados y enviados a la entidad evento (26), para ser almacenados, una vez guardados los cambios se transfiere un mensaje de respuesta (27), el cual será mostrado en la interfaz al usuario (28). Si se desea documentar un registro se activa la interfaz para documentar (29), la cual mostrará un formulario con las documentaciones correspondientes al evento y los campos a llenar, el usuario ingresa los datos pertinentes (30), y estos son enviados al gestor correspondiente para ser validados (31) y transferidos a la entidad documentación (32) la cual se encargará de almacenar dichos datos, se devuelve el estado de la operación (33) y se transfiere el mensaje de respuesta mostrando la conclusión de la operación (34).



**Figura 4.6. Diagrama de Colaboración para el caso de uso Gestionar Eventos:
Consultar Eventos.**

Fuente: Elaboración Propia.

La tabla 4.6 muestra la leyenda para el diagrama de colaboración de la figura 4.6.

Tabla 4.6. Leyenda para el Diagrama de Colaboración para el caso de uso Gestionar Eventos: Consultar Eventos.

1→ Activar interfaz consultar eventos	2→ Enviar selección realizada	3→ Activar interfaz editar evento
4→ Ingresar datos a actualizar	5→ Transferir datos actualizados	6→ Enviar datos validados
7→ Transferir estado de actualización	8→ Mostrar mensaje conclusión satisfactoria	9→ Mostrar interfaz eliminar evento
10→ Seleccionar factor	11→ Transferir datos a eliminar	12→ Enviar datos validados
13→ Mensaje de eliminación satisfactoria	14→ Recibo de mensaje	15→ Mostrar interfaz documentar evento
16→ Ingresar datos de la documentación	17→ Transferir datos documentación	18→ Enviar datos validados
19→ Transferir mensaje de conclusión satisfactoria	20→ Mostrar mensaje	21→ Activar interfaz eventos pendientes
22→ Enviar selección realizada	23→ Activar interfaz editar evento	24→ Ingresar datos a actualizar
25→ Transferir datos actualizados	26→ Gestionar datos a actualizar	27→ Transferir mensaje de respuesta
28→ Mostrar mensaje de respuesta	29→ Mostrar interfaz documentar evento	30→ Ingresar datos documentación
31→ Transferir datos documentación	32→ Gestionar datos documentación	33→ Transferir mensaje de conclusión satisfactoria
34→ Mostrar mensaje de conclusión satisfactoria		

Fuente: Elaboración Propia.

4.2.1.7. Descripción del Diagrama de Colaboración para el caso de uso Gestionar Eventos: Consultar Eventos Cerrados

El caso de uso comienza cuando el actor selecciona la opción que desea ejecutar en la interfaz principal como se muestra en la figura 4.7, en caso de seleccionar buscar evento, se activara la interfaz correspondiente (1, 26) y el gestor procesar eventos se encarga de procesar la selección realizada (2, 27) y enviarla al gestor receptora eventos cerrados donde se validara el atributo ingresado (3, 28) y se transferirá a la entidad evento donde se realizaran las consultas de los datos de el(los) evento(s) (4, 29), la entidad devuelve los datos consultados (5, 30) y estos son transferidos al gestor procesar eventos (6, 31) para ser mostrados al usuario en la interfaz (7, 32). Desde las interfaces buscar evento y búsqueda avanzada evento se puede proceder con otras acciones como es el caso de editar el evento (8, 33) donde el actor procederá a registrar todos los datos correspondientes para la actualización de dicho evento (9, 34) que serán validados y transferidos por el gestor receptora eventos cerrados (10, 35) a la entidad evento (11, 36) el cual se encargará de actualizar los datos; esta entidad retorna el estado del registro (12, 37) que se transfiere a la interfaz y se muestra un mensaje de conclusión satisfactoria (13, 38). Si la opción seleccionada corresponde a eliminar evento cerrado se desplegará la interfaz correspondiente (14, 39), por medio de la cual el usuario seleccionará la opción a eliminar (15, 40), el número de caso del evento será enviado al gestor receptora eventos cerrados (16, 41) donde será validado y enviado a la clase entidad estado evento (17, 42) el cual realizara las operaciones necesarias para cambiar el estado del evento, y generara un mensaje de conclusión satisfactoria (18, 43), el cual será mostrado al usuario por medio de la interfaz correspondiente (19, 44). Si se desea documentar el evento entonces se desplegará la interfaz documentar evento cerrado (20, 45), a partir de la cual el usuario ingresara la información correspondiente (21, 46), que será transferida al gestor receptora eventos para validar los datos ingresados (22, 47), y enviarlos a la entidad documentación (23, 48) donde se realizarán las

operaciones necesarias para almacenar la información, una vez guardados los datos se transfiere un mensaje de conclusión satisfactoria (24, 49) y se muestra al usuario en la interfaz correspondiente (25, 50).

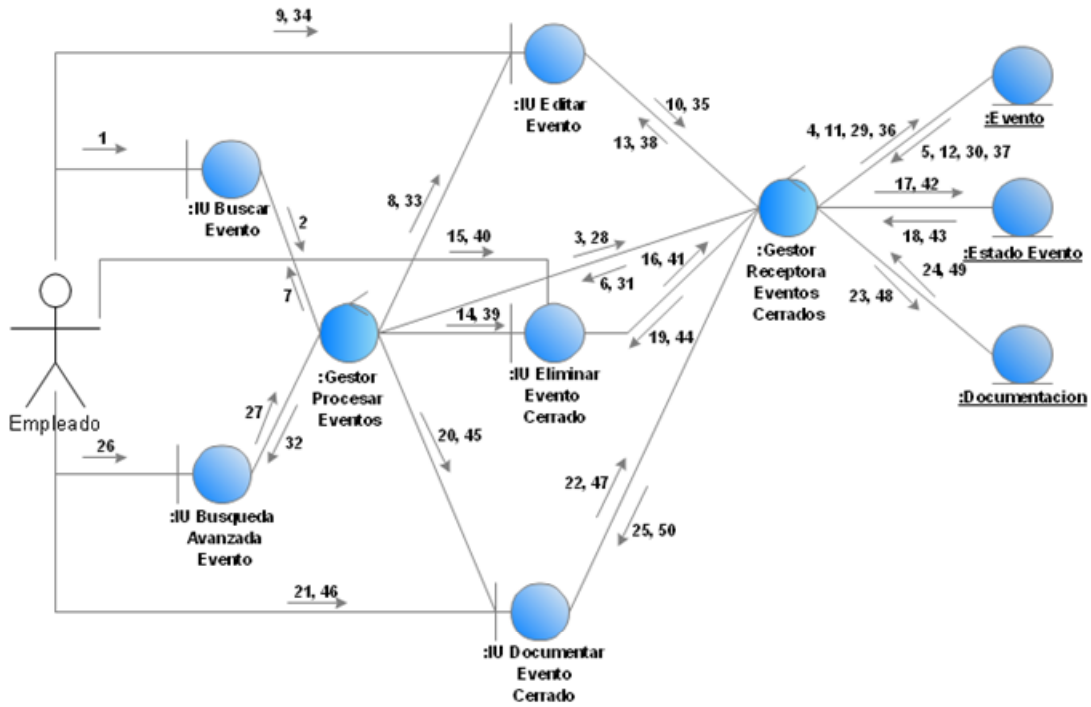


Figura 4.7. Diagrama de Colaboración para el caso de uso Gestionar Eventos: Consultar Eventos Cerrados.

Fuente: Elaboración Propia.

La tabla 4.7 muestra la leyenda para el diagrama de colaboración mostrado en la figura 4.7.

4.2.1.8. Descripción del Diagrama de Colaboración para el caso de uso Gestionar Servicios

Los actores involucrados en este caso de uso corresponden a el administrador del sistema, el analista de monitoreo y el analista de control de gestión, los cuales van a ser representados con el nombre de empleado como se muestra en la figura 4.8.

Tabla 4.7. Leyenda para el Diagrama de Colaboración para el caso de uso Gestionar Eventos: Consultar Eventos Cerrados.

1,26→ Activar interfaz seleccionada.	2,27→ Enviar solicitud realizada	3,28→ Validar solicitud
4,29→ Consultar datos	5,30→ Transferir datos	6,31→ Enviar datos
7,32→ Mostrar datos	8,33→ Activar interfaz editar	9,34→ Ingresar datos a modificar
10,35→ Transferir datos actualizados	11,36→ Realizar operaciones de consultas	12,37→ Devolver Estado de almacenamiento
13,38→ Mostrar Estado	14,39→ Activar interfaz eliminar	15,40→ Seleccionar opción
16,41→ Transferir número de caso	17,42→ Cambiar estado evento	18,43→ Transferir mensaje de conclusión satisfactoria
19,44→ Mostrar mensaje	20,45→ Activar interfaz documentar	21,46→ Ingresar datos documentación
22,47→ Transferir datos	23,48→ Validar datos	24,49→ Transferir mensaje de conclusión satisfactoria
25,50→ Mostrar mensaje		

Fuente: Elaboración Propia.

Si el usuario desea ingresar un nuevo registro se despliega la interfaz ingresar servicio donde el empleado suministrará todos los datos relacionados con el nuevo servicio (1) que serán validados por el gestor receptora servicio (2) y enviados a la entidad servicio que se encargará de almacenar los datos del nuevo servicio (3), y

devolverá un mensaje con el estado del almacenamiento de los datos (4) el cual se enviara como respuesta a la interfaz y será mostrado como parte del estado del registro al usuario (5).

Para buscar servicio el empleado debe ingresar el atributo de la consulta (6), que será validado por el gestor receptora servicios (7) y enviados a la entidad servicio encargada de solicitar los datos del servicio (8), dicha entidad transferirá los datos solicitados (9), y mostrados en la interfaz buscar servicio (10).

Si el actor desea modificar los datos del servicio se desplegará la interfaz de modificar servicio donde se mostrarán los datos del servicio permitiendo modificarlos (11), una vez modificados se pasan los datos al gestor receptora servicio que validará los datos modificados (12) y los transferirá a la entidad servicio (13) para que ésta se encargue de almacenarlos en la base de datos del sistema. Una vez actualizados los datos, el gestor receptora servicio enviará un mensaje de conclusión satisfactoria (14) para mostrar el estado de la actualización al usuario (15).

En caso de desear eliminar algún servicio se seleccionará la opción correspondiente y se desplegará la interfaz eliminar servicio, donde el actor ingresará el atributo del servicio a retirar (16) el cual será enviado al gestor receptora servicio (17) que transmitirá la información a la entidad servicio (18) y eliminará los datos del servicio correspondiente, y a su vez enviará un mensaje de respuesta satisfactoria (19), el cual será transferido a la interfaz correspondiente para ser mostrado al actor (20).

La tabla 4.8 muestra la leyenda para el diagrama de colaboración mostrado en la figura 4.8.

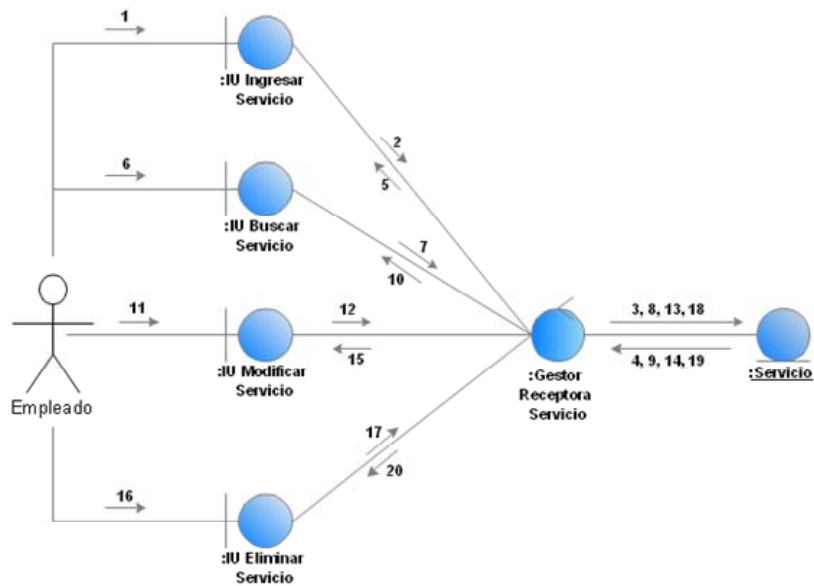


Figura 4.8. Diagrama de Colaboración para el caso de uso Gestionar Servicios

Fuente: Elaboración Propia.

4.2.1.9. Descripción del Diagrama de Colaboración para el caso de uso Generar Reportes

En la figura 4.9 se puede observar el diagrama de colaboración para el caso de uso generar reportes. El actor selecciona una opción e ingresa los datos según sea el caso (1, 8, 13, 18, 23), estos datos serán enviados al gestor receptora reportes, el cual validará los datos (2, 9, 14, 19, 24) y los transferirá a las entidades correspondientes (3, 5, 10, 15, 20, 25) las cuales realizarán las consultas necesarias para solicitar los datos en la base de datos del sistema; dichas entidades devolverán la información solicitada según sea el caso (4, 6, 11, 16, 21, 26), y esta será enviada a las interfaces correspondientes para mostrarlos al usuario (7, 12, 17, 22, 27).

Tabla 4.8. Leyenda para el Diagrama de Colaboración para el caso de uso Gestionar Servicios.

1→ Ingresar datos nuevo registro	2→ Transferir datos de nuevo registro	3→ Enviar datos de nuevo registro para almacenarlos
4→ Devolver estado de registro	5→ Mostrar mensaje de conclusión satisfactoria	6→ Activar interfaz buscar servicio
7→ Transferir datos consulta	8→ Enviar datos a consultar	9→ Devolver datos servicio
10→ Mostrar datos consulta	11→ Ingresar datos a modificar	12→ Transferir datos actualizados
13→ Transferir datos validados	14→ Devolver estado de actualización	15→ Mostrar mensaje de conclusión satisfactoria
16→ Activar interfaz eliminar	17→ Transferir atributo	18→ Procesar retiro
19→ Devolver estado de confirmación retiro	20→ → Mostrar estado de confirmación de retiro	

Fuente: Elaboración Propia.

En la tabla 4.9 se puede observar la leyenda para el diagrama de colaboración de la figura 4.9.

4.3. Flujo de Trabajo de Diseño

En esta fase los casos de uso significativos se diseñan en términos de subsistemas de diseño.

Los propósitos del diseño son:

- Adquirir una comprensión en profundidad de los aspectos relacionados con los requisitos no funcionales y otras restricciones que se suponen.

- Crear una entrada apropiada y un punto de partida para actividades de implementación subsiguientes capturando los requisitos o subsistemas individuales e interfaces.
- Ser capaces de descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo, teniendo en cuenta la posible concurrencia.

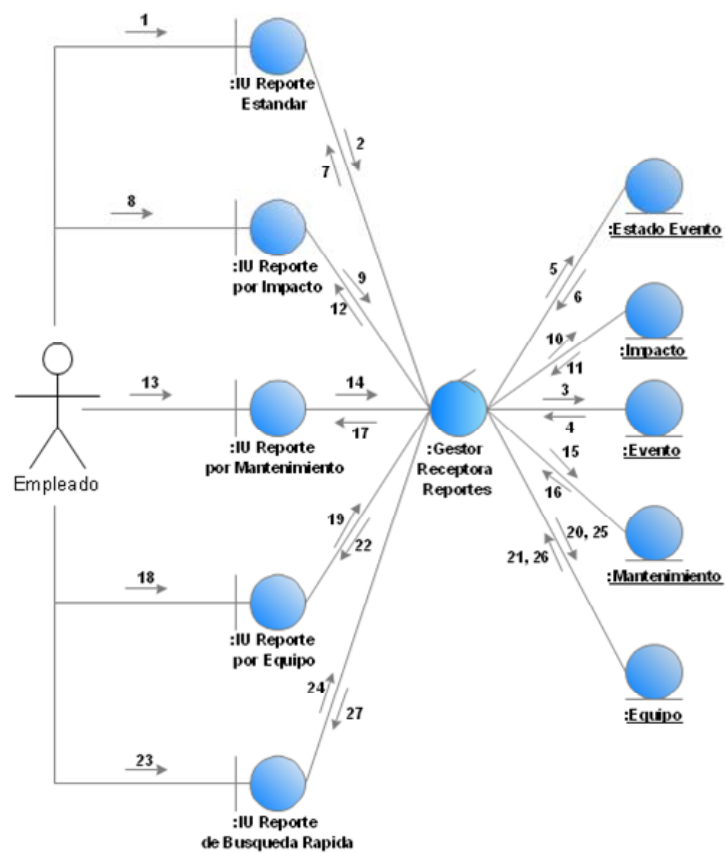


Figura 4.9. Diagrama de Colaboración para el caso de uso Generar Reportes

Fuente: Elaboración Propia.

Tabla 4.9. Leyenda para el Diagrama de Colaboración para el caso de uso Generar Reportes.

1→ Ingresar datos reporte estándar	2→ Transferir datos ingresados	3→ Enviar datos a consultar
4→ Devolver datos eventos	5→ Consultar datos estado evento	6→ Devolver datos estado evento
7→ Mostrar datos consultados al usuario	8→ Ingresar datos reporte por impacto	9→ Transferir datos ingresados
10→ Enviar datos a consultar	11→ Devolver datos impacto	12→ Mostrar datos al usuario
13→ Ingresar datos reporte por mantenimiento	14→ Transferir datos ingresados	15→ Transferir datos consultados
16→ Devolver datos mantenimiento	17→ Mostrar datos al usuario	18→ Ingresar datos reporte por equipo
19→ Transferir datos ingresados	20→ Enviar datos a consultar	21→ Devolver datos equipo
22→ Mostrar datos consultados al usuario	23→ Ingresar datos reporte de búsqueda rápida	24→ Transferir datos ingresados
25→ Enviar datos a consultar	26→ Transferir datos consultados	27→ Mostrar datos consultados al usuario

Fuente: Elaboración Propia.

4.3.1. Diagramas de Clase de Diseño

A fin de facilitar la visualización y comprensión de las clases de diseño, y debido a que RUP se basa en casos de uso, se realizará un diagrama de clases de diseño para cada uno de los casos de uso del sistema.

4.3.1.1. Diagrama de Clase de Diseño Administrar Usuarios

La figura 4.10 representa el diagrama de clase de diseño para el caso de uso Administrar Usuarios.

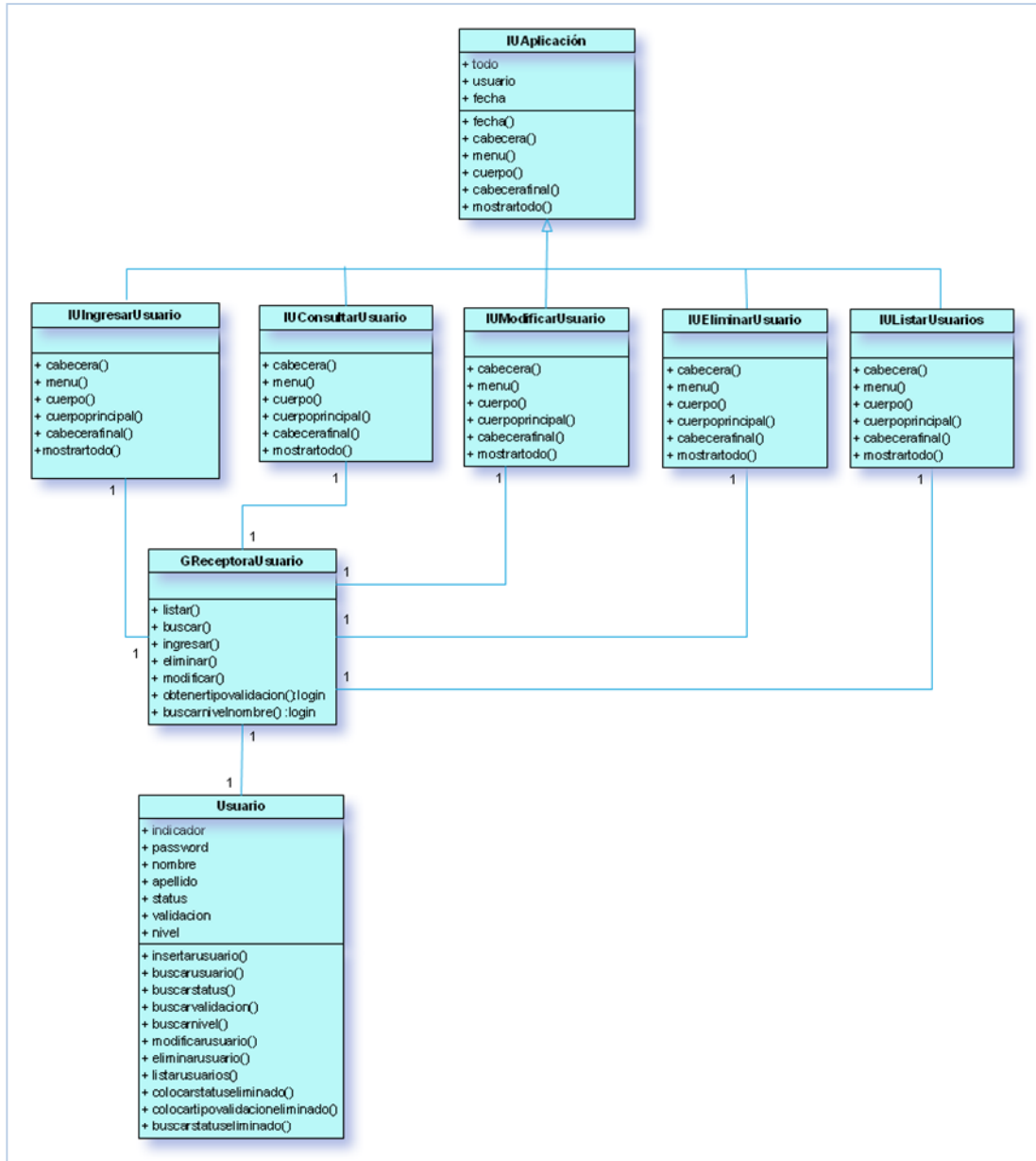


Figura 4.10. Diagrama de Clase de Diseño Administrar Usuarios

Fuente: Elaboración Propia.

4.3.1.2. Diagrama de Clase de Diseño Gestionar Alarmas: Consultar Alarmas

La figura 4.11 representa el diagrama de clase de diseño para el caso de uso Consultar Alarmas.

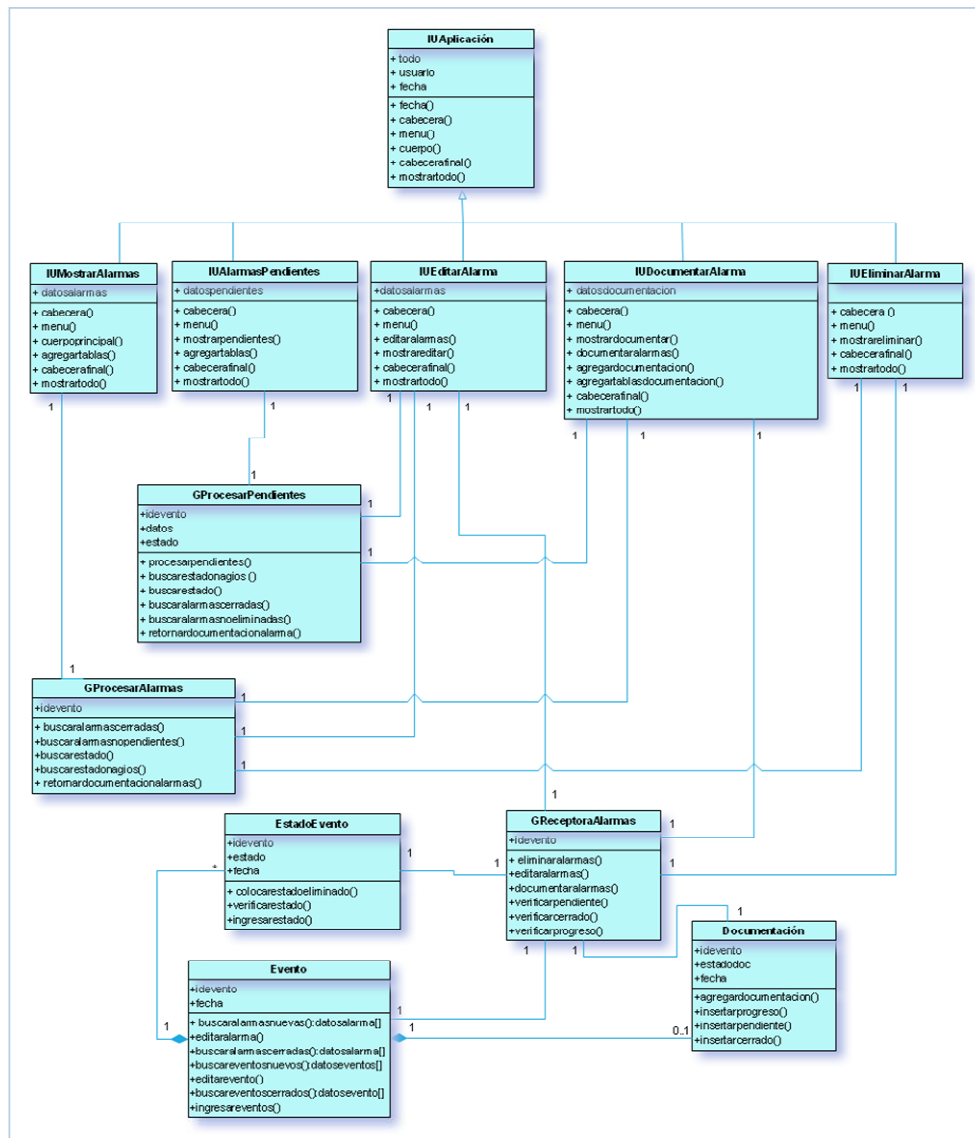


Figura 4.11. Diagrama de Clase de Diseño Gestionar Alarmas: Consultar Alarmas

Fuente: Elaboración Propia.

4.3.1.3. Diagrama de Clase de Diseño Gestionar Alarmas: Consultar Alarmas Cerradas

La figura 4.12 representa el diagrama de clase de diseño para el caso de uso Consultar Alarmas Cerradas.

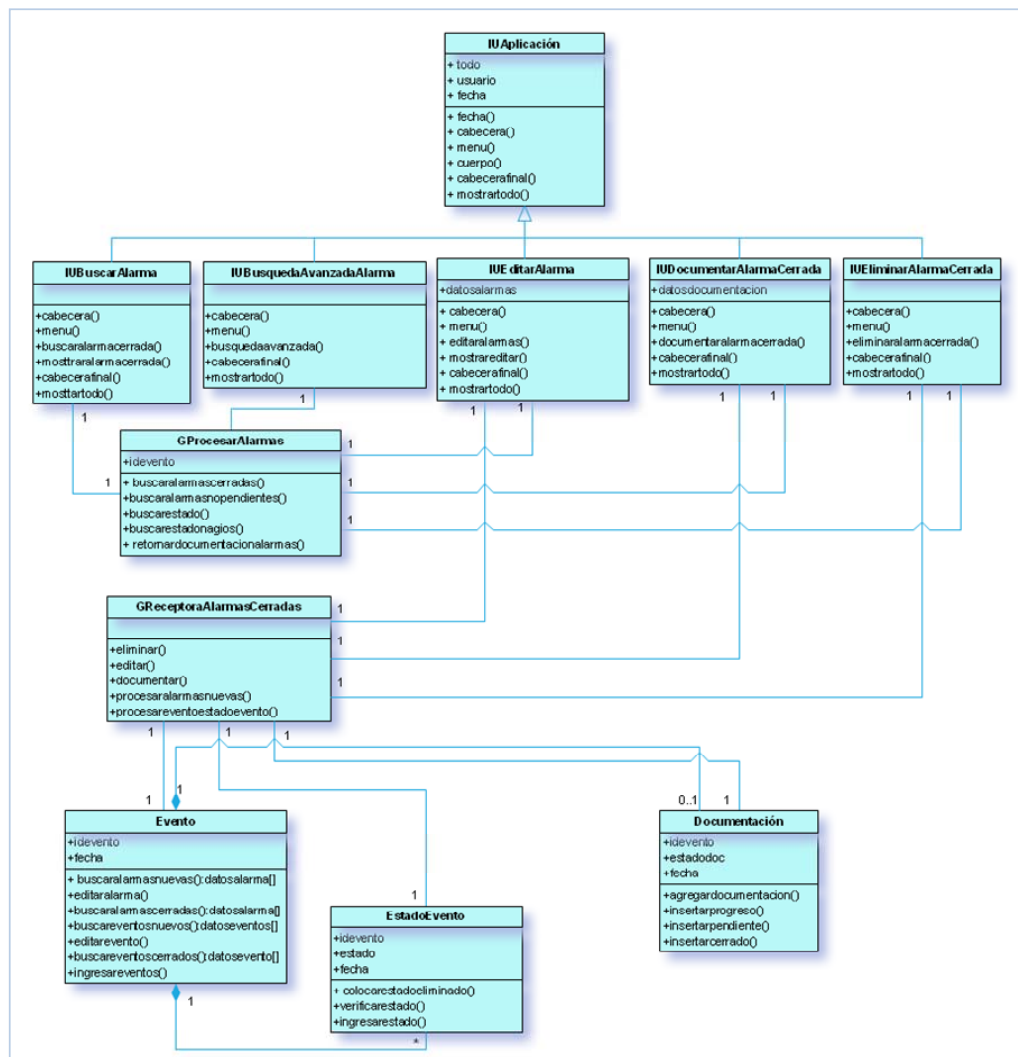


Figura 4.12. Diagrama de Clase de Diseño Gestionar Alarmas: Consultar Alarmas Cerradas

Fuente: Elaboración Propia.

4.3.1.4. Diagrama de Clase de Diseño Gestionar Alarmas: Gestionar Configuración

La figura 4.13 representa el diagrama de clase de diseño para el caso de uso Gestionar Configuración.

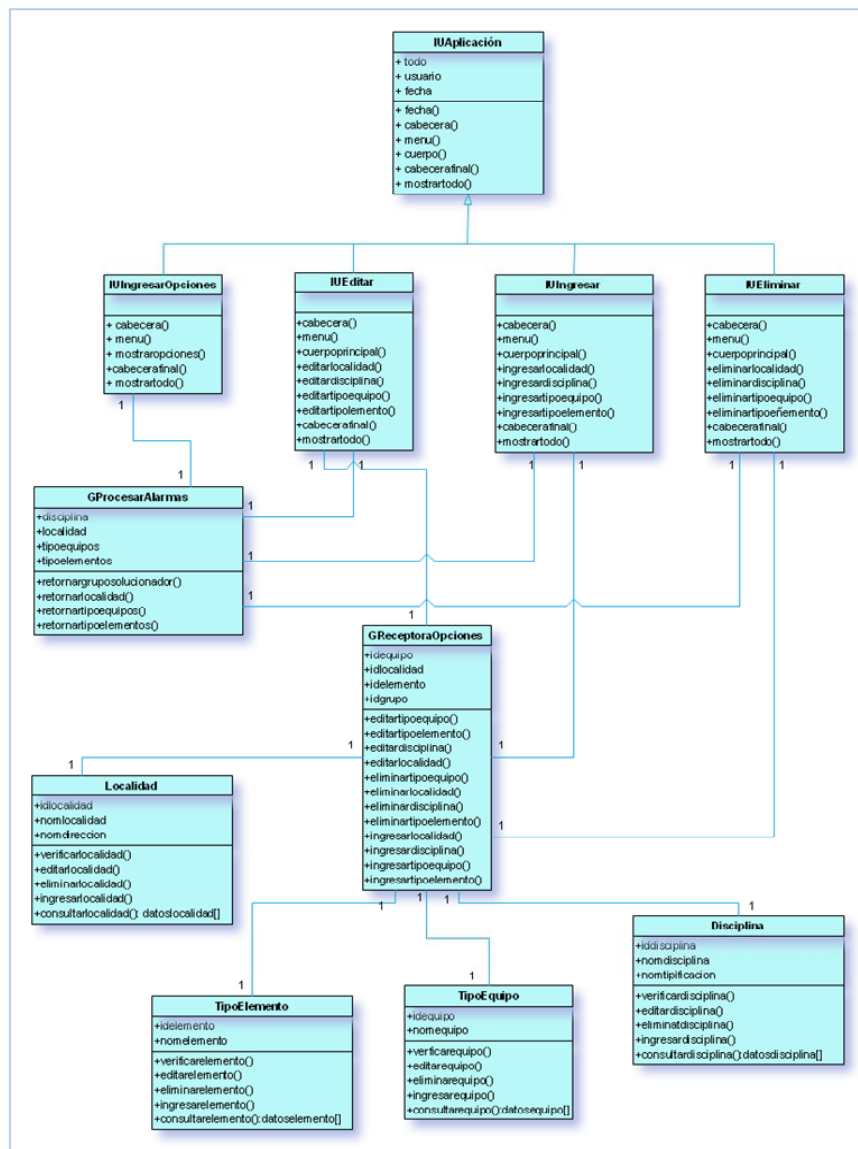


Figura 4.13. Diagrama de Clase de Diseño Gestionar Alarmas: Gestionar Configuración

Fuente: Elaboración Propia.

4.3.1.5. Diagrama de Clase de Diseño Gestionar Eventos: Ingresar Eventos

La figura 4.14 representa el diagrama de clase de diseño para el caso de uso Ingresar Eventos.

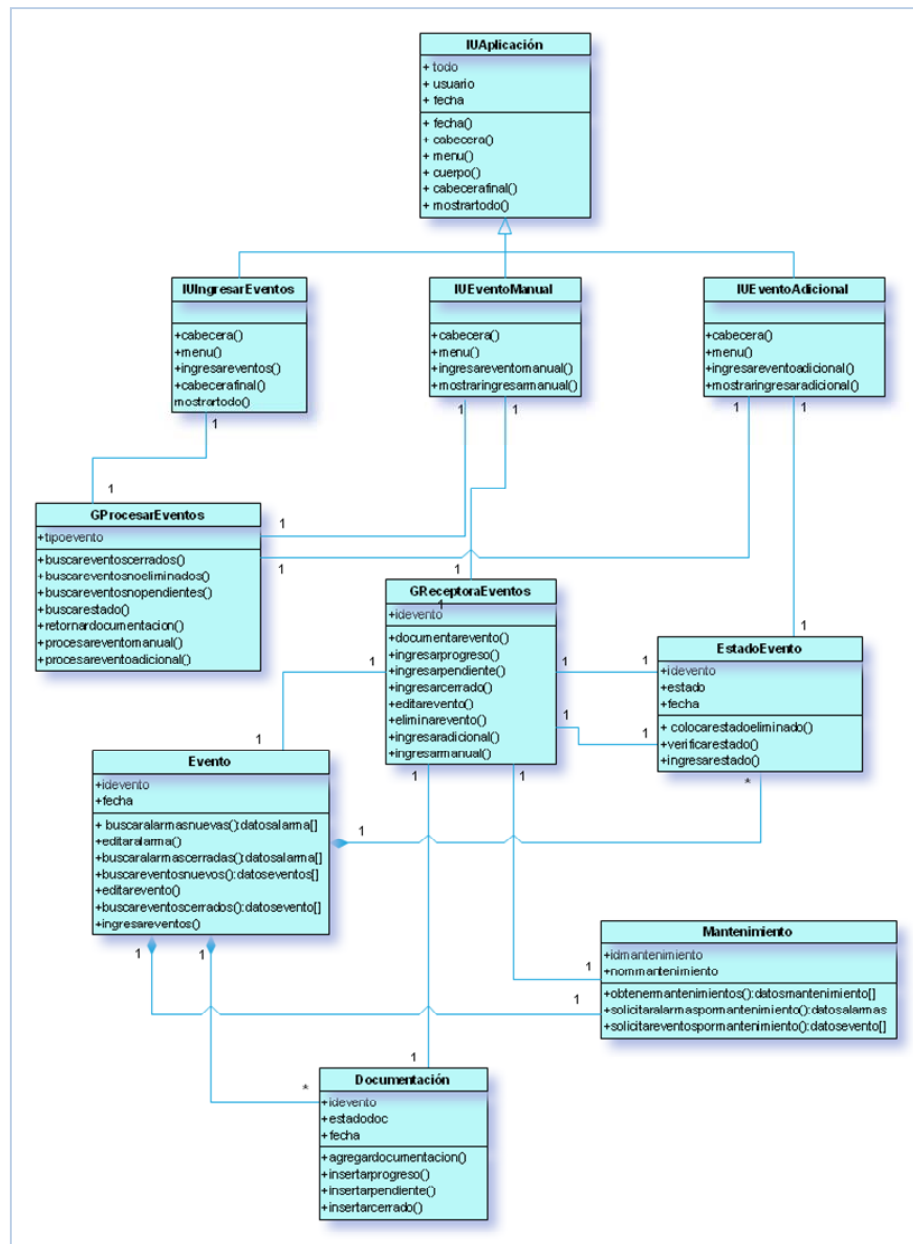


Figura 4.14. Diagrama de Clase de Diseño Gestionar Eventos: Ingresar Eventos

Fuente: Elaboración Propia.

4.3.1.6. Diagrama de Clase de Diseño Gestionar Eventos: Consultar Eventos

La figura 4.15 representa el diagrama de clase de diseño para el caso de uso Consultar Eventos.

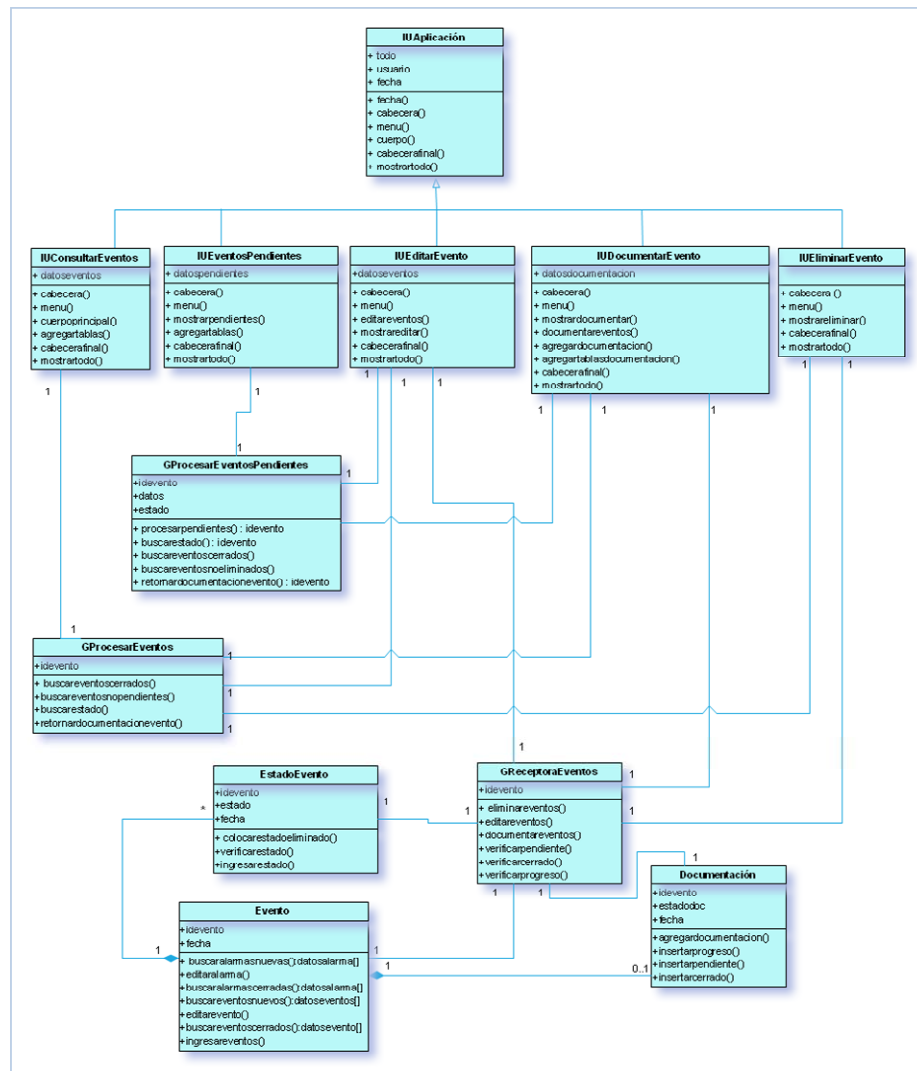


Figura 4.15. Diagrama de Clase de Diseño Gestionar Eventos: Consultar Eventos

Fuente: Elaboración Propia.

4.3.1.7. Diagrama de Clase de Diseño Gestionar Eventos: Consultar Eventos Cerrados

La figura 4.16 representa el diagrama de clase de diseño para el caso de uso Consultar Eventos Cerrados.

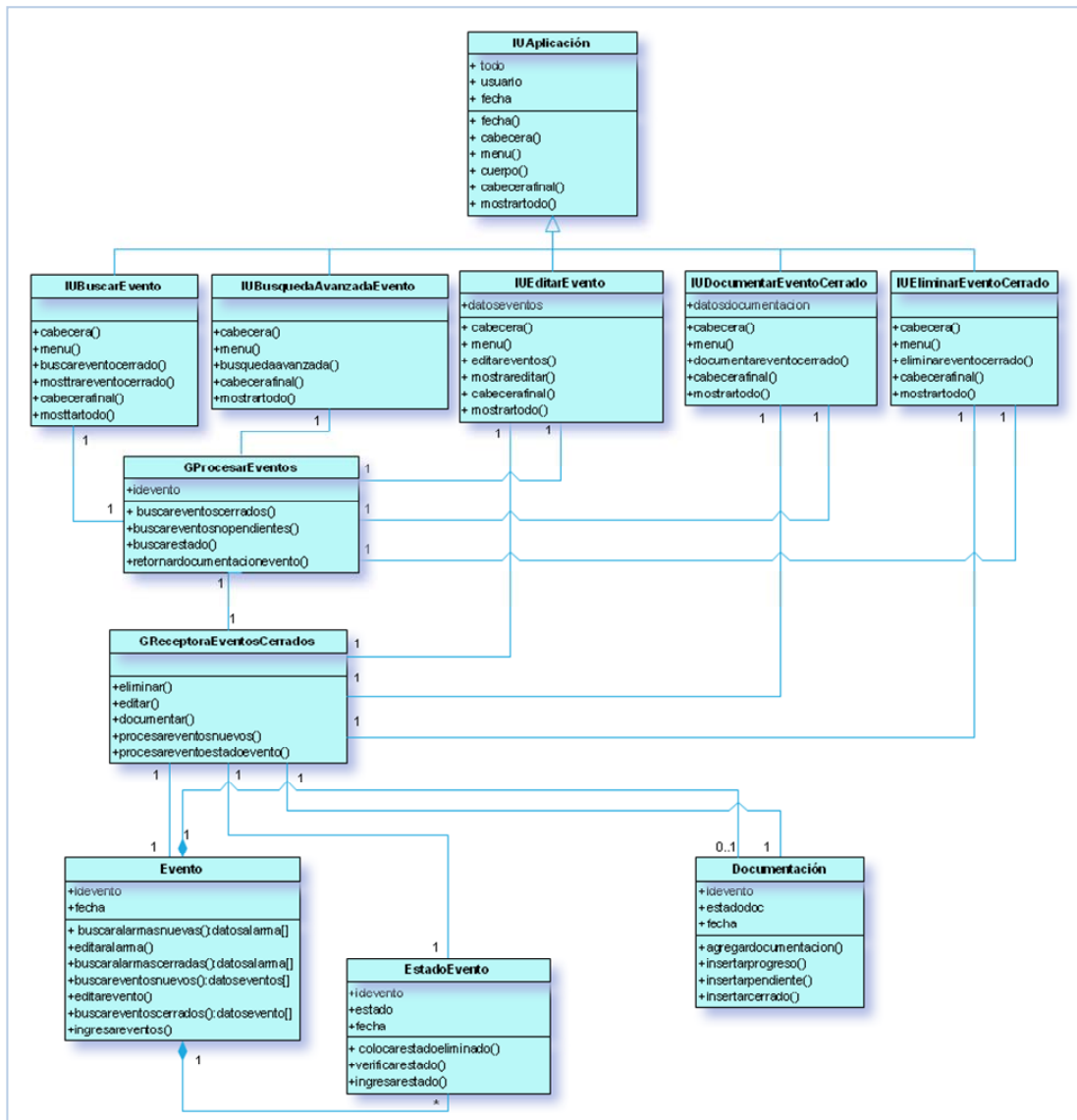


Figura 4.16. Diagrama de Clase de Diseño Gestionar Eventos: Consultar Eventos Cerrados

Fuente: Elaboración Propia.

4.3.1.8. Diagrama de Clase de Diseño Gestionar Servicios

La figura 4.17 representa el diagrama de clase de diseño para el caso de uso Gestionar Servicios.

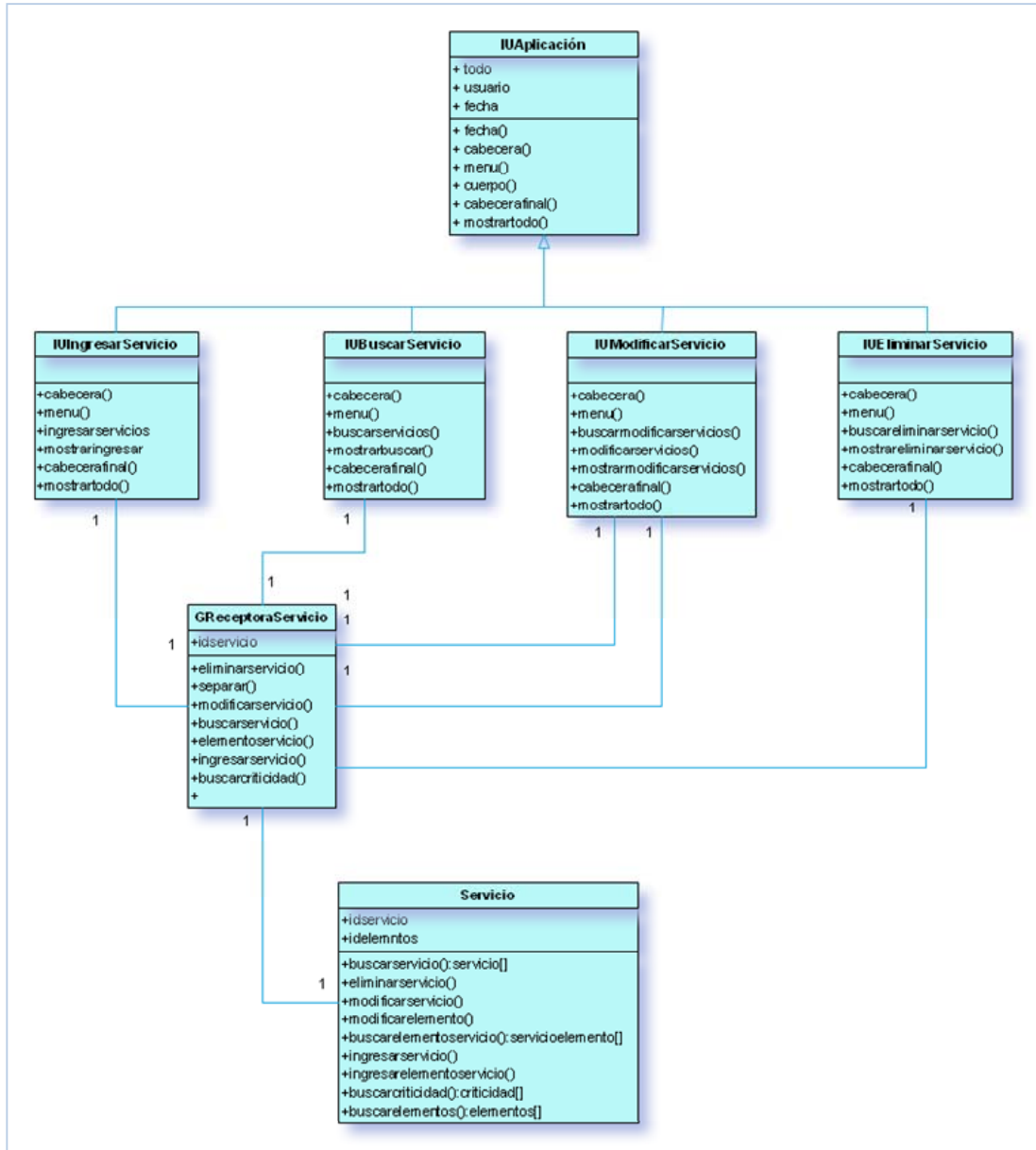


Figura 4.17. Diagrama de Clase de Diseño Gestionar Servicios

Fuente: Elaboración Propia.

4.3.1.9. Diagrama de Clase de Diseño Generar Reportes

La figura 4.18 representa el diagrama de clase de diseño para el caso de uso Generar Reportes.

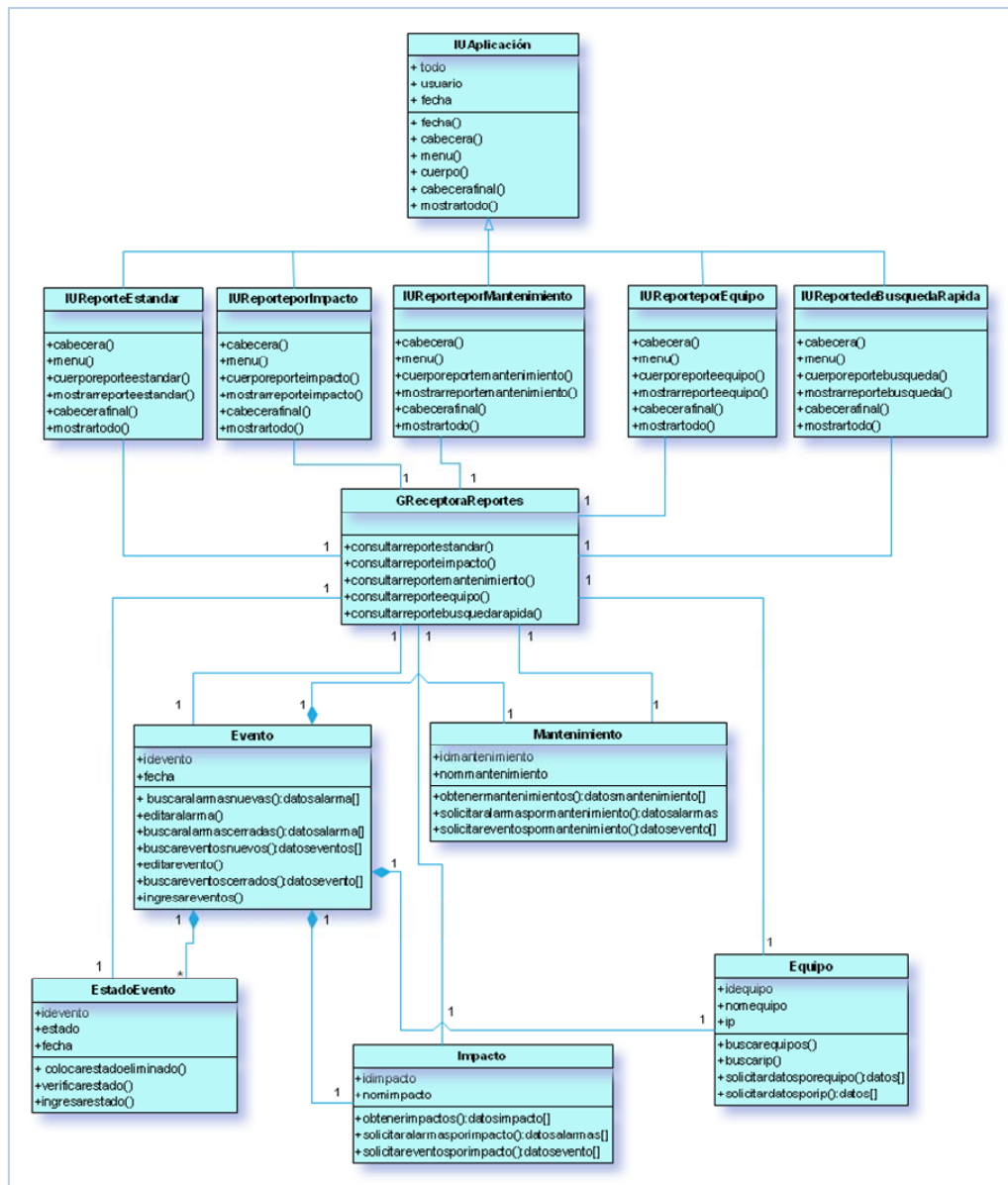


Figura 4.18. Diagrama de Clase de Diseño Generar Reportes

Fuente: Elaboración Propia.

4.3.2. Diseño de la arquitectura

El Sistema Integrado de Notificación y Comunicación de Alarmas (SIDENCA) está basado en el uso de páginas Web HTML y scripts de servidor. La arquitectura se divide en cuatro capas, la capa específica y la capa general de la aplicación que contempla los paquetes propios del sistema diferenciadas por lo delimitado o genérico que sea el uso de cada uno de dichos paquetes. La capa intermedia que comprende el comportamiento lógico del sistema, en esta capa se encuentra el navegador web que para poder procesar las peticiones del cliente utiliza el subsistema servidor Web que consiste del servidor Apache, y este a su vez usa el motor de scripts PHP para generar las páginas web mediante el subsistema PostgreSQL que es el encargado del manejo de la base de datos del sistema.

Finalmente tenemos la capa de software del sistema formada por el subsistema sistema operativo que sería Linux, y también formada por el subsistema TCP/IP que representa el protocolo de comunicación entre el servidor y los clientes. El diseño de la arquitectura del sistema se puede observar en la figura 4.19.

4.3.3. Diseño de la base de datos

La Base de Datos representa la fuente central de datos para todo sistema, es por esto que la integridad y fiabilidad de la misma reviste una altísima importancia para la correcta funcionalidad de las aplicaciones asociadas a la misma. Los datos que deben ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independizados de éstos, y su definición, que es la estructura de la Base de Datos, única y almacenada junto con los datos, apoyándose en un modelo, el cual permita captar las interrelaciones y restricciones existentes en el mundo real. Toda Base de Datos debe ser definida como un conjunto de datos relacionados entre sí y con un significado implícito.

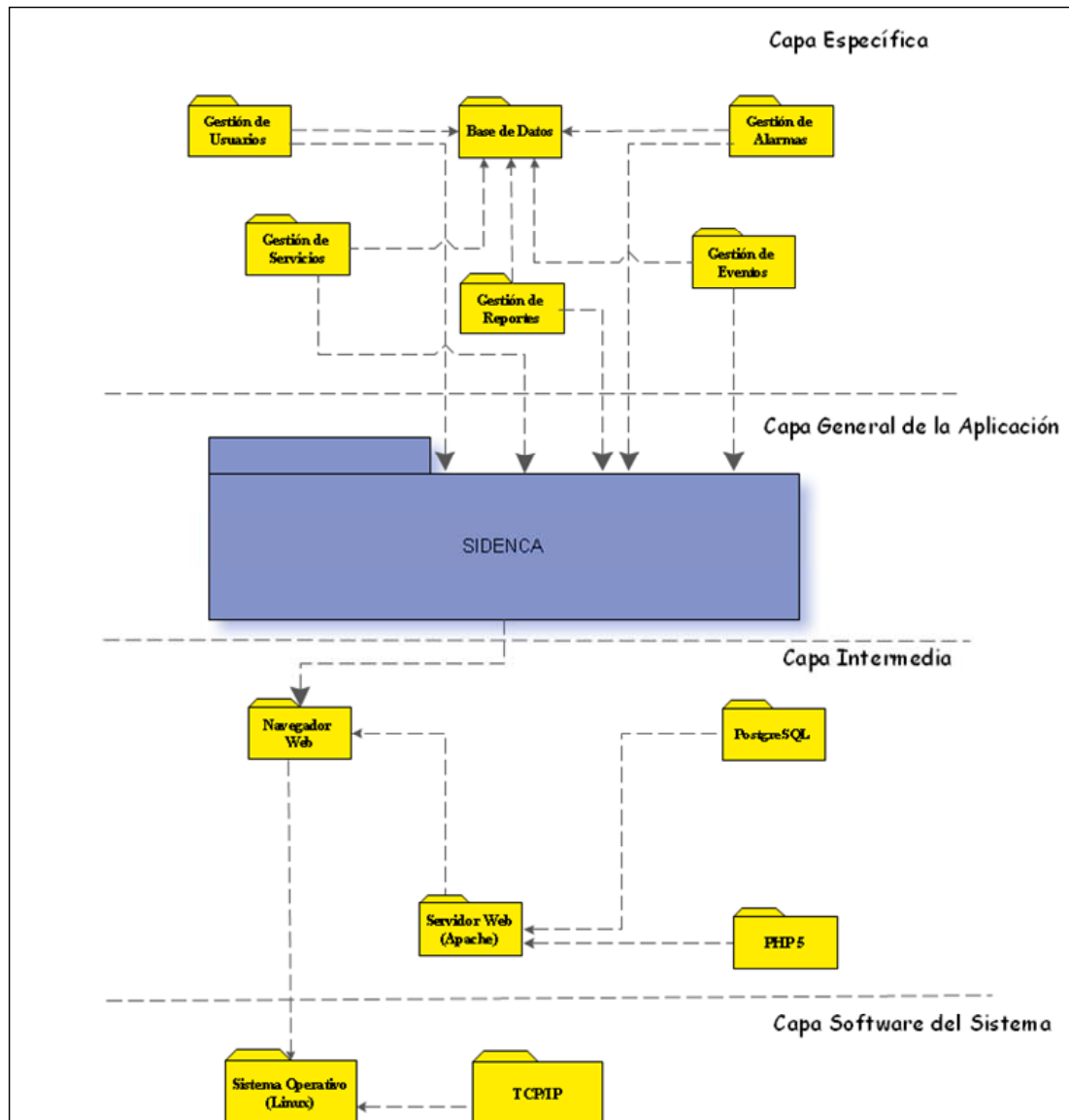


Figura 4.19. Vista lógica de la arquitectura.

Fuente: Elaboración Propia

Para el diseño de la base de datos se utilizara el modelo de datos de la especificación de WEBML y la descripción de las tablas de la base de datos.

Conociendo el volumen de datos que podrá manejar el sistema SIDENCA, se determinó el diseño de una Base de Datos relacional BDD SIDENCA para el control y gestión de los datos.

Una base de datos relacional es un conjunto de dos o más tablas estructuradas en registros (líneas) y campos (columnas), que se vinculan entre sí por un campo en común, en ambos casos posee las mismas características como por ejemplo el nombre de campo, tipo y longitud; a este campo generalmente se le denomina ID, identificador o clave.

A continuación se pueden observar las tablas diseñadas para el sistema SIDENCA con una breve descripción de las mismas.

4.3.3.1. Modelo De Datos

El modelo de datos de WebML es una adaptación de los modelos conceptuales de diseño de base datos, es compatible con el modelo de datos Entidad – Relación usado en el diseño conceptual de bases de datos, y también es compatible con los diagramas de clases UML empleados en el modelado orientado a objetos.

El modelo de datos define la estructura de la base de datos. En la figura 4.20 pueden observarse los atributos de cada entidad así como su relación con otras entidades.

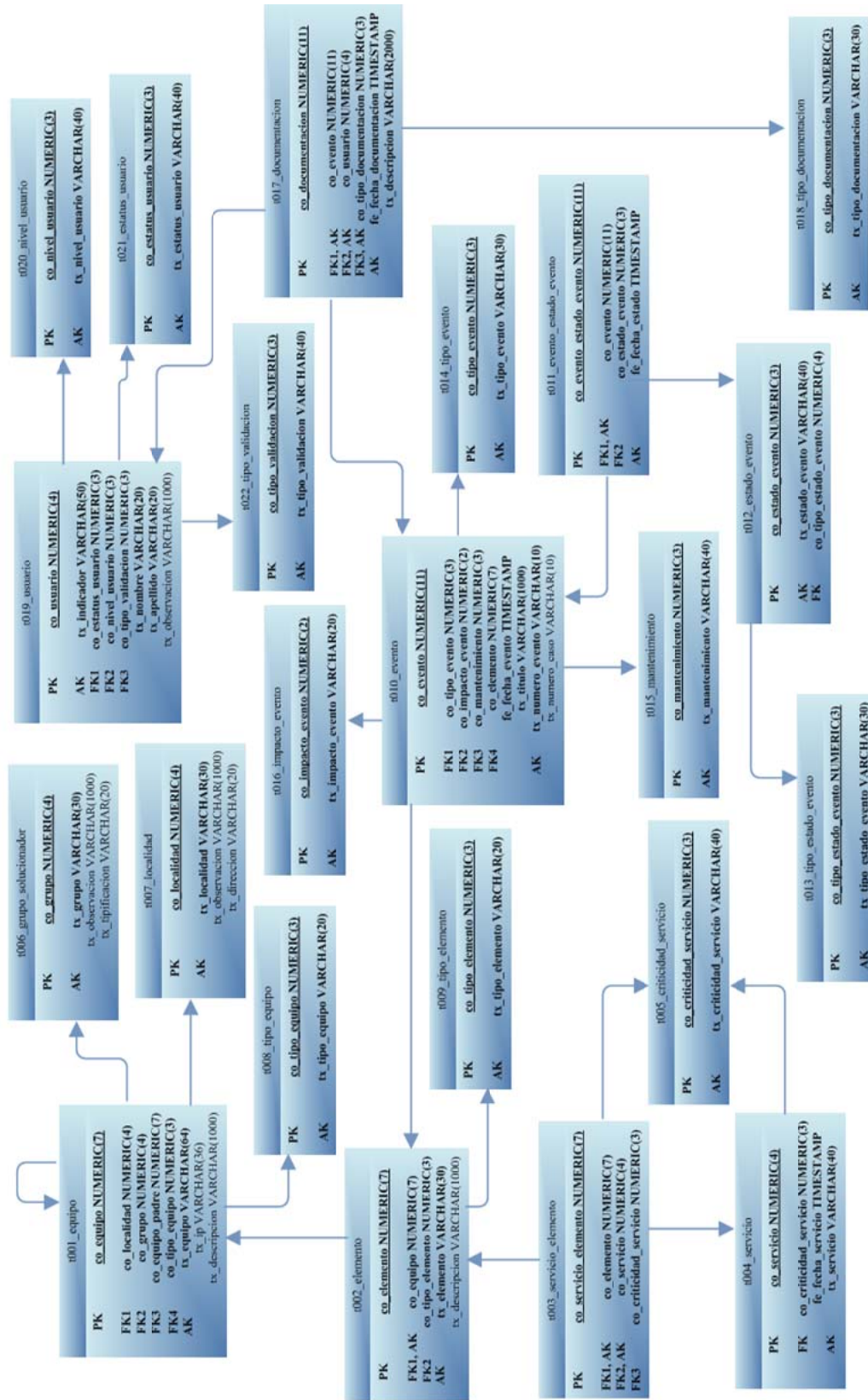


Figura 4.20. Relación de tablas de la base de datos.

Fuente: Elaboración Propia.

4.3.3.2. Estructura de la Base de Datos

Partiendo del modelo de datos y una vez normalizada las entidades, se implementó el diseño de la base de datos del sistema, a través del sistema gestor de base de datos PostgreSQL. El conjunto de tablas que conforman la base de datos serán mostradas y descritas a continuación.

Tabla Equipo: Registra todos los equipos utilizados por Servicios Comunes Oriente. Dichos equipos son almacenados por NAGIOS. Ver figura 4.21.

t001_equipo	
PK	<u>co_equipo</u> NUMERIC(7)
FK1	co_localidad NUMERIC(4)
FK2	co_grupo NUMERIC(4)
FK3	co_equipo_padre NUMERIC(7)
FK4	co_tipo_equipo NUMERIC(3)
AK	tx_equipo VARCHAR(64)
	tx_ip VARCHAR(36)
	tx_descripcion VARCHAR(1000)

Figura 4.21. Estructura de la Tabla Equipo.

Fuente: Elaboración Propia.

Tabla Elemento: Registra todos los elementos asociados a cada equipo. Ver figura 4.22

t002_elemento	
PK	<u>co_elemento</u> NUMERIC(7)
FK1, AK	co_equipo NUMERIC(7)
FK2	co_tipo_elemento NUMERIC(3)
AK	tx_elemento VARCHAR(30)
	tx_descripcion VARCHAR(1000)

Figura 4.22. Estructura de la Tabla Elemento.

Fuente: Elaboración Propia.

Tabla Servicio Elemento: Registra todos los diferentes elementos asociados a un servicio. Ver figura 4.23

t003_servicio_elemento	
PK	<u>co_servicio</u> NUMERIC(7)
FK1, AK	co_elemento NUMERIC(7)
FK2, AK	co_servicio NUMERIC(4)
FK3	co_criticidad_servicio NUMERIC(3)

Figura 4.23. Estructura de la Tabla Servicio Elemento.

Fuente: Elaboración Propia

Tabla Servicio: Registra todos los datos de los servicios utilizados por la Gerencia de AIT Servicios Comunes Oriente. Ver figura 4.24

t004_servicio	
PK	<u>co_servicio</u> NUMERIC(4)
FK	co_criticidad_servicio NUMERIC(3)
	fe_fecha_servicio TIMESTAMP
AK	tx_servicio VARCHAR(40)

Figura 4.24. Estructura de la Tabla Servicio.

Fuente: Elaboración Propia.

Tabla Criticidad Servicio: Registra los tipos de criticidad asociados a cada servicio y a cada elemento. Ver figura 4.25

t005_criticidad_servicio	
PK	<u>co_criticidad_servicio</u> NUMERIC(3)
AK	tx_criticidad_servicio VARCHAR(40)

Figura 4.25. Estructura de la Tabla Criticidad Servicio.

Fuente: Elaboración Propia.

Tabla Grupo Solucionador: Registra la información correspondiente a cada una de las disciplinas que integran el departamento de Mantenimiento de la Plataforma de PDVSA Servicios Comunes Oriente. Ver figura 4.26

t006_grupo_solucionador	
PK	<u>co_grupo</u> NUMERIC(4)
AK	tx_grupo VARCHAR(30) tx_observacion VARCHAR(1000) tx_tipificacion VARCHAR(20)

Figura 4.26. Estructura de la Tabla Grupo Solucionador.

Fuente: Elaboración Propia.

Tabla Localidad: Registra los datos de las localidades que integran a PDVSA Servicios Comunes Oriente. Ver figura 4.27.

t007_localidad	
PK	<u>co_localidad</u> NUMERIC(4)
AK	tx_localidad VARCHAR(30) tx_observacion VARCHAR(1000) tx_direccion VARCHAR(20)

Figura 4.27. Estructura de la Tabla Localidad.

Fuente: Elaboración Propia.

Tabla Tipo Equipo: Registra todas las categorías de tipo equipo asociadas a cada equipo. Ver figura 4.28

t008_tipo_equipo	
PK	<u>co_tipo_equipo</u> NUMERIC(3)
AK	tx_tipo_equipo VARCHAR(20)

Figura 4.28. Estructura de la Tabla Tipo Equipo.

Fuente: Elaboración Propia.

Tabla Tipo Elemento: Registra todas las categorías de tipo elemento asociadas a cada elemento. Ver figura 4.29.

t009_tipo_elemento	
PK	<u>co_tipo_elemento</u> NUMERIC(3)
AK	tx_tipo_elemento VARCHAR(20)

Figura 4.29. Estructura de la Tabla Tipo Elemento.

Fuente: Elaboración Propia.

Tabla Evento: Registra todos los datos referentes a las alarmas generadas por NAGIOS y los eventos ingresados por los analistas. Ver figura 4.30.

t010_evento	
PK	<u>co_evento</u> NUMERIC(11)
FK1	co_tipo_evento NUMERIC(3)
FK2	co_impacto_evento NUMERIC(2)
FK3	co_mantenimiento NUMERIC(3)
FK4	co_elemento NUMERIC(7)
	fe_fecha_evento TIMESTAMP
	tx_titulo VARCHAR(1000)
AK	tx_numero_evento VARCHAR(10)
	tx_numero_caso VARCHAR(10)

Figura 4.30. Estructura de la Tabla Evento.

Fuente: Elaboración Propia.

Tabla Evento Estado Evento: Registra la relación de los estados asociados a cada alarma y evento. Ver figura 4.31.

t011_evento_estado_evento	
PK	<u>co_evento_estado_evento</u> NUMERIC(11)
FK1, AK	co_evento NUMERIC(11)
FK2	co_estado_evento NUMERIC(3)
AK	fe_fecha_estado TIMESTAMP

Figura 4.31. Estructura de la Tabla Evento Estado Evento.

Fuente: Elaboración Propia.

Tabla Estado Evento: Registra la relación entre los estados y su correspondiente categoría en cuanto al sistema donde se almacena la falla. Ver figura 4.32.

t012_estado_evento	
PK	<u>co_estado_evento</u> NUMERIC(3)
AK	tx_estado_evento VARCHAR(40)
FK	co_tipo_estado_evento NUMERIC(4)

Figura 4.32. Estructura de la Tabla Estado Evento.

Fuente: Elaboración Propia.

Tabla Tipo Estado Evento: Registra los diferentes sistemas de almacenamiento asociados a cada falla. Ver figura 4.33.

t013_tipo_estado_evento	
PK	<u>co_tipo_estado_evento</u> NUMERIC(3)
AK	tx_tipo_estado_evento VARCHAR(30)

Figura 4.33. Estructura de la Tabla Tipo Estado Evento.

Fuente: Elaboración Propia.

Tabla Tipo Evento: Registra los tipos de eventos y alarmas. Ver figura 4.34.

t014_tipo_evento	
PK	<u>co_tipo_evento</u> NUMERIC(3)
AK	tx_tipo_evento VARCHAR(30)

Figura 4.34. Estructura de la Tabla Tipo Evento.

Fuente: Elaboración Propia.

Tabla Mantenimiento: Registra todos los tipos de mantenimientos, manejados por la Gerencia de AIT Servicios Comunes Oriente. Ver figura 4.35.

t015_mantenimiento	
PK	<u>co_mantenimiento</u> NUMERIC(3)
AK	tx_mantenimiento VARCHAR(40)

Figura 4.35. Estructura de la Tabla Mantenimiento.

Fuente: Elaboración Propia.

Tabla Impacto Evento: Registra todos los tipos de impactos manejados por la Gerencia de AIT Servicios Comunes Oriente. Ver Figura 4.36.

t016_impacto_evento	
PK	<u>co_impacto_evento</u> NUMERIC(2)
AK	tx_impacto_evento VARCHAR(20)

Figura 4.36. Estructura de la Tabla Impacto Evento.

Fuente: Elaboración Propia.

Tabla Documentación: Registra todos los datos asociados a las diferentes documentaciones registradas de cada alarma y evento y sus respectivas descripciones. Ver figura 4.37.

t017_documentacion	
PK	<u>co_documentacion</u> NUMERIC(11)
FK1, AK	co_evento NUMERIC(11)
FK2, AK	co_usuario NUMERIC(4)
FK3, AK	co_tipo_documentacion NUMERIC(3)
AK	fe_fecha_documentacion TIMESTAMP
	tx_descripcion VARCHAR(2000)

Figura 4.37. Estructura de la Tabla Documentación.

Fuente: Elaboración Propia.

Tabla Tipo Documentación: Registra los tipos de documentación que se deben realizar para documentar las alarmas y los eventos. Ver figura 4.38.

t018_tipo_documentacion	
PK	<u>co_tipo_documentacion</u> NUMERIC(3)
AK	tx_tipo_documentacion VARCHAR(30)

Figura 4.38. Estructura de la Tabla Tipo Documentación.

Fuente: Elaboración Propia.

Tabla Usuario: Registra el detalle de todos los datos que corresponde a los usuarios registrados. Ver figura 4.39.

t019_usuario	
PK	<u>co_usuario</u> NUMERIC(4)
AK	tx_indicador VARCHAR(50)
FK1	co_estatus_usuario NUMERIC(3)
FK2	co_nivel_usuario NUMERIC(3)
FK3	co_tipo_validacion NUMERIC(3)
	tx_nombre VARCHAR(20)
	tx_apellido VARCHAR(20)
	tx_observacion VARCHAR(1000)

Figura 4.39. Estructura de la Tabla Usuario.

Fuente: Elaboración Propia.

Tabla Nivel Usuario: Registra los diferentes roles de usuario con los que cuenta el sistema. Ver figura 4.40.

t020_nivel_usuario	
PK	<u>co_nivel_usuario</u> NUMERIC(3)
AK	tx_nivel_usuario VARCHAR(40)

Figura 4.40. Estructura de la Tabla Nivel Usuario.

Fuente: Elaboración Propia.

Tabla Estatus Usuario: Registra los diferentes tipos de estatus para cada usuario. Ver figura 4.41.

t021_estatus_usuario	
PK	<u>co_estatus_usuario</u> NUMERIC(3)
AK	tx_estatus_usuario VARCHAR(40)

Figura 4.41. Estructura de la Tabla Estatus Usuario.

Fuente: Elaboración Propia.

Tabla Tipo Validación: Registra los tipos de validación asignados a cada usuario, para verificar el sistema por el cual se debe validar su entrada a la aplicación. Ver figura 4.42.

t022_tipo_validacion	
PK	<u>co_tipo_validacion</u> NUMERIC(3)
AK	tx_tipo_validacion VARCHAR(40)

Figura 4.42. Estructura de la Tabla Tipo Validación.

Fuente: Elaboración Propia.

4.3.4. Modelo De Hipertexto y Modelo De Presentación

El modelo de hipertexto especifica cómo se compone el software en función de las páginas que forman la vista para los usuarios, y de la navegación por cada una de éstas, mientras que el modelo de presentación describe como lucirá la vista del sistema, es decir, el diseño de cómo será cada página que forma una vista.

Ya que se tiene un conocimiento completo del funcionamiento del sistema que se va a construir y de la base de datos que lo conforma, se procede a diseñar las vistas (site views) que complementan la arquitectura del sistema, así como el diseño de las página que conformarán cada vista.

4.3.4.1. Vista Administrador del Sistema

La vista del administrador del sistema comprende las áreas de validar usuarios, consultar alarmas, alarmas cerradas, configuración, ingresar eventos, consultar eventos, eventos cerrados, servicios y reportes, dentro de las cuales se encuentran las diferentes páginas que permiten realizar las diversas acciones dentro de cada una de ellas. Esta vista es una de las más importantes del sistema ya que se realizan todas las acciones, pero a pesar de esto hay ciertas páginas que son compartidas con otros grupos de usuarios. La figura 4.43 muestra la vista de administrador del sistema.

A continuación se muestra la tabla 4.10 con la descripción de la vista y más adelante se mostrará el diseño de las páginas que forman parte de ella.

Tabla 4.10. Descripción de la vista administrador del sistema.

Vista	Vista Administrador del Sistema
Descripción	Permite realizar todas las acciones que comprende la administración de los usuarios, la gestión de las alarmas y sus diferentes reportes, así como también otras operaciones propias del sistema.
Actores	Administrador del Sistema.
Casos Uso	Administrar Usuarios, Gestionar Alarmas, Gestionar Eventos, Gestionar Servicios, Generar Reportes.

Fuente: Elaboración Propia

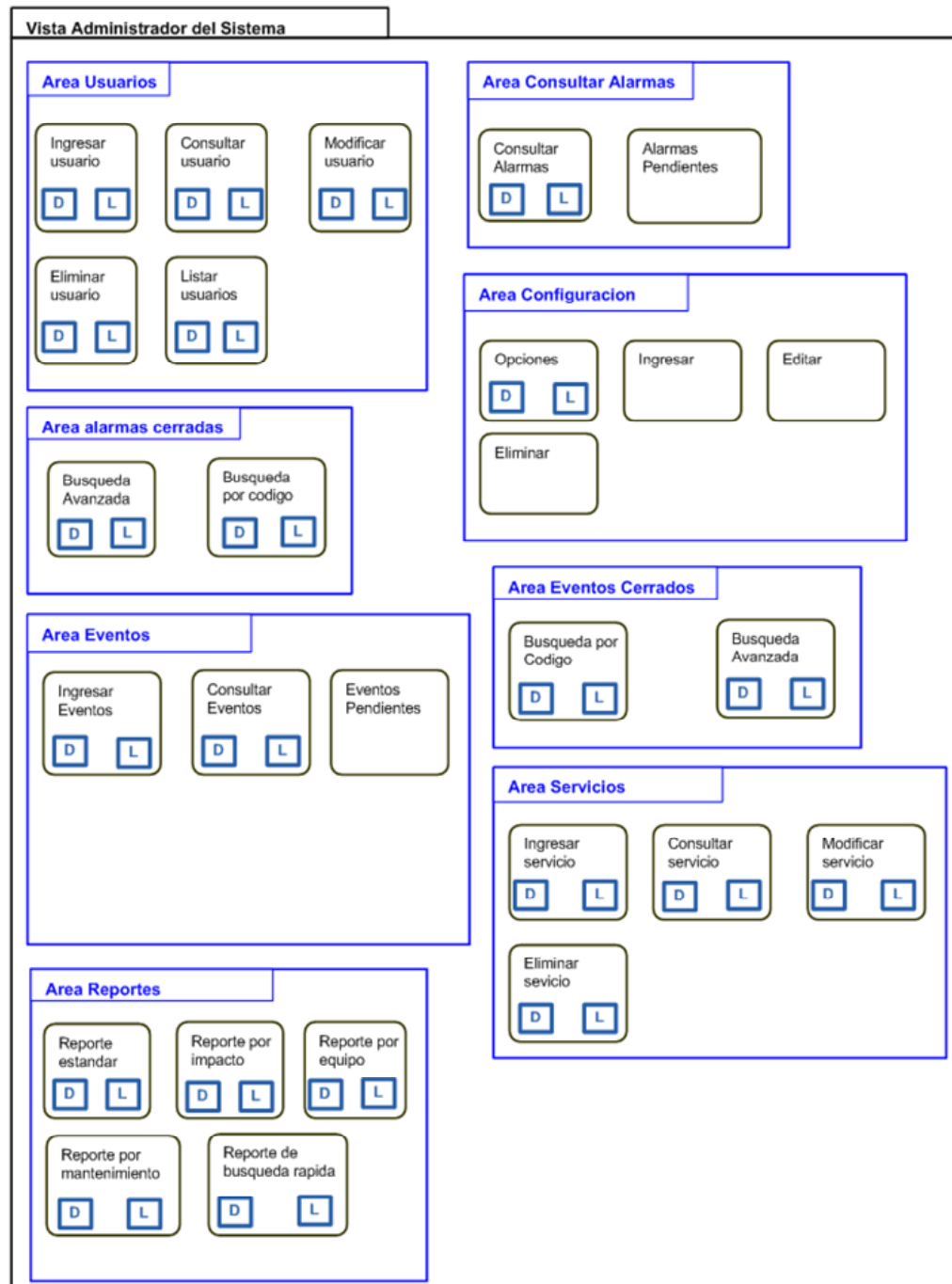


Figura 4.43 Vista de Administrador del Sistema.

Fuente: Elaboración Propia.

Área Usuarios

La página ingresar usuario mostrada en la figura 4.44 definida dentro del área de usuarios permite registrar los datos de un nuevo usuario al sistema.

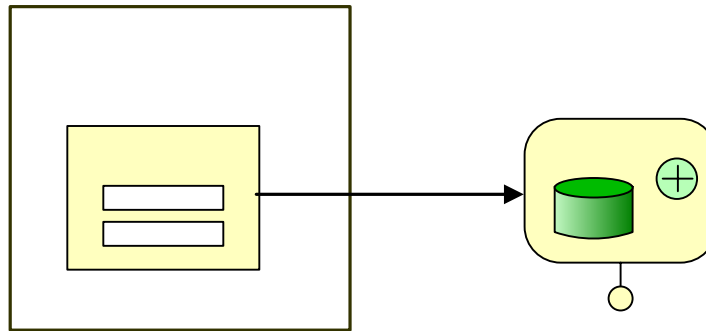


Figura 4.44. Página Ingresar Usuario.

Fuente: Elaboración propia.

La página consultar usuario mostrada en la figura 4.45 permite ingresar el indicador de un usuario, a partir del cual se realiza la búsqueda de los datos correspondientes a éste.

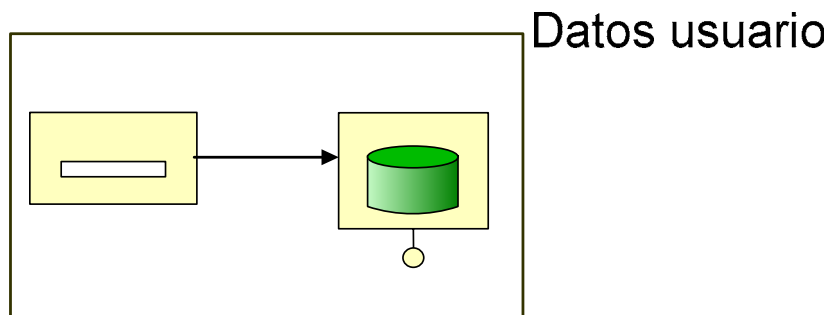


Figura 4.45. Página Consultar Usuario.

Fuente: Elaboración propia.

La página modificar usuario mostrada en la figura 4.46 permite actualizar los datos de un usuario, ingresando el indicador del mismo.

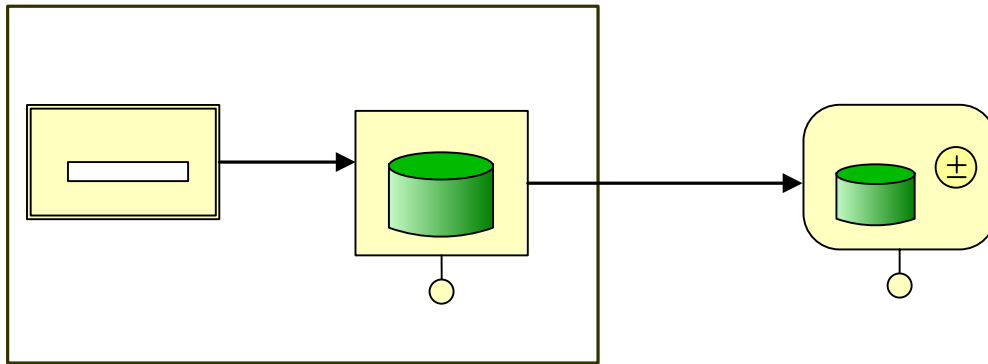


Figura 4.46. Página Modificar Usuario.

Fuente: Elaboración propia.

La figura 4.47 muestra la página eliminar usuario la cual permite retirar un usuario del sistema o bien cambiar el estatus del mismo en caso de este tenga asignada alguna documentación, a partir de un formulario en el cual se ingresa el indicador del mismo, y el sistema realiza la búsqueda correspondiente.

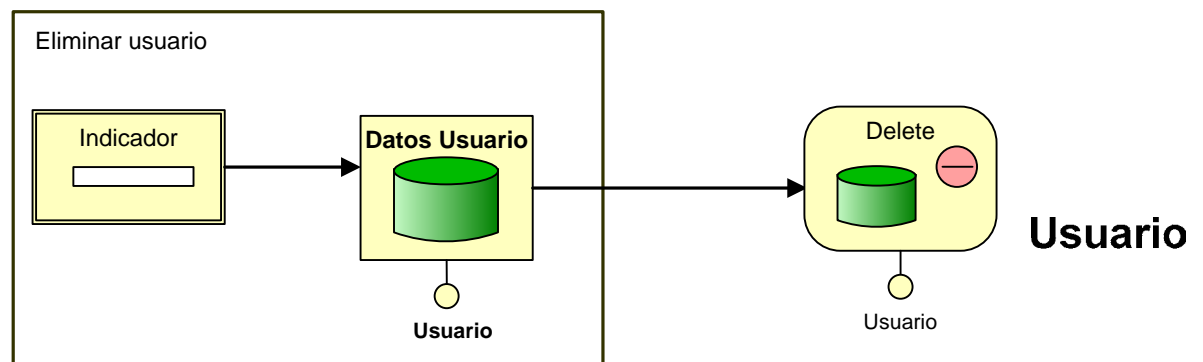


Figura 4.47. Página Eliminar Usuario.

Fuente: Elaboración propia.

La figura 4.48 muestra la página listar usuarios que permite visualizar los datos de todos los usuarios registrados en el sistema.

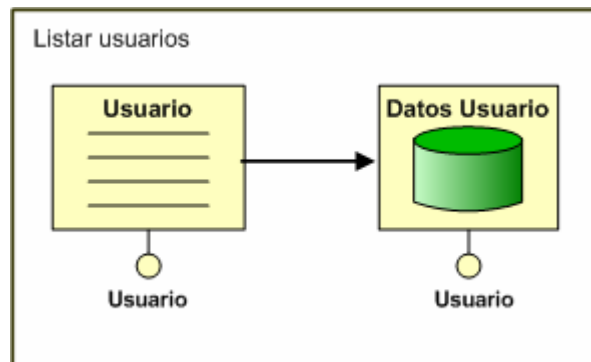


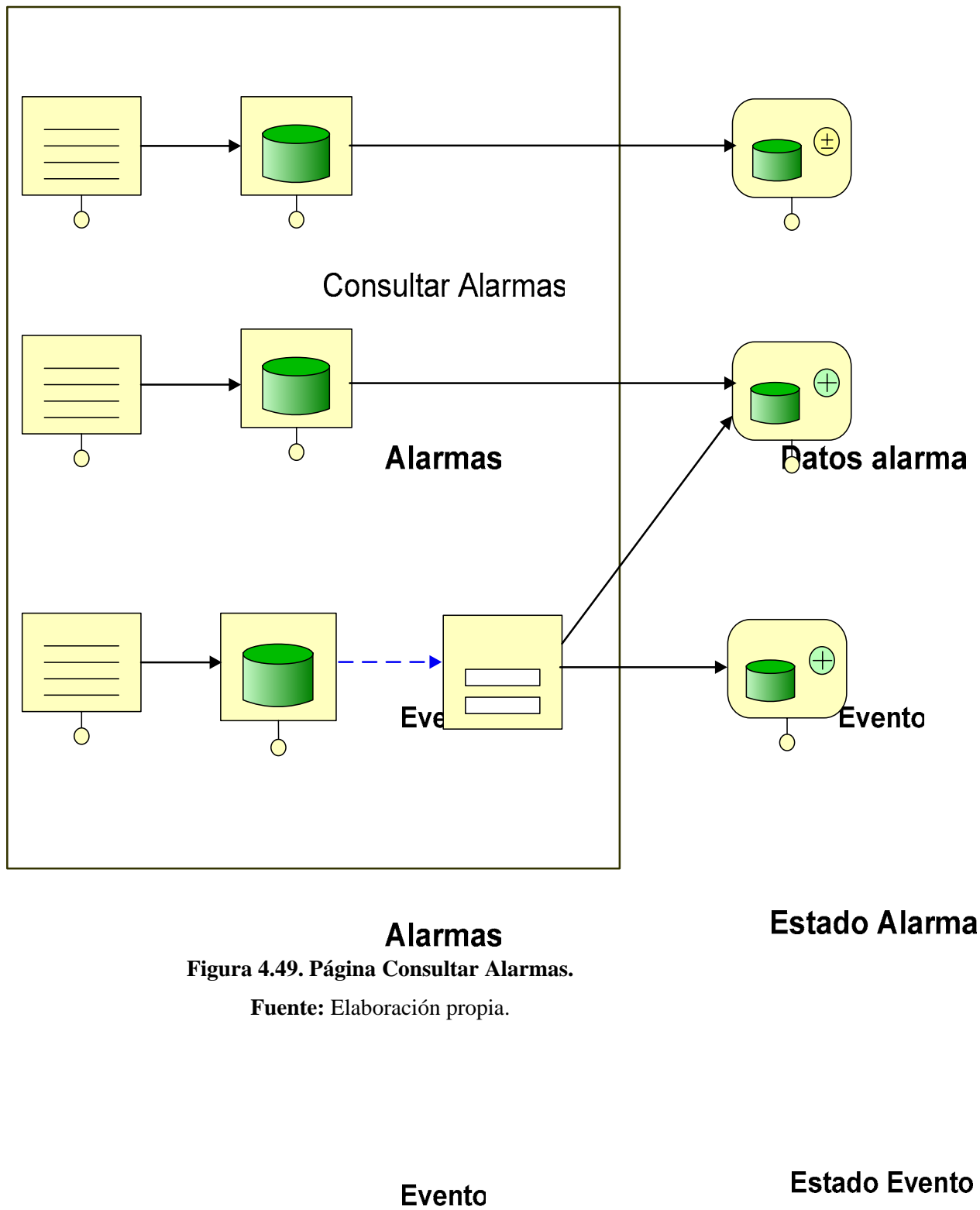
Figura 4.48. Página Listar Usuarios.

Fuente: Elaboración propia.

Área Consultar Alarmas

La página consultar alarmas mostrada en la figura 4.49 permite visualizar las alarmas generadas por NAGIOS, y a su vez seleccionar una de ellas y realizar diferentes acciones como: editar, donde se muestra un formulario con los datos de la alarma y permite al usuario actualizarlos, eliminar, seleccionar una alarma y el sistema cambiara el estado de la misma y documentar, a partir de la selección de una alarma se muestra la documentación de ésta y permite al usuario cambiar el estado y agregar más documentación del caso.

En la figura 4.50 se muestra la página alarmas pendientes, que muestra las alarmas a las que se les ha asignado un estado pendiente. Esta página permite al usuario seleccionar una de las alarmas mostradas para editar sus datos o documentarla.



Alarmas
Figura 4.49. Página Consultar Alarmas.
Fuente: Elaboración propia.

Estado Alarma

Evento

Estado Evento

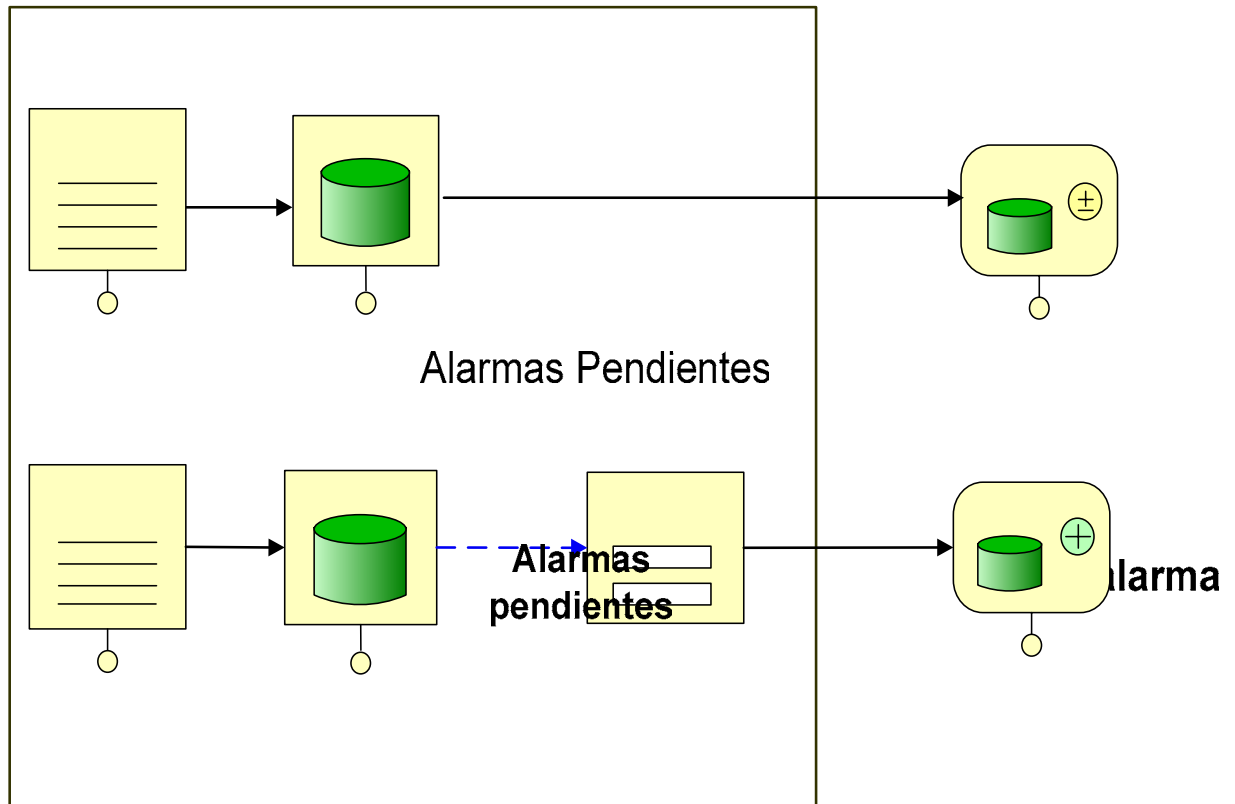


Figura 4.50. Página Alarmas Pendientes.

Fuente: Elaboración propia.

Evento

Evento

Área Alarmas Cerradas

La página búsqueda avanzada mostrada en la figura 4.51, permite visualizar las últimas diez alarmas generadas por NAGIOS que ya hayan sido documentadas y cerradas, y a su vez permite seleccionar una de ellas y realizar diferentes acciones como: editar, eliminar o documentar. En el caso de eliminar una alarma solo se retira de la visualización ya que en el sistema debe quedar un historial de la misma en caso de que el administrador del sistema desee consultarla en un futuro.

Alarmas
pendientes

Documentacion

Evento

Documentacion

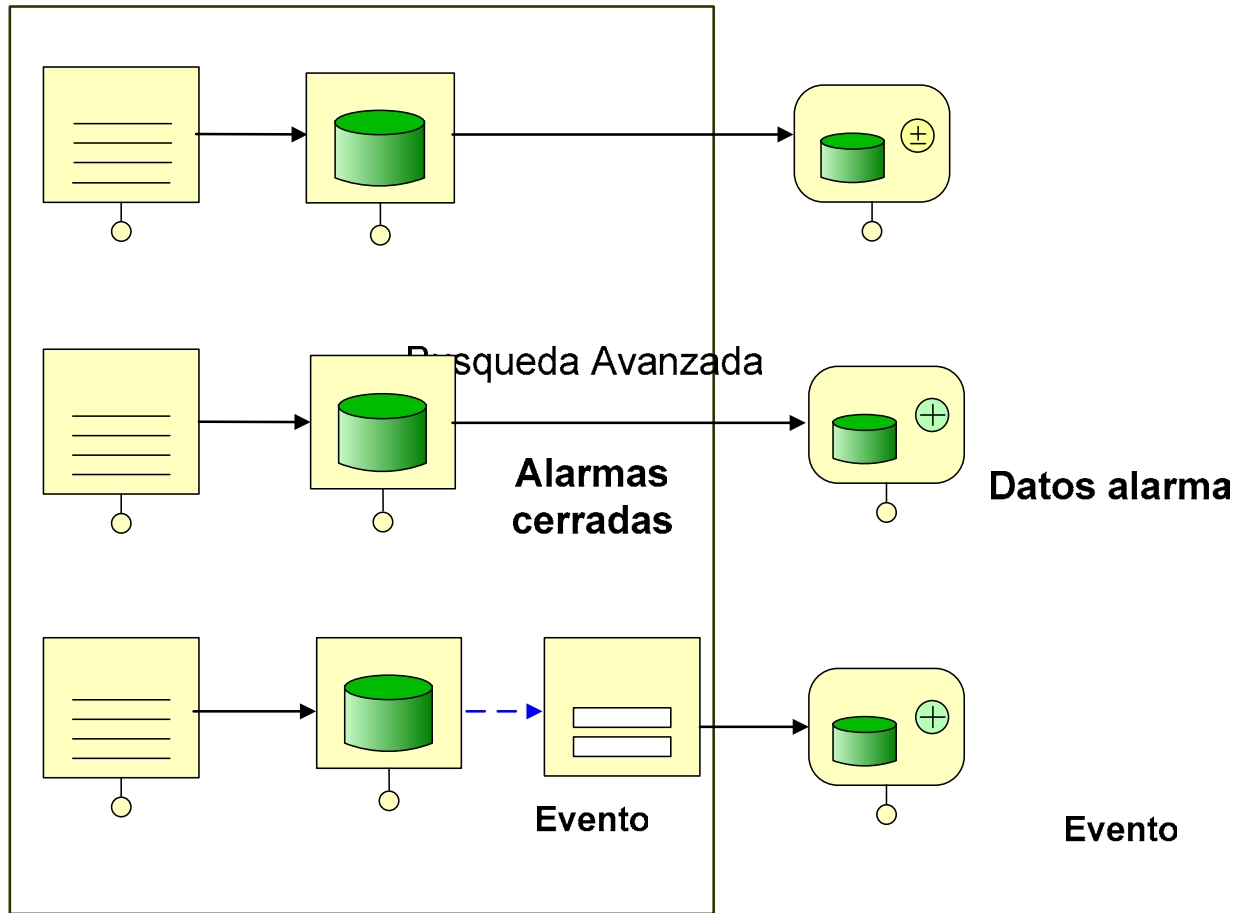


Figura 4.51. Página Búsqueda Avanzada.

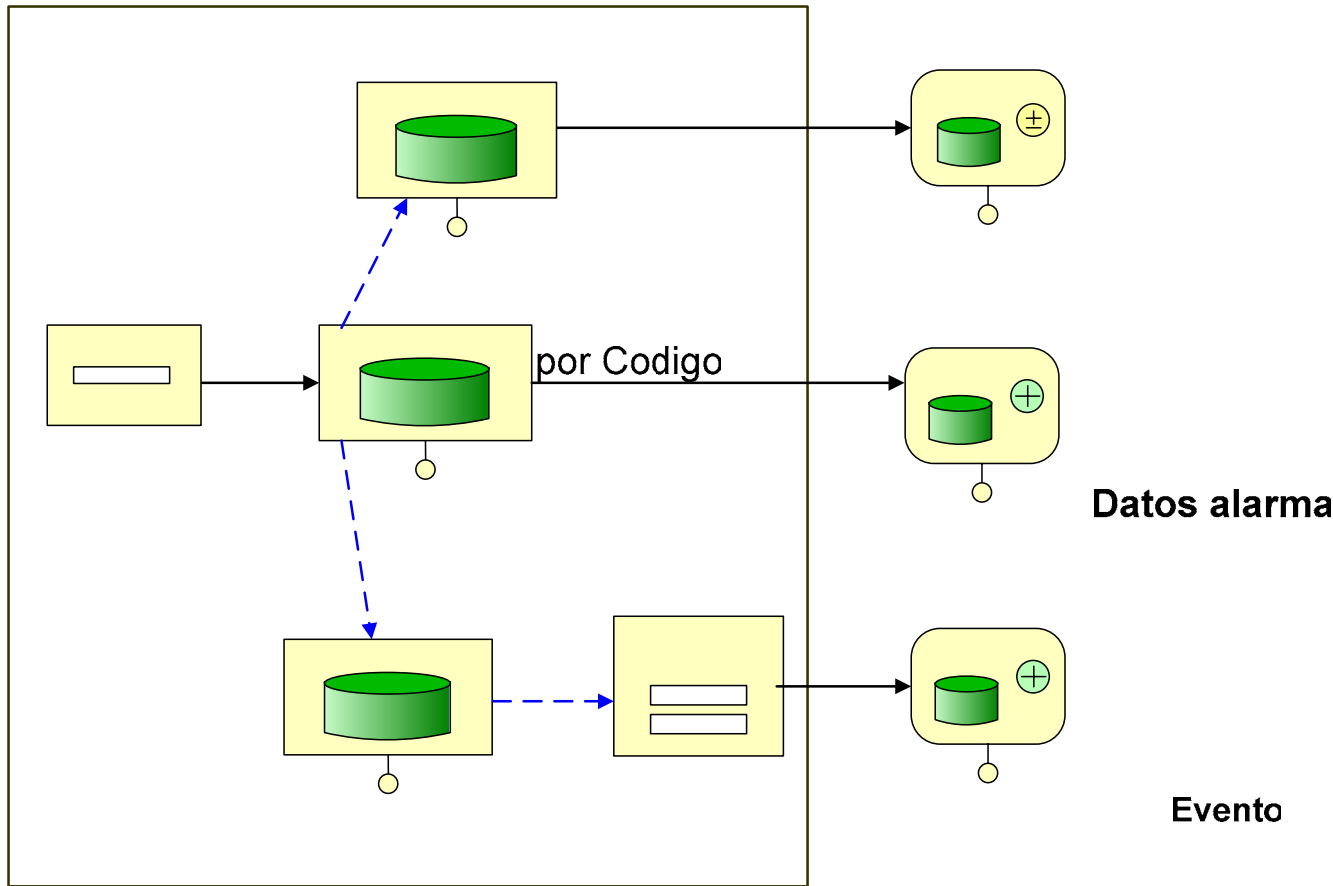
Fuente: Elaboración propia.

La página mostrada en la figura 4.52 se deberá ingresar el número de caso de la alarma a buscar, luego se desplegará toda la información referente a la alarma y las diferentes acciones que se pueden realizar como: editar la alarma, eliminarla o seguir documentando el caso después de haberlo cerrado. Para eliminar una alarma se sigue el mismo procedimiento explicado anteriormente para la figura 4.42.

Alarmas cerradas **Estado alarma**

Evento **Estado Evento**

Alarmas cerradas **Datos alarma**



Numero de caso
 Figura 4.52. Página Búsqueda por Código.

Fuente: Elaboración propia.

Área Configuración

La página mostrada en la figura 4.53 permite ingresar cualquiera de la diferentes opciones mostradas en la página, como: localidad, disciplina, tipo equipo y tipo elemento.

Datos documentacion

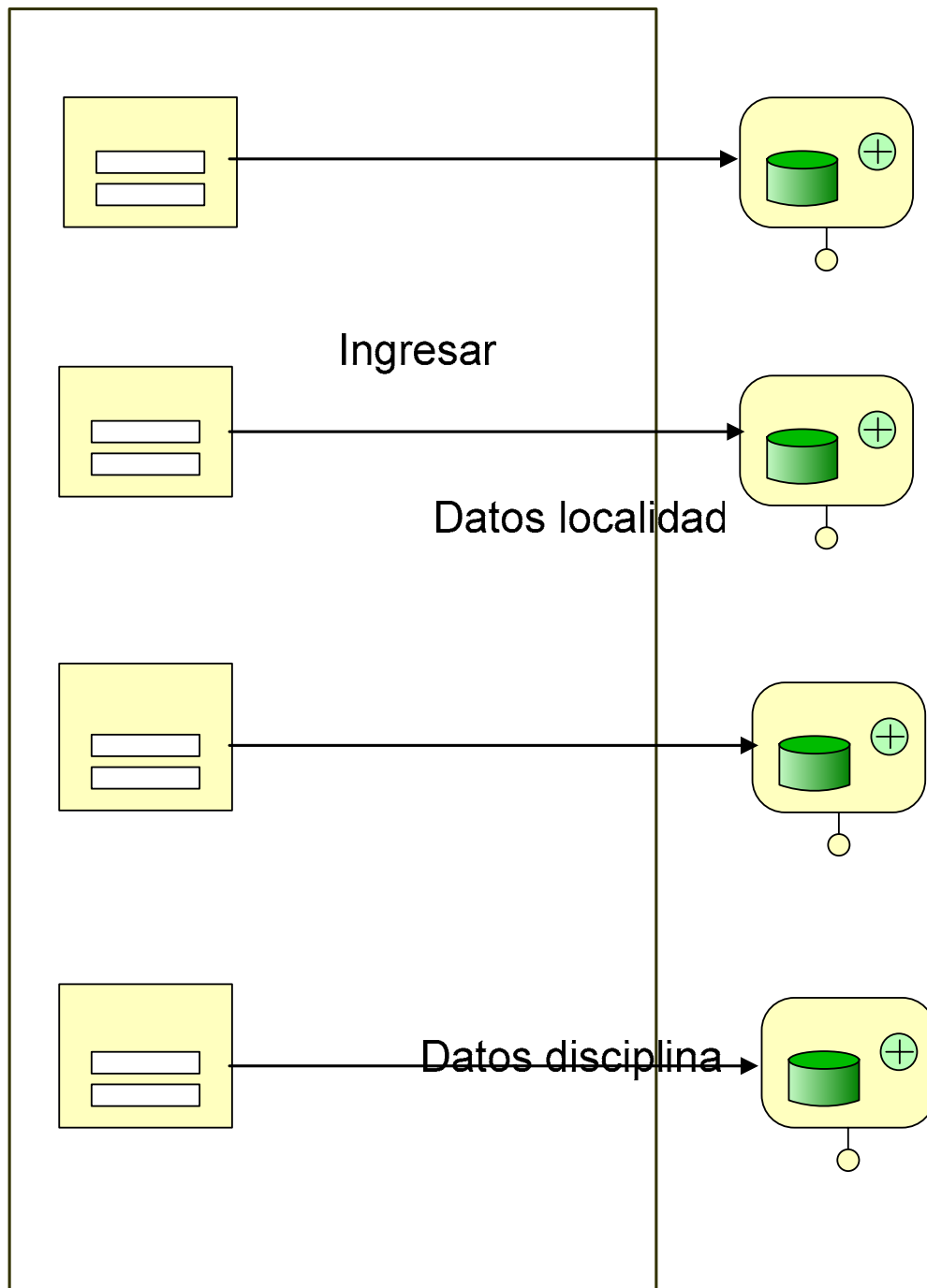


Figura 4.53. Página Ingresar.

Fuente: Elaboración propia.

Datos tipo
equipo

La página editar mostrada en la figura 4.54 permite actualizar una de las opciones mostradas en la página opciones, el usuario debe seleccionar una categoría ya sea por localidades, disciplinas, tipo equipos o tipo elementos.

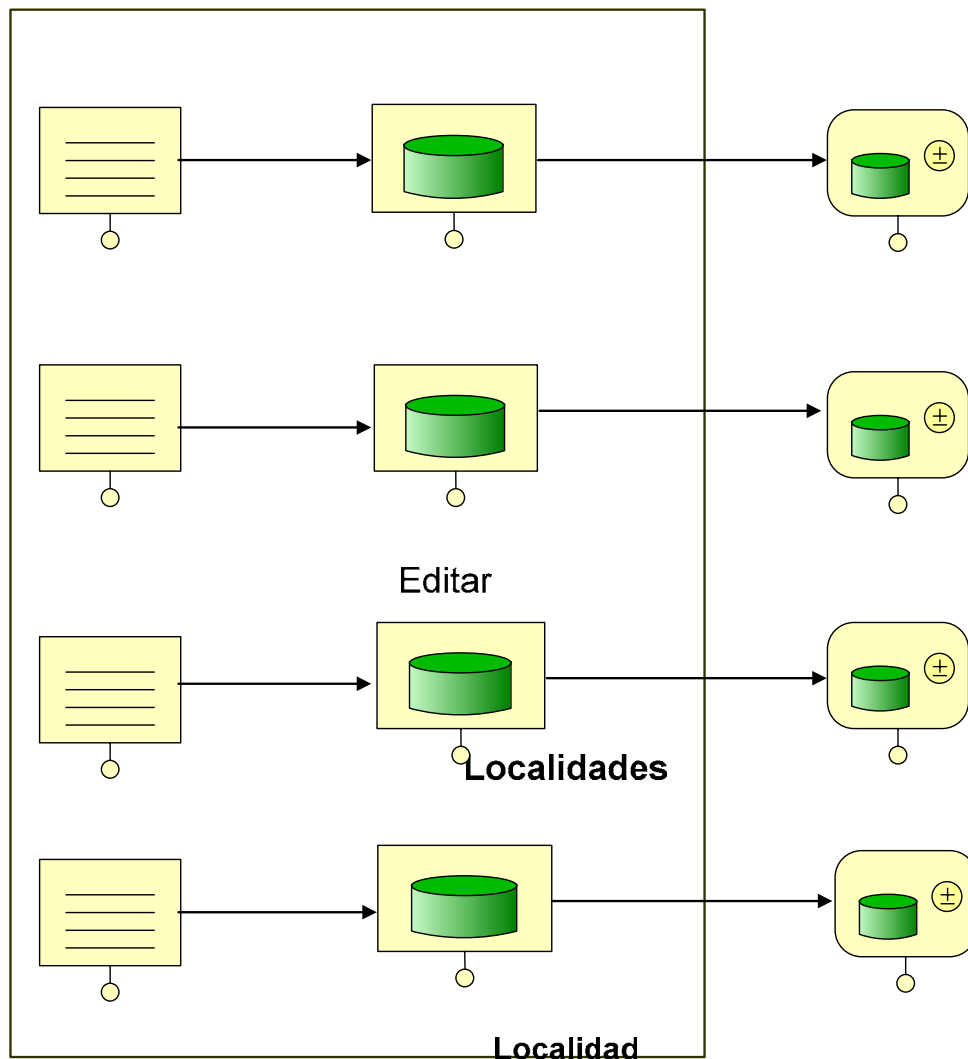


Figura 4.54. Página Editar.

Fuente: Elaboración propia.

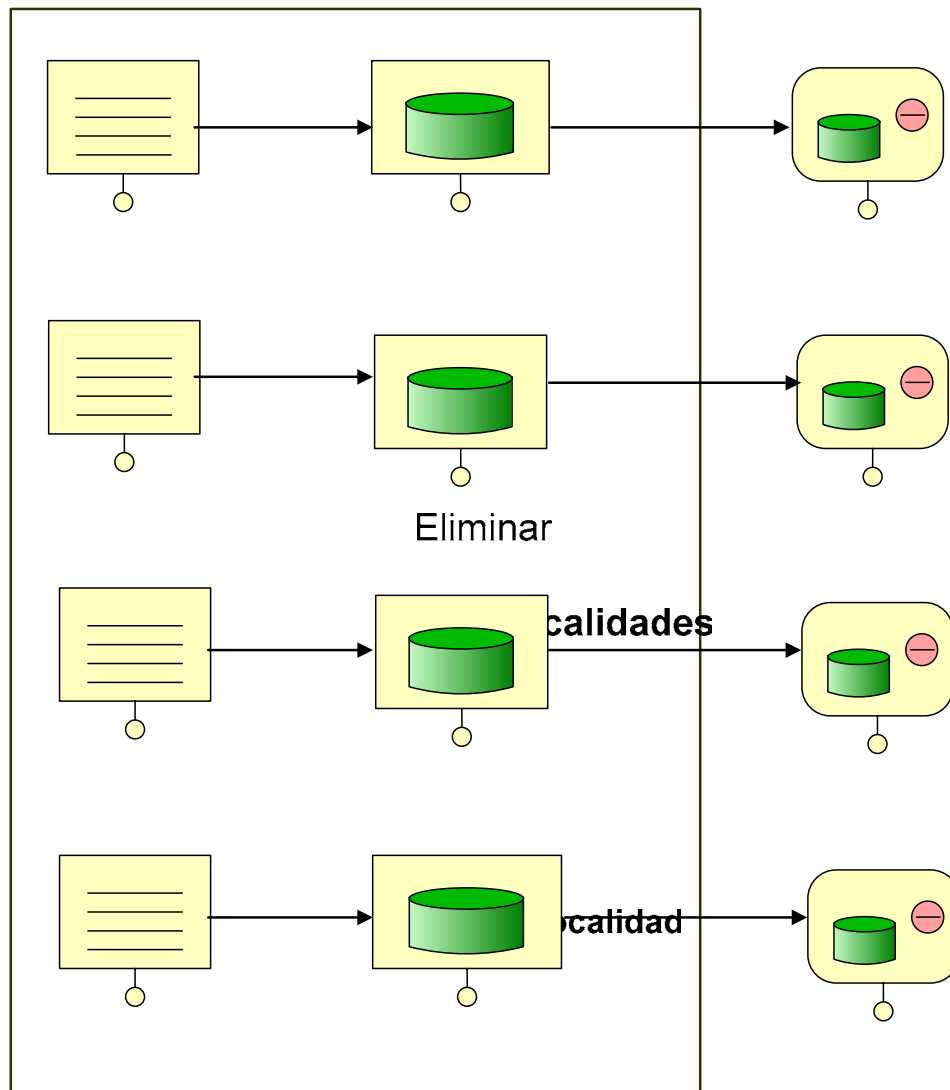
Disciplinas

Datos g

Grupo solucionador

Gupo solu

La figura 4.55 mostrada a continuación representa la pagina eliminar, la cual permite seleccionar una de las opciones mostradas y eliminar sus datos del sistema.



Disciplinas
Figura 4.55. Página Eliminar.

Fuente: Elaboración propia.

Grupo solucionador

Datos l

Loc

Datos

Grupo so

Área Eventos

La página ingresar eventos mostrada en la figura 4.56 permite ingresar los eventos al sistema. Se debe elegir un tipo de evento a ingresar, a partir de allí se generará el formulario correspondiente a dicho evento, dicho formulario consta de datos generales y la documentación de la falla.

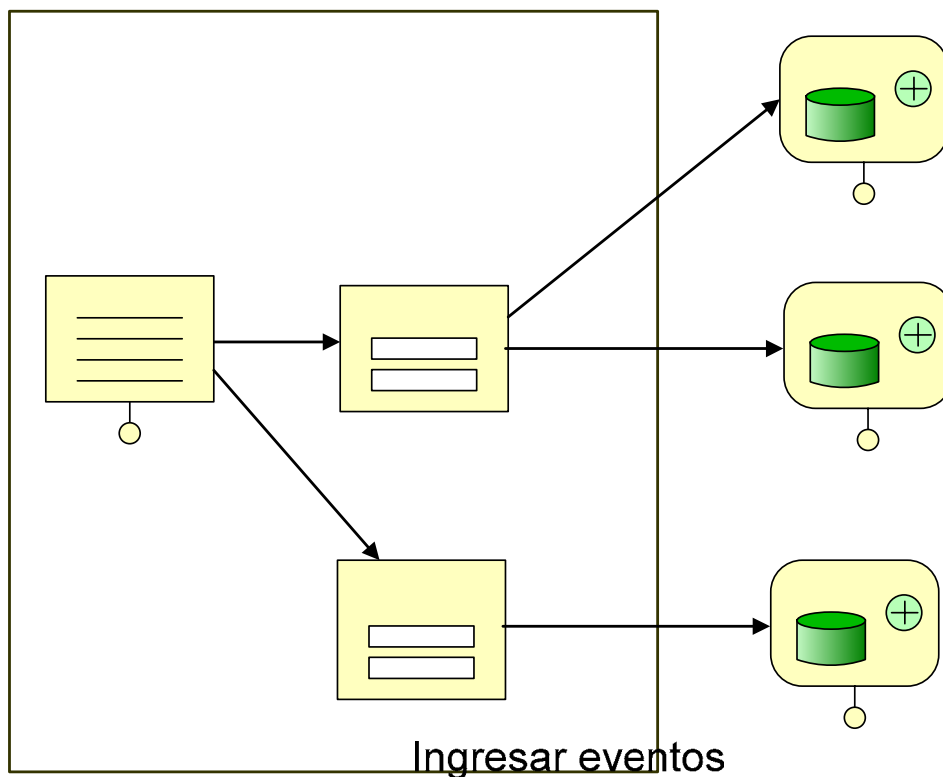


Figura 4.56. Página Ingresar Eventos.

Fuente: Elaboración propia.

La página consultar eventos mostrada en la figura 4.57 permite visualizar los eventos ingresados, y a su vez seleccionar uno de ellos y realizar diferentes acciones como: editar, donde se muestra un formulario con los datos del evento y permite al

Tipo evento

Datos gener

usuario actualizarlos, eliminar, permite al usuario seleccionar un evento y el sistema cambiará el estado del mismo y documentar, a partir de la selección de un evento se muestra la documentación de éste y permite al usuario cambiar el estado y agregar más documentación del caso.

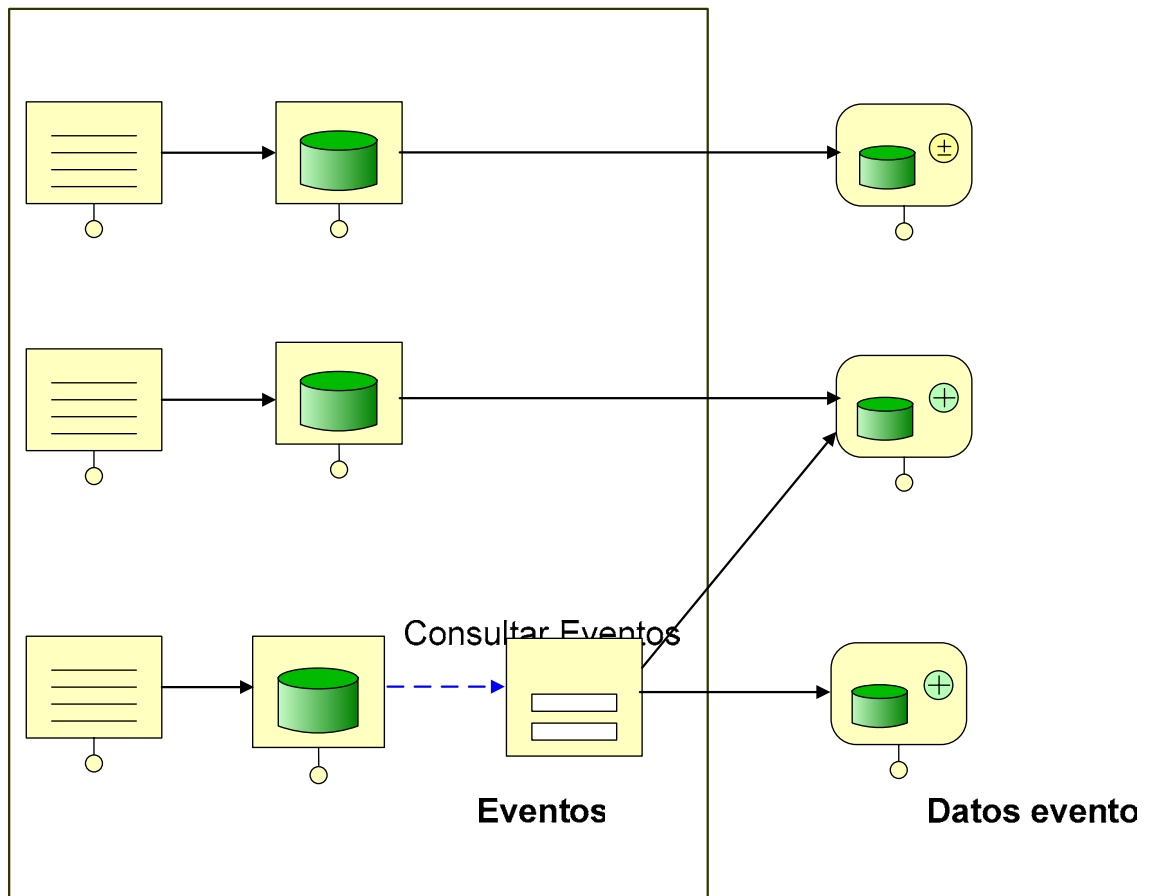


Figura 4.57. Página Consultar Eventos.

Fuente: Elaboración propia.

Evento

Evento

En la figura 4.58 se muestra la página eventos pendientes, que muestra los eventos a los que se les ha asignado un estado pendiente. Esta página permite al usuario seleccionar uno de los eventos mostrados, para editar sus datos o documentar el caso.

Eventos

Datos evento

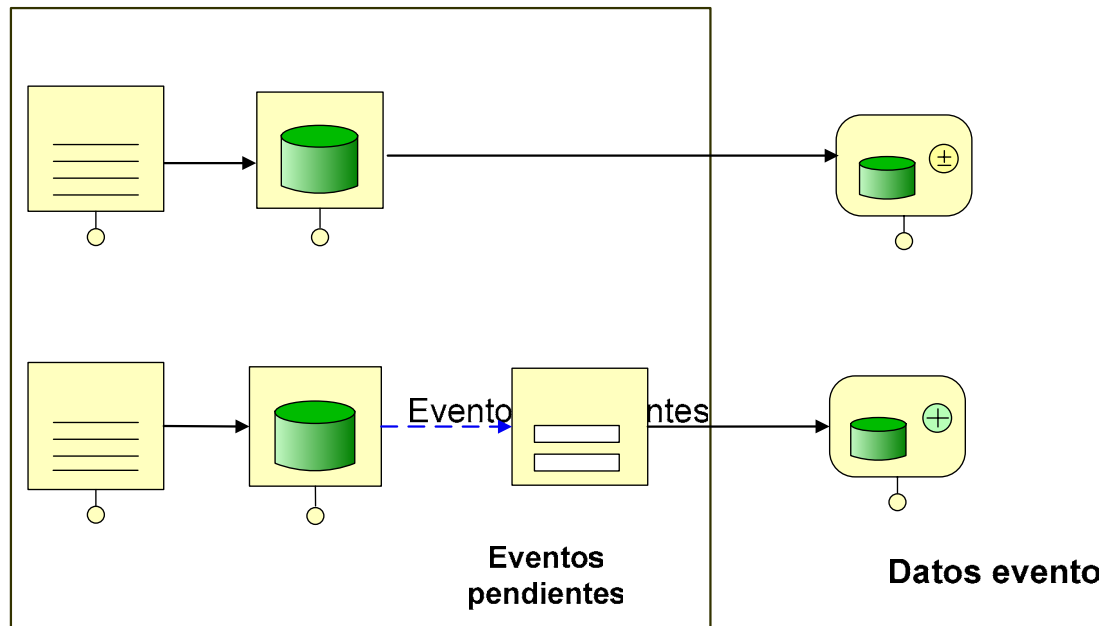


Figura 4.58. Página Eventos Pendientes.

Fuente: Elaboración propia.

Área Eventos Cerrados

La página búsqueda avanzada mostrada en la figura 4.59 permite visualizar los últimos diez eventos ingresados, que ya hayan sido documentados y cerrados, y a su vez permite seleccionar uno de ellos y realizar diferentes acciones como: editar, eliminar o documentar. En el caso de eliminar un evento solo se retira de la visualización ya que en el sistema debe mantenerse el historial del mismo en caso de que el administrador del sistema desee consultarlo en un futuro.

La página mostrada en la figura 4.60, se deberá ingresar el número de caso del evento a buscar, luego se desplegará toda la información referente al mismo y las diferentes acciones que se pueden realizar como: editar el evento, eliminarlo o seguir.

documentando el caso después de haberlo cerrado. Para eliminar un evento se sigue el mismo procedimiento explicado anteriormente para la figura 4.59.

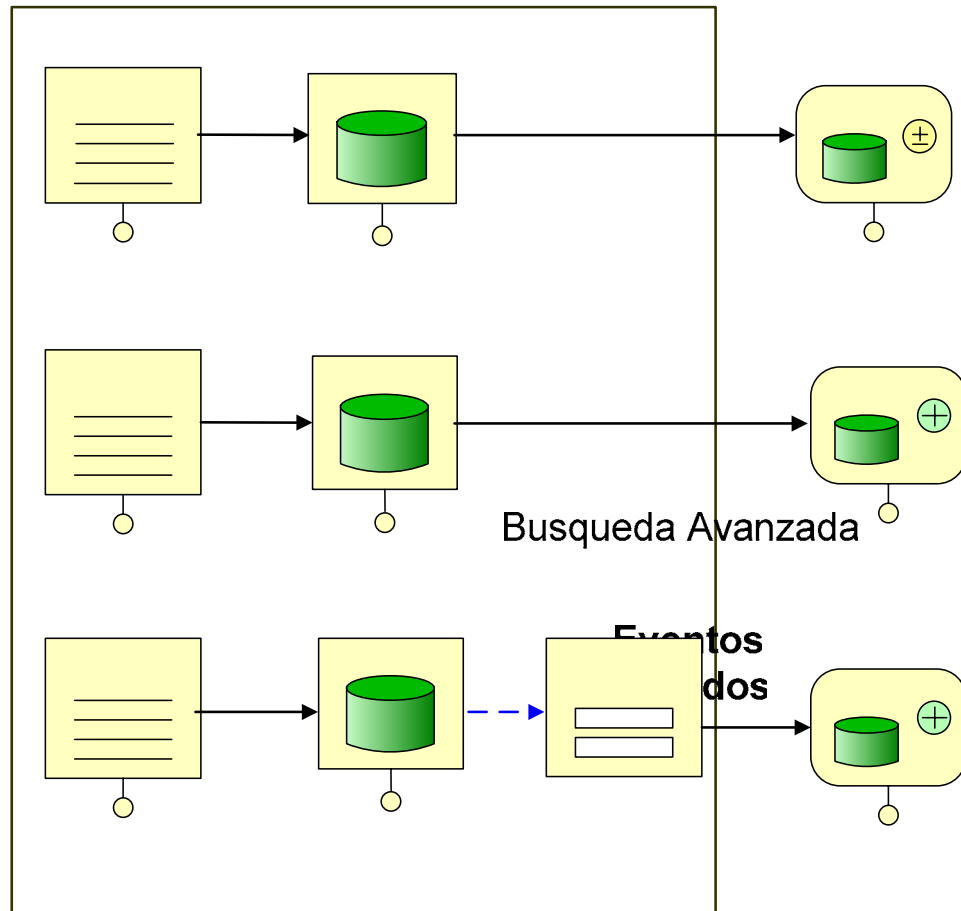


Figura 4.59. Página Búsqueda Avanzada

Fuente: Elaboración propia.

Área Servicios

En la figura 4.61 se muestra la página ingresar servicios, la cual permite al usuario ingresar un servicio en el sistema.

La página mostrada en la figura 4.62, permite al usuario ingresar el nombre del servicio a buscar y el sistema mostrará los datos del mismo para consultarlos.

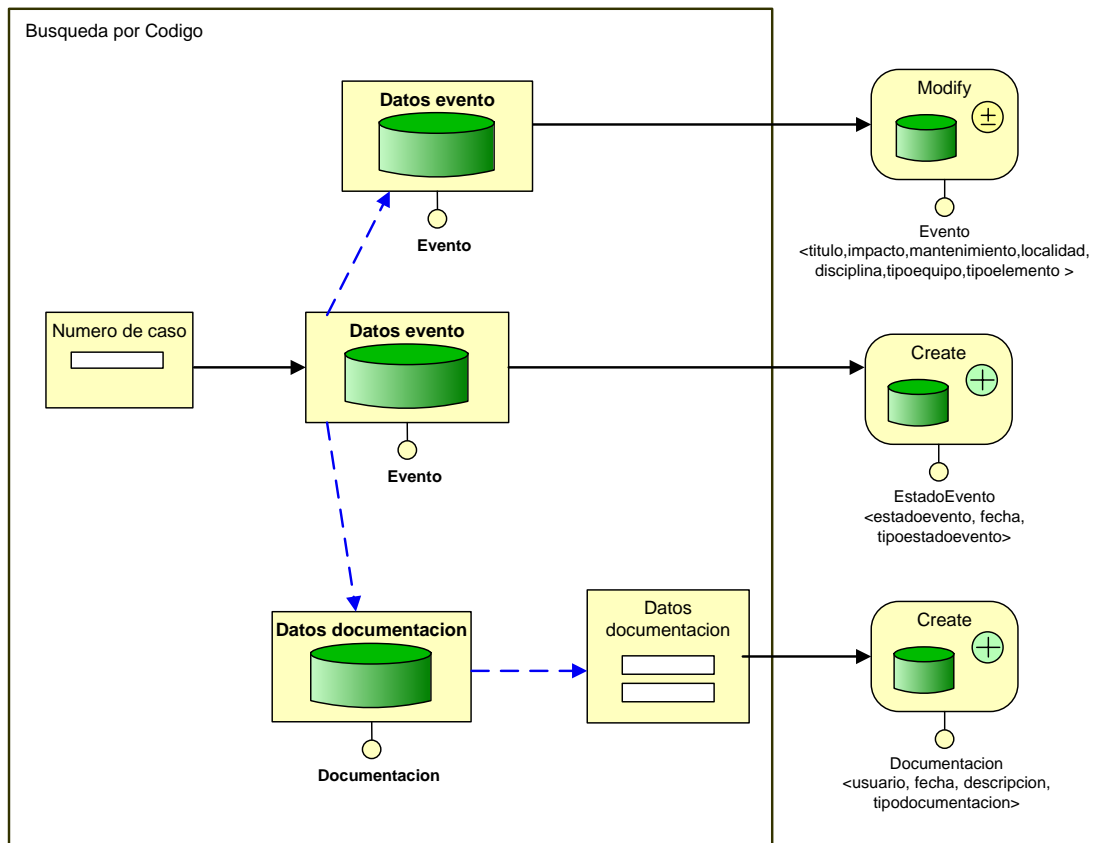


Figura 4.60. Página Búsqueda por Código.

Fuente: Elaboración propia.

La página mostrada en la figura 4.63, permite al usuario editar un servicio, para ello el usuario debe ingresar el nombre del servicio a modificar y el sistema buscará los datos que coincidan con el nombre ingresado.

En la figura 4.64 se muestra la página eliminar servicio, la cual permite al usuario eliminar un servicio del sistema, ingresando el nombre del mismo.

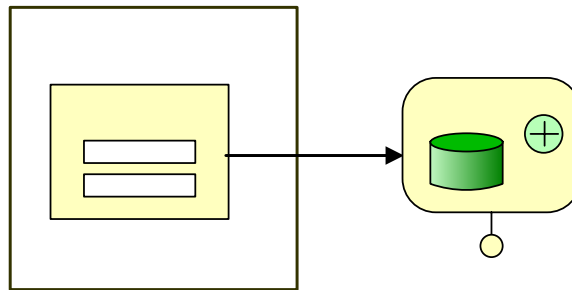


Figura 4.61. Página Ingresar Servicio.

Fuente: Elaboración propia.

Ingresar servicio

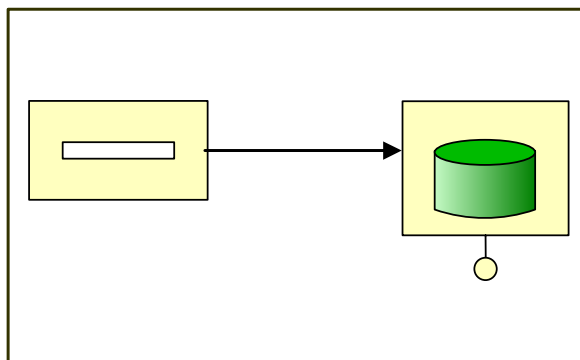


Figura 4.62. Página Consultar Servicio.

Fuente: Elaboración propia.

Datos servicio

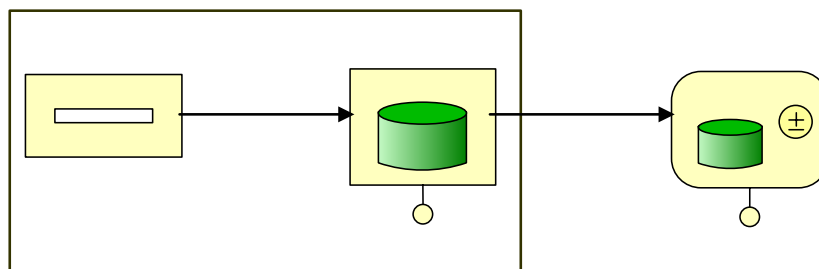
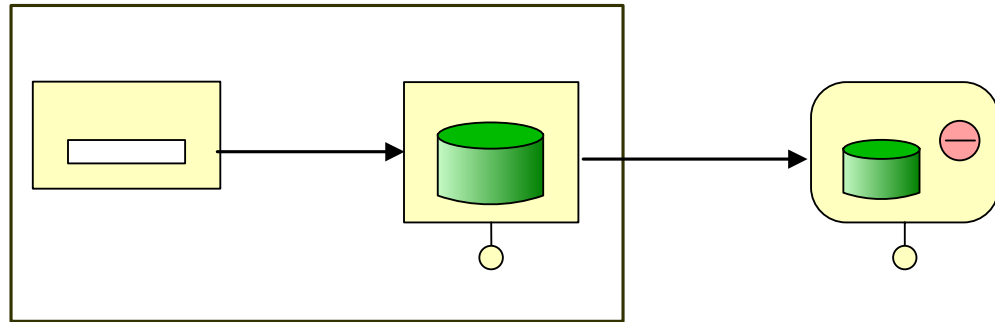


Figura 4.63. Página Modificar Servicio.

Fuente: Elaboración propia.

Consultar servicio

Nombre servicio



Eliminar servicio
Figura 4.64. Página Eliminar Servicio.

Fuente: Elaboración propia.

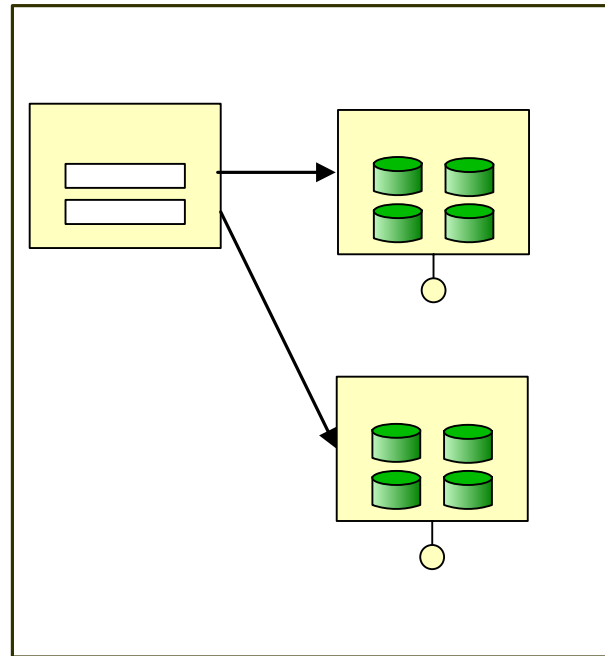
Nombre servicio

Área Reportes

La figura 4.65 muestra la página reporte estándar la cual permite al usuario generar un reporte de todas las alarmas y los eventos de acuerdo a su estado, seleccionando los datos mostrados en el formulario.

La página mostrada en la figura 4.66 permite generar un tipo de reporte por impactos, seleccionando en el formulario el tipo de impacto que se desea generar. Este reporte muestra la cantidad de impactos generados en cada rango de fecha, las alarmas y eventos.

La figura 4.67 muestra la página reporte por mantenimiento la cual permite generar los diferentes tipos de mantenimientos ejecutados por cada disciplina en una localidad, o viceversa, en un rango de fecha definido por el usuario.



Reporte estandar

Reporte estandar

Figura 4.65. Página Reporte Estándar.

Fuente: Elaboración propia.

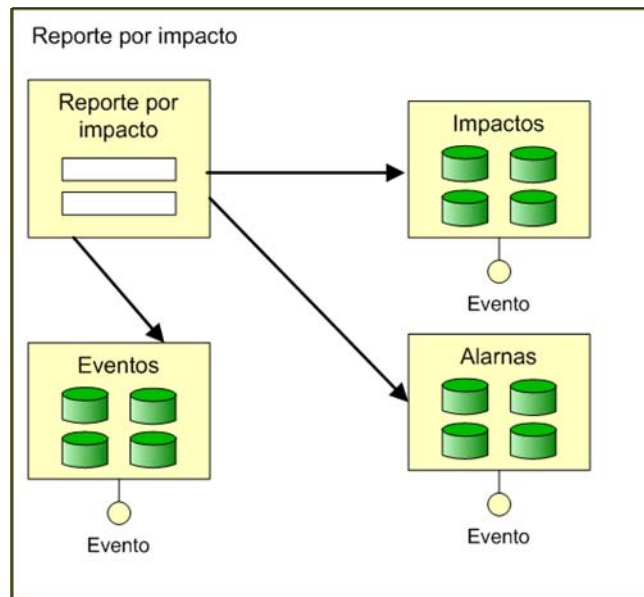
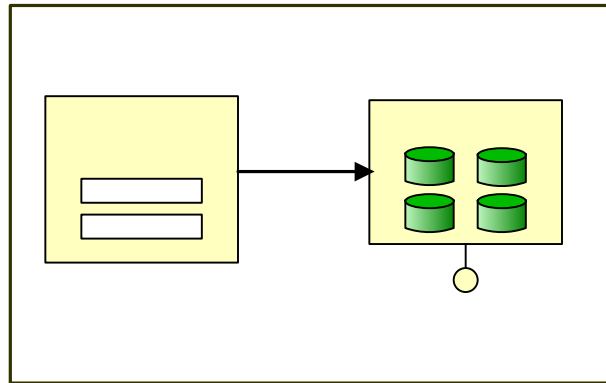


Figura 4.66. Página Reporte por Impacto.

Fuente: Elaboración propia.



Reporte por manteni

Figura 4.67. Página Reporte por Mantenimiento.

Fuente: Elaboración propia.

**Reporte por
mantenimiento**

La figura 4.68 muestra la página reporte por equipo, esta permite seleccionar un tipo de reporte, bien sea por mantenimientos, impactos, alamas y eventos incluyendo otros parámetros contenidos en el formulario. A partir de éste se generarán los datos que coincidan con la selección realizada por el usuario.

La página mostrada en la figura 4.69 permite generar los datos de las alarmas y/o los eventos, a partir de una búsqueda, bien sea por nombre de equipo o por dirección IP, se deben ingresar los datos asignados a la búsqueda y a partir del resultado arrojado se selecciona la opción por la cual se va a realizar la consulta para generar el reporte. Para realizar este tipo de búsqueda el usuario debe tener conocimiento acerca del nombre y las direcciones asignadas a los equipos, o de lo contrario el sistema no mostrará ningún resultado.

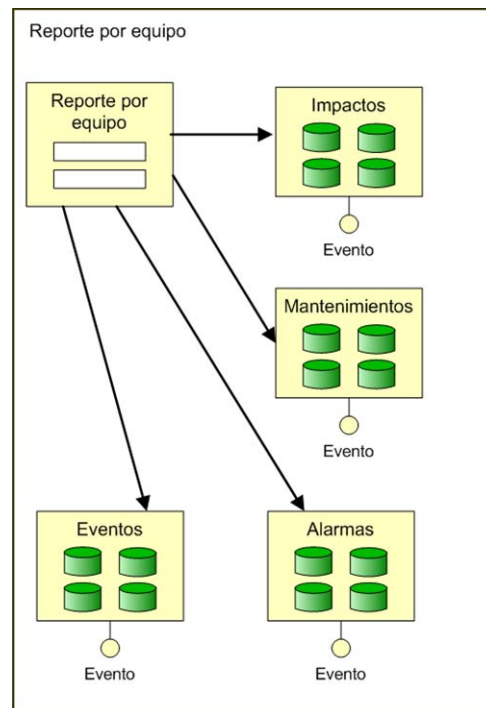


Figura 4.68. Página Reporte por Equipo.

Fuente: Elaboración propia.

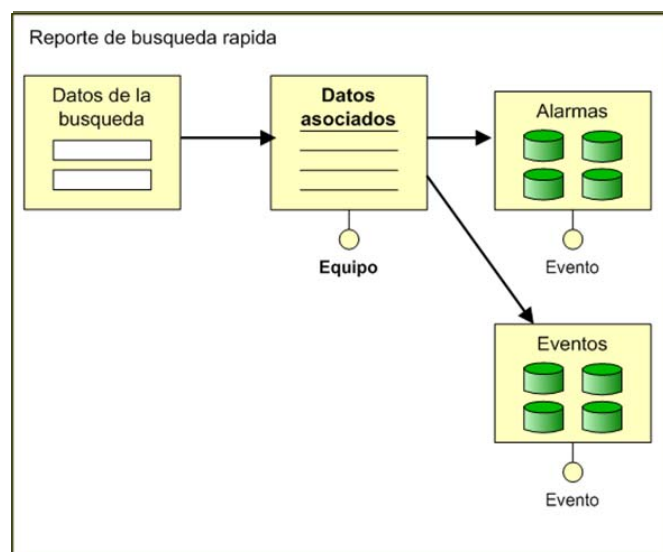


Figura 4.69. Página Reporte de Búsqueda Rápida.

Fuente: Elaboración propia.

4.3.4.2. Vista Analista de Monitoreo

La vista de analista de monitoreo comprende las áreas de inicio de sesión, consultar alarmas, alarmas cerradas, eventos, eventos cerrados, servicios y reportes dentro de las cuales se encuentran las diferentes páginas que permiten realizar las diversas acciones dentro de cada una de ellas. Esta vista es similar a la anterior ya que comprende la mayoría de las áreas descritas en la vista administrador del sistema, por lo tanto no se van a mostrar las páginas ya explicadas. La vista analista de monitoreo se puede observar en la figura 4.70.

A continuación se muestra la tabla 4.11 con la descripción de la vista.

Tabla 4.11. Descripción de la vista analista de monitoreo.

Vista	Vista Analista de Monitoreo
Descripción	Permite realizar todas las acciones que comprenden, la gestión de las alarmas, los eventos, servicios y sus diferentes reportes.
Actores	Analista de Monitoreo.
Casos Uso	Gestionar Alarmas, Gestionar Eventos, Gestionar Servicios, Generar Reportes.

Fuente: Elaboración Propia

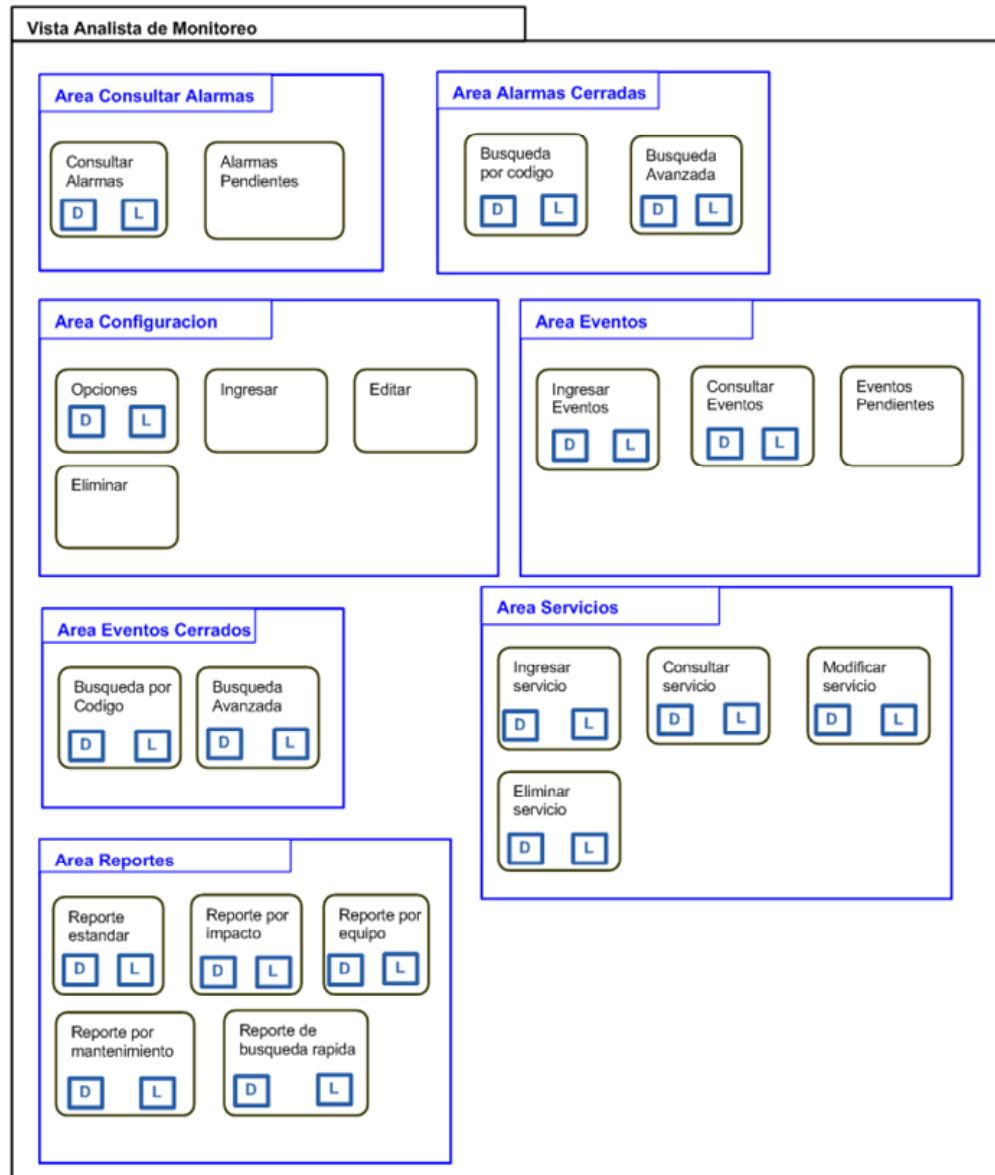


Figura 4.70. Vista Analista de Monitoreo

Fuente: Elaboración Propia

4.3.4.3. Vista Analista de Control de Gestión

La vista analista de control de gestión comprende las áreas servicios y reportes ya explicadas en la vista administrador del sistema, por lo tanto solo se mostrará la vista en la figura 4.71.

A continuación se muestra la tabla 4.12 con la descripción de la vista superintendente.

Tabla 4.12. Descripción de la vista analista de control de gestión.

Vista	Vista Analista de Control de Gestión
Descripción	Permite realizar todas las acciones que comprenden, la gestión de los servicios y generar reportes de las alarmas y los eventos.
Actores	Analista de Control de Gestión.
Casos Uso	Gestionar Alarmas, Gestionar Eventos, Gestionar Servicios, Generar Reportes.

Fuente: Elaboración Propia

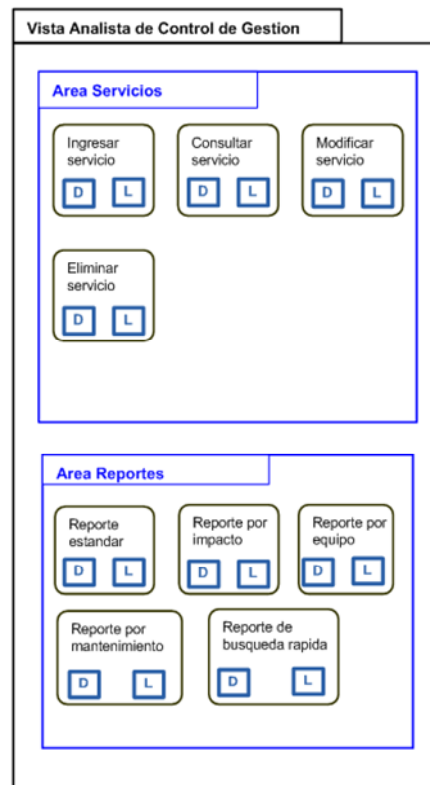


Figura 4.71. Vista Analista de Control de Gestión.

Fuente: Elaboración Propia.

4.3.4.4. Vista Superintendente

La vista superintendente comprende solo el área de reportes, a partir de allí el actor involucrado puede generar los diferentes tipos de reportes que ofrece la aplicación. La figura 4.72 muestra la vista superintendente.

A continuación se muestra la tabla 4.13 con la descripción de la vista superintendente.

Tabla 4.13. Descripción de la vista superintendente.

Vista	Vista Superintendente
Descripción	Permite realizar todas las acciones que comprende la generación de reportes.
Actores	Superintendente.
Casos Uso	Generar Reportes.

Fuente: Elaboración Propia.

4.3.5. Modelo de navegación

En el modelo de navegación como se puede observar en la figura 4.73, se muestran las vistas que componen el sitio y las páginas que las constituyen, es decir, se especifican los vínculos (links) entre páginas y entre unidades de una misma página. El objetivo del diagrama es mostrar los enlaces entre las páginas, debido a que WebML se enfoca en el diseño de la interfaz.

Para simplificar el diagrama WebML permite identificar las páginas que son accesibles desde cualquier otra página dentro de la misma vista como Landmark.

Las páginas por defecto de cada vista de usuario son identificadas como Default.

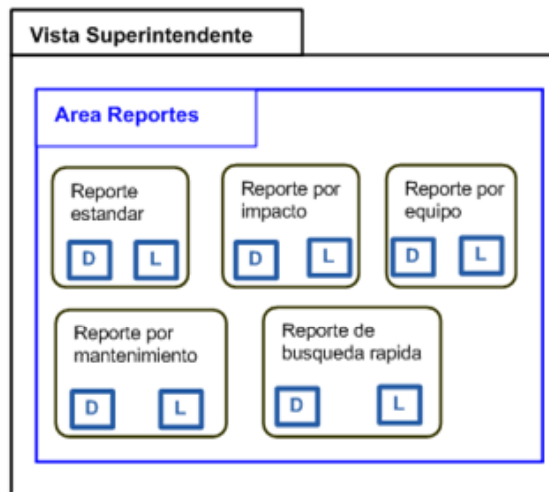


Figura 4.72. Vista Analista Superintendente.

Fuente: Elaboración Propia

4.4. Flujo de Trabajo de Implementación

En la implementación se comenzará con el resultado del diseño y se implementa el sistema en términos de componentes. Para esta fase sólo se creará la línea base ejecutable de la arquitectura. En este flujo de trabajo se comienza a desarrollar la arquitectura del sistema en documentos PHP, y se identifica los distintos componentes necesarios para la ejecución del sistema. Para la fase de elaboración, sólo se presentará un módulo que permitirá demostrar el funcionamiento de la arquitectura.

4.4.1. Identificación de Componentes de la Arquitectura

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo.

Para poder implementar la arquitectura de este sistema, se identificaron los componentes mostrados en la figura 4.74, en ella se muestra el servidor Web Apache

2 que permite la distribución de las páginas, el motor de PHP 5 para ejecutar los scripts que contienen la funcionalidad del sistema, y el sistema manejador de base de datos (DBMS) PostgreSQL que contiene la base de datos y provee la conexión entre ésta y la administración del manejo de los datos.

Adicionalmente, en la misma figura se observa el cliente, provisto de navegador Web encargado de generar el documento HTML, mediante el cual los usuarios podrán visualizar e interactuar con la aplicación. El cliente y el servidor se conectan a través de la Intranet mediante el protocolo de transmisión http.

4.4.2. Implementación de la Arquitectura

En esta fase se implementa la arquitectura a través de la construcción de un prototipo de la página principal de inicio de sesión denominada *índex*, la cual se puede ver en la figura 4.68, la cual conforma el caso de uso Validar Usuarios, y también se ejecutan los componentes que conforman esta arquitectura. Esta página está realizada utilizando lenguaje de scripts PHP, mostrando la interfaz de inicio de sesión y conteniendo las instrucciones necesarias para conectar y consultar la base de datos.

Con esto se pudo comprobar el funcionamiento de la arquitectura, ejecutándose los componentes que la conforman, el servidor Web, el motor de scripts PHP y el manejador de base de datos PostgreSQL, así como el servidor Web que permite la interacción entre el cliente y el servidor utilizando el protocolo de Internet HTTP.

4.4.3. Conclusión de la Fase de Elaboración

Para esta fase se han terminado los artefactos de los flujos de requisitos, análisis y diseño, de tal manera que la siguiente fase, se limitará a codificar y probar los distintos componentes de software del sistema.



Figura 4.73. Modelo de Navegación.

Fuente: Elaboración Propia.

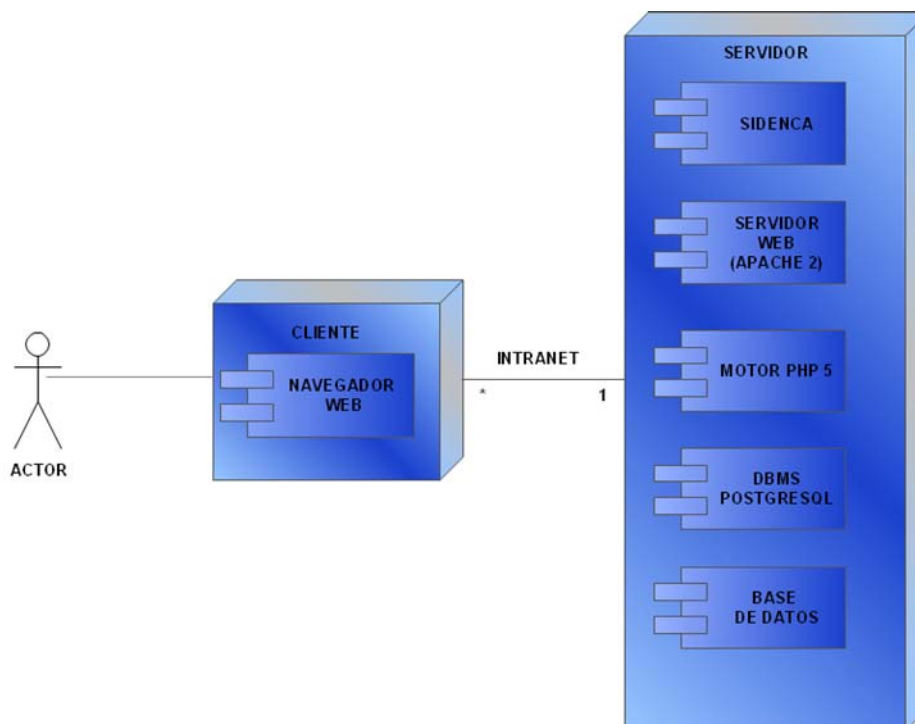


Figura 4.74. Diagrama de Despliegue del Sistema Integrado de Notificación y Comunicación de Alarmas (SIDENCA).

Fuente: Elaboración Propia.

CAPÍTULO V

FASE DE CONSTRUCCIÓN

5.1. Introducción

La fase de construcción tiene como propósito primordial dejar producto de software en su versión operativa inicial en el entorno de usuario, comúnmente llamada “versión beta”. El producto debería tener la calidad adecuada para su aplicación y asegurarse de cumplir con los requisitos, esta fase es análoga al desarrollo, por tanto el énfasis se traslada de la acumulación del conocimiento básico necesario para construir el proyecto a la construcción propiamente dicha de un sistema o producto.

Las fases anteriores han reducido los riesgos críticos y significativos a niveles rutinarios que pueden ser gestionados mediante el plan de construcción. En esta fase se integran los subsistemas y se prueban, luego se integra todo el sistema y se prueba.

Los flujos trabajados en esta fase son: Implementación y prueba, en implementación se podrán observar los subsistemas con los componentes que los integran y las interfaces con sus respectivos bloques de código, y en prueba todas las pruebas realizadas para obtener el correcto funcionamiento del sistema SIDENCA.

5.2. Flujo de Trabajo de Implementación

El flujo de trabajo Implementación, procura la codificación de los módulos que se describieron en las fases anteriores del desarrollo, así como la integración de éstos con las capas definidas en la arquitectura propuesta, de esta manera se

desarrollarán las pantallas de la aplicación, basadas en el modelo de diseño desarrollado con WebML.

A continuación se muestra la descripción de las áreas principales dentro de la aplicación Web luego de su implementación. Se llevará a cabo bajo el siguiente formato:

- Breve explicación del área (en qué consiste).
- Identificación de los datos a manejar por el área. (datos que debe introducir el usuario durante su visita).
- Imagen referencial de la pantalla del sistema.
- Código Fuente.

5.2.1. Área Alarmas

Durante el modelado de la aplicación se definió el área de alarmas, que contempla las páginas consultar alarmas, documentar, editar, eliminar, alarmas cerradas y configuración. Esta vista está solo disponible para el actor administrador del sistema y analista de monitoreo. En la figura 5.1 se observa la página consultar alarmas, la cual muestra todas las fallas generadas por NAGIOS. Para Mostrar los datos de las alarmas se crearon dos scripts los cuales permiten establecer la conexión entre el Sistema Integrado de Notificación y Comunicación de Alarmas (SIDENCA) y NAGIOS.

NAGIOS	CIMOR	Título	Equipo	Elemento	Fecha
OK	NUEVO	MANTMUS2: Falta General	MANTMUS2	DISCO D:	2010-05-06 15:14:13
OK	PROGRESO	MATPCU02: Falta General	MATPCU02	UPTIME	2010-05-05 21:09:25
OK	NUEVO	ER-PLCGUAS: Falta General	ER-PLCGUAS	ENLACE GUARAQUAO - CE COLON PROYECTO ACFJ	2010-05-05 09:19:42
OK	NUEVO	ER-PLCGUAS: Falta General	ER-PLCGUAS	ENLACE PLC PETROGUARICO SERIAL1/0/0	2010-05-04 12:33:57
ALARMADO	NUEVO	PRIDE-319: Falta General	PRIDE-319	UNICO	2010-05-04 10:18:48
ALARMADO	NUEVO	PRIDE-319: Falta General	PRIDE-319	UNICO	2010-05-04 10:18:48
ALARMADO	NUEVO	PRIDE-319: Falta General	PRIDE-319	UNICO	2010-05-04 10:18:48

Figura 5.1. Pantalla Consultar Alarmas: Área Alarmas.

Fuente: Elaboración Propia.

5.2.1.1. Código Fuente Consultar Alarmas

Las clase Piensa se encarga de obtener la información enviada por NAGIOS y realizar las operaciones necesarias para organizar, validar y almacenar los datos recibidos en la base de datos del sistema. De esta forma la aplicación podrá consultar la información directamente desde la base de datos. El archivo mini.php contiene el código necesario que se encarga de recibir los datos directamente de NAGIOS y transferirlos vía URL a la dirección especificada.

```
<?php
require_once(dirname(__FILE__) . '/../db/EventoDBClass.php');
class Piensa
{
var $PIENSA;
```

```

function __construct()
{
    $this->PIENSA = new EventoDB();
}

function __destruct()
{
    unset($this->PIENSA);
}

function notificar($lista)
{
    {
    if ($lista[0]['tx_elemento']!= '')
        $g['elemento']=$lista[0]['tx_elemento'];
    else
        $g['elemento']='UNICO';

    $g['equipo']=$lista[0]['tx_equipo'];
    $g['estado']=$lista[0]['tx_estado'];
    $g['fecha']=$lista[0]['fe_fecha'];

    $g['herramienta']='NAGIOS';

    $g['titulo']=$g['equipo'].": Falla General";

    $g['impacto']='NO_DEFINIDO';
    $g['mantenimiento']='CORRECTIVO';
    $g['tipo_evento']='AUTOMATICO';

    if(!$this->PIENSA->buscarEquipo($g['equipo']))
        $this->PIENSA->ingresarEquipo($g['equipo']);

    if(!$this->PIENSA->buscarElemento($g))
        $this->PIENSA->ingresarElemento($g);
}
}

```

```

$g['herramienta']='CIMOR';
$c=$this->PIENSA->consultaEventoEstado($g);
if($c===false)
    $nc= 0;
else
    $nc= count($c);

    if ($g['estado']=== "DOWN" || $g['estado']=== "UNREACHABLE" ||
$g['estado']=== "CRITICAL" || $g['estado']=== "WARNING" ||
$g['estado']=== "UNKNOWN")
        $mal=true;
    else
        $mal=false;

if(($nc === 0 || $c[0]['tx_estado_evento'] === 'CERRADO' ) &&
$mal)
    {
    $this->PIENSA->ingresarEvento($g);
    $g['co_evento'] = $this->PIENSA->ultimoEvento($g);
    $g['herramienta']='NAGIOS';
    $this->PIENSA->ingresarEventoEstadoEvento($g);
    $g['herramienta']='CIMOR';
    $g['estado']='NUEVO';
    $this->PIENSA->ingresarEventoEstadoEvento($g);
    }
else
    if($nc > 0)
        {
        $g['co_evento'] = $this->PIENSA->ultimoEvento($g);
        $g['herramienta'] = 'NAGIOS';
        $t=$this->PIENSA->consultaEventoEstado($g);

        if($t[0]['tx_estado_evento']!=$g['estado'])
            $this->PIENSA-
>ingresarEventoEstadoEvento($g);

```



```

        }
    }

function analizar($lista)
{
    $n = count($lista);

    if( $tiempo=true )
    {
        $this->notificar($lista);
        for($i=0;$i<$n;$i++)
            $this->PIENSA->borrarFila($lista[$i]['co_buffer']);
    }
    else
        if($n > 2)
        {
            for($i=2;$i<$n;$i++)
                $this->PIENSA-
>borrarFila($lista[$i]['co_buffer']);
        }
    }

function equipos()
{
    $r = $this->PIENSA->listarEquiposElementos();

    $n = count($r);
    for($i=0;$i<1;$i++)
    {
        $eq = $this->PIENSA->un_equipo($r[$i]);
        print '<pre>';print_r($eq);print '</pre>';
        $this->analizar($eq);
        unset($eq);
    }
}

```

```
function elementos()  
{  
    $r = $this->PIENSA->listarEquiposElementos(false);  
  
    $n = count($r);  
  
    for($i=0;$i<1;$i++)  
    {  
        $eq = $this->PIENSA->un_elemento($r[$i]);  
        print '<pre>';print_r($eq);print '</pre>';  
        $this->analizar($eq);  
        unset($eq);  
    }  
}  
?  
#!/usr/bin/php5  
<?php  
$tipo = $argv['1'];  
$equipo = $argv['2'];  
  
if($tipo == "S")  
{  
    $elemento = $argv['3'];  
    $fecha = $argv['4'];  
    $estado = $argv['5'];  
    $informacion = $argv['6'];  
}  
else  
{  
    $fecha = $argv['3'];  
    $estado = $argv['4'];  
    $informacion = $argv['5'];  
}
```

```

$a=fopen("http://10.172.20.8/consola/mini.php?tipo=".urlencode($tipo)
."&equipo=".urlencode($equipo)."&elemento=".urlencode($elemento)."&
fecha=".urlencode($fecha)."&estado=".urlencode($estado)."&informacio
n=".urlencode($informacion),$info);
fclose($a);

```

```

$a=fopen("http://167.175.55.108/consola/mini.php?tipo=".urlencode($t
ipo)."&equipo=".urlencode($equipo)."&elemento=".urlencode($elemento)
."&fecha=".urlencode($fecha)."&estado=".urlencode($estado)."&informa
cion=".urlencode($informacion),$info);
fclose($a);
?>

```

IUmostraralarmas

```

<?php
require("IUAplicacion.php");
require_once(dirname(__FILE__)."/../dominio/procesaralarmas.php");
require_once(dirname(__FILE__)."/../dominio/procesareventos.php");
require_once(dirname(__FILE__)."/../dominio/ReceptoraAlarmas.php");

```

```

class IUmostraralarmas extends Aplicacion {

```

```

    function IUmostraralarmas(){

```

```

        $this->proc_dat=new procesar_alarmas();

```

```

        $this->proc_event=new procesar_eventos();

```

```

    }

```

```

function AgregarTablas($num_evento,$datos1,$datos6){

```

```

    $Out.='<table width="100%" border="0">

```

```



```

```

<input type="image" src="./iu/imagenes-aplicacion/documentar.jpg"
width="32" height="32" onclick="hacer('\2\',
\'accion'. $num_evento. '\');" />
</td>
<td width="5%" align="center">
<div class="Principio-Boton"> </div>
<input id="boton'. $datos1[0]['tx_numero_evento']. '" name="boton"
class="Boton" background:red type="button" value="+"
onclick="mostrarOcultarTablas('\tabla'. $datos1[0]['tx_numero_evento'
]. '\', '\tabla_estado'. $datos1[0]['tx_numero_evento']. '\',
\'boton'. $datos1[0]['tx_numero_evento']. '\');" />
<div class="Final-Boton"></div>
</td>
</tr>
</table>
&nbsp;

```

```

<div id="tabla'. $datos1[0]['tx_numero_evento']. '"
style="display: none">
<table class="estilo-tabla1" border="0">
<tr>
<td align="center"><span class="Sub-Titulo-
cobose">Mantenimiento</span></td>
<td align="center"><span class="Sub-Titulo-
cobos">'. $datos1[0]['tx_mantenimiento']. '</span></td>
<td align="center"><span class="Sub-Titulo-cobose">Impacto
Evento:</span></td>
<td align="center"><span class="Sub-Titulo-
cobos">'. $datos1[0]['tx_impacto_evento']. '</span></td>
</tr>
<tr>

```

```

        <td align="center"><span class="Sub-Titulo-cobose">Tipo
Equipo:</span></td>
        <td align="center"><span class="Sub-Titulo-
cobos">' . $datos1[0][ 'tx_tipo_equipo' ] . '</span></td>
        <td align="center"><span class="Sub-Titulo-
cobose">IP:</span></td>
        <td align="center"><span class="Sub-Titulo-
cobos">' . $datos1[0][ 'tx_ip' ] . '</span></td>
    </tr>
    <tr>

        <td align="center"><span class="Sub-Titulo-cobose">Tipo
Evento</span></td>
        <td align="center"><span class="Sub-Titulo-
cobos">' . $datos1[0][ 'tx_tipo_evento' ] . '</span></td>
        <td align="center"><span class="Sub-Titulo-cobose">Tipo
Elemento:</span></td>
        <td align="center"><span class="Sub-Titulo-
cobos">' . $datos1[0][ 'tx_tipo_elemento' ] . '</span></td>
    </tr>
    <tr>

        <td align="center"><span class="Sub-Titulo-
cobose">Localidad:</span></td>
        <td align="center"><span class="Sub-Titulo-
cobos">' . $datos1[0][ 'tx_localidad' ] . '</span></td>
        <td align="center"><span class="Sub-Titulo-
cobose">Direccion:</span></td>
        <td align="center"><span class="Sub-Titulo-
cobos">' . $datos1[0][ 'tx_direccion' ] . '</span></td>
    </tr>
    <tr>

        <td align="center"><span class="Sub-Titulo-
cobose">Disciplina:</span></td>

```

```

        <td align="center"><span class="Sub-Titulo-cobos">'. $datos1[0]['tx_grupo'] .'</span></td>
        <td align="center"><span class="Sub-Titulo-cobose">Tipificacion</span></td>
        <td align="center"><span class="Sub-Titulo-cobos">'. $datos1[0]['tx_tipificacion'] .'</span></td>
    </tr>
    <tr>
        <td align="center"><a href="javascript:mostrardiv('\`tabla_estado'. $datos1[0]['tx_numero_evento'] .'\'');" class="contenedor-opciones"><span class="Sub-Titulo-cobose-link">Ver mas >>></span></a></td>
        <td></td>
        <td></td>
        <td></td>
    </tr>
</table>

<div id="tabla_estado'. $datos1[0]['tx_numero_evento'] ." style="display: none">
    <table class="estilo-tabla2" border="0">
        <tr>
            <td align="center"><span class="Sub-Titulo-cobose">Tipo Estado Evento</span></td>
            <td align="center"><span class="Sub-Titulo-cobose">Estado Evento</span></td>
            <td align="center"><span class="Sub-Titulo-cobose">Fecha Estado Evento</span></td>
        </tr>
        ';
        for($i=0;$i<count($datos6);$i++){
            if($datos6[$i]['tx_tipo_estado_evento']=="NAGIOS" and ($datos6[$i]['tx_estado_evento']=="DOWN" || $datos6[$i]['tx_estado_evento']=="CRITICAL")){

```

```

        $Out.='<tr>
        <td align="center"><span class="Sub-Titulo-
cobos">'. $datos6[$i]['tx_tipo_estado_evento'] .'</span></td>
        <td bgcolor="#FF3300" align="center"><span class="Sub-Titulo-
cobos">'. $datos6[$i]['tx_estado_evento'] .'</span></td>
        <td align="center"><span class="Sub-Titulo-
cobos">'. $datos6[$i]['fe_fecha_estado'] .'</span></td>
        </tr>';}

```

```

        else if($datos6[$i]['tx_tipo_estado_evento']=="NAGIOS" and
($datos6[$i]['tx_estado_evento']=="UP" ||
$datos6[$i]['tx_estado_evento']=="OK")){

```

```

        $Out.='<tr>
        <td align="center"><span class="Sub-Titulo-
cobos">'. $datos6[$i]['tx_tipo_estado_evento'] .'</span></td>
        <td bgcolor="#66FF00" align="center"><span class="Sub-Titulo-
cobos">'. $datos6[$i]['tx_estado_evento'] .'</span></td>
        <td align="center"><span class="Sub-Titulo-
cobos">'. $datos6[$i]['fe_fecha_estado'] .'</span></td>
        </tr>';}

```

```

        else if($datos6[$i]['tx_tipo_estado_evento']=="NAGIOS" and
$datos6[$i]['tx_estado_evento']=="WARNING"){

```

```

        $Out.='<tr>
        <td align="center"><span class="Sub-Titulo-
cobos">'. $datos6[$i]['tx_tipo_estado_evento'] .'</span></td>
        <td bgcolor="#FFCC00" align="center"><span class="Sub-Titulo-
cobos">'. $datos6[$i]['tx_estado_evento'] .'</span></td>
        <td align="center"><span class="Sub-Titulo-
cobos">'. $datos6[$i]['fe_fecha_estado'] .'</span></td>
        </tr>';}

```

```

        else if($datos6[$i]['tx_tipo_estado_evento']=="CIMOR" ||
$datos6[$i]['tx_tipo_estado_evento']=="SIGA"){

```

```

        $Out.='<tr>
        <td align="center"><span class="Sub-Titulo-
cobos">'. $datos6[$i]['tx_tipo_estado_evento'] .'</span></td>
        <td align="center"><span class="Sub-Titulo-
cobos">'. $datos6[$i]['tx_estado_evento'] .'</span></td>

```



```

                <td align="center"><span class="Sub-Titulo-
cobos">'. $datos6[$i]['fe_fecha_estado'].'</span></td>
            </tr>';}
        }
        $Out.='</table>
</div>
</div>';
$this->todo.=$Out;

}
}
?>

```

procesar_alarmas

```

<?php

require_once(dirname(__FILE__)."/../bdd/usuario.php");
require_once(dirname(__FILE__)."/../bdd/ConsultasEventos.php");
require_once(dirname(__FILE__)."/../bdd/ConsultasReportesCimor.php");
;

class procesar_alarmas{

function procesar_alarmas(){

    $this->user=new usuario();
    $this->event=new consultas_eventos();
    $this->report=new ConsultasReportesCimor();

}

function Buscar_alarmas_no_pendientes(){
$j=0;
$eventos=$this->user->buscar_eventos_no_eliminados();

```

```

for($i=0;$i<count($eventos);$i++){
    $eventos_nuevos=$this->user-
>buscar_no_cerrados($eventos[$i]['co_evento']);//BUSCO  LOS  QUE  NO
ESTEN CERRADOS
    if($eventos_nuevos[0]['tx_estado_evento']!= 'CERRADO'           &&
$eventos_nuevos[0]['tx_estado_evento']!= 'PENDIENTE'             &&
$eventos_nuevos[0]['tx_estado_evento']!= 'ELIMINADO'){
        $arre_eventos[$j]=$eventos[$i]['co_evento'];
        $j++;}
    }

return $arre_eventos;

}

function buscar_alarmas_no_eliminadas(){
$j=0;
$eventos=$this->user->buscar_eventos_no_eliminados();
for($i=0;$i<count($eventos);$i++){
    $eventos_nuevos=$this->user-
>buscar_no_cerrados($eventos[$i]['co_evento']);//BUSCO  LOS  QUE  NO
ESTEN CERRADOS
    if($eventos_nuevos[0]['tx_estado_evento']!= 'CERRADO'           &&
$eventos_nuevos[0]['tx_estado_evento']!= 'PROGRESO'             &&
$eventos_nuevos[0]['tx_estado_evento']!= 'ELIMINADO'           &&
$eventos_nuevos[0]['tx_estado_evento']!= 'NUEVO'){
        $arre_eventos[$j]=$eventos[$i]['co_evento'];
        $j++;}
    }

return $arre_eventos;
}
}?>

```

En la página gestionar configuración mostrada a continuación en la figura 5.2 se puede visualizar las opciones (disciplinas, Localidades, tipo equipos, tipo elementos) a partir de las cuales el usuario puede realizar las distintas operaciones como ingresar, editar o eliminar. Desde esta pantalla se puede ver la información detallada.



Figura 5.2. Pantalla Gestionar Configuración: Área Alarmas.

Fuente: Elaboración Propia.

5.2.1.2. Código Fuente Gestionar Configuración

IUIngresar_Opciones

```
<?php
require_once("IUAplicacion.php");
require_once(dirname(__FILE__)."/../dominio/procesaralarmas.php");
require_once(dirname(__FILE__)."/../dominio/ReceptoraOpciones.php");

class IUIngresar_Opciones extends Aplicacion{
```

```

function IUIngresar_Opciones(){
$this->proc_dat=new procesar_alarmas();
}

function MostrarOpciones(){
$Out.='<span class="Titulo-Aplicacion">INGRESAR OPCIONES</span>
      <span class="Separador_Modulo"></span>
      <br></br>
      <br></br>
      <br></br>
      <br></br>
      <br></br>
      <span class="Sub-Titulo-cobose">Seleccione la opcion
que desea ingresar</span>
      <br></br>
      <br></br>'.

//////////////////////////////////////////FORMULARIO
LOCALIDAD//////////////////////////////////////////
'<form name="formlocalidad" id="formlocalidad" method="POST">
<table border="0">
  <tr>
    <td width="10">&nbsp;</td>
    <td></td>
    <td></td>
  </tr>
  <tr>
    <td width="10">&nbsp;</td>
    <td><span class="Sub-Titulo-cobose">Localidad</span></td>
    <td></td>
  </tr>
  <tr>
    <td width="10">&nbsp;</td>

```

```

        <td><select          class="select"          name="selectlocalidad"
id="selectlocalidad">';

        $tx_localidad=$this->proc_dat->retornar_localidad();
        for($i=0;$i<count($tx_localidad);$i++)
        {
            $Out.='<option
value="'. $tx_localidad[$i]['tx_localidad']. "'>' . $tx_localidad[$i]['t
x_localidad']. '</option>';
        }

        $Out.='</select></td>

        <td>
<input type="image" src="./iu/imagenes-aplicacion/editar.jpg"
width="28" height="28" onclick="validarlocalidad(\ '1\ '); ">
<a href="javascript:validarlocalidad(\ '2\ '); "></a>
<input type="image" src="./iu/imagenes-aplicacion/SimboloMas.gif"
width="25" height="25" onclick="validarlocalidad(\ '3\ '); ">
        </td>
        </tr>

    </table>
</form>'.

    ///////////////////////////////////////////FORMULARIO
DISCIPLINA////////////////////////////////////
'<form name="formdisciplina" id="formdisciplina" method="POST">
<table border="0">
    <tr>
        <td width="10">&nbsp;</td>
        <td></td>
        <td></td>
    </tr>
    <tr>
        <td width="10">&nbsp;</td>

```

```

<td><span class="Sub-Titulo-cobose">Disciplina</span></td>
<td></td>
</tr>
<tr>
<td width="10">&nbsp;</td>
<td><select          class="select"          name="selectdisciplina"
id="selectdisciplina">';

        $tx_grupo_solucionador=$this->proc_data-
>retornar_grupo_solucionador();
        for($i=0;$i<count($tx_grupo_solucionador);$i++)
        {
            $Out.='<option
value="'. $tx_grupo_solucionador[$i]['tx_grupo'].'">'. $tx_grupo_soluc
ionador[$i]['tx_grupo']. '</option>';
        }

        $Out.='</select></td>
<td>
        <input          type="image"          src="./iu/imagenes-
aplicacion/editar.jpg"          width="28"          height="28"
onclick="validardisciplina(\`1\`);">
        <a href="javascript:validardisciplina(\`2\`);"></a>
        <input          type="image"          src="./iu/imagenes-
aplicacion/SimboloMas.gif"          width="25"          height="25"
onclick="validardisciplina(\`3\`);">
        </td>
</tr>
</table>
</form>'.

'<form name="formtipoequipo" id="formtipoequipo" method="POST">
<table border="0">

```

```

<tr>
<td width="10">&nbsp;</td>
<td></td>
<td></td>
</tr>
<tr>
<td width="10">&nbsp;</td>
<td><span class="Sub-Titulo-cobose">Tipo Equipo</span></td>
<td></td>
</tr>
<tr>
<td width="10">&nbsp;</td>
<td><select      class="select"      name="tipo_equipo"      id=
"tipo_equipo">' ;
      $tx_tipoEquipos=$this->proc_dat-
>retornar_tipoEquipos();

      for($i=0;$i<count($tx_tipoEquipos);$i++)
      {

          $Out.='<option
value="'. $tx_tipoEquipos[$i]['tx_tipoEquipo']. "'>' . $tx_tipoEquipos[
$i]['tx_tipoEquipo']. '</option>';
      }

      $Out.='</select></td>
      <td><input type="image" src="./iu/imagenes-
aplicacion/editar.jpg" width="28" height="28"
onclick="validartipoequipo(\'1\');">
      <a href="javascript:validartipoequipo(\'2\');"></a>
      <input type="image" src="./iu/imagenes-
aplicacion/SimboloMas.gif" width="25" height="25"
onclick="validartipoequipo(\'3\');">

```

```

                </td>
            </tr>
        </table>
    </form>'.
    ////////////////////////////////////FORMULARIO
ELEMENTO//////////////////////////////////////TIPO
'<form name="formtipoelemento" id="formtipoelemento" method="POST">
    <table border="0">
        <tr>
            <td width="10">&nbsp;</td>
            <td></td>
            <td></td>
        </tr>
        <tr>
            <td width="10">&nbsp;</td>
            <td><span class="Sub-Titulo-cobose">Tipo Elemento</span></td>
            <td></td>
        </tr>
        <tr>
            <td width="10">&nbsp;</td>
            <td><select class="select" name="tipo_elemento"
id="tipo_elemento">';
                $tx_tipo_elementos=$this->proc_dat-
>retornar_tipo_elementos();
                for($i=0;$i<count($tx_tipo_elementos);$i++)
                {
                    $Out.='<option
value="'. $tx_tipo_elementos[$i]['tx_tipo_elemento'].'">' . $tx_tipo_el
ementos[$i]['tx_tipo_elemento']. '</option>';
                }

            $Out.='</select></td>
            <td>
            <input type="image" src="./iu/imagenes-aplicacion/editar.jpg"
width="28" height="28" onclick="validartipoelemento(\`1\`);">

```



```

<a href="javascript:validartipoelemento('\2\');"></a>
<input type="image" src="./iu/imagenes-aplicacion/SimboloMas.gif"
width="25" height="25" onclick="validartipoelemento('\3\');">
</td>
</tr>
</table>
</div>
</form>
<br></br>
<br></br>' ;

$this->todo.=$Out;
}
}

?>

```

5.2.2. Área Eventos

Durante el modelado de la aplicación se definió el área eventos, la cual contempla las páginas ingresar eventos, consultar eventos, documentar, editar, eliminar y eventos cerrados. Esta vista está disponible sólo para el administrador del sistema y el analista de monitoreo. En la figura 5.3 se observa la página consultar eventos.

PDVSA Sistema Integrado de Notificación y Comunicación de Alarmas

Bienvenido ESTHER MARIA COBOS RAMIREZ ADMINISTRADOR Lunes 26 de Julio de 2010

SISTEMA ALARMAS **EVENTOS** SERVICIO REPORTES AYUDA CERRAR SESION

EVENTOS

[Ver Eventos Pendientes](#)

CIMOR	Titulo	Equipo	Elemento	Fecha	
PROGRESO	ES-QQPIUM01. FALLA GENERAL	ES-QQPIUM01	UNICO	2010-05-21 17:23:51	
PROGRESO	FALLA	MOTOGENERADOR	UNICO	2010-06-04 18:07:29	
PROGRESO	FALLA-TOTAL DEL URA_PMON_ORIF25	ORIMAT02	URA_PMON_ORIF25	2010-06-14 16:30:56	

Figura 5.3. Pantalla Consultar Eventos: Área Eventos.

Fuente: Elaboración Propia.

5.2.2.1. Código Fuente Consultar Eventos

IUconsultar_eventos

```
<?php
require_once("IUAplicacion.php");
require_once(dirname(__FILE__)."../dominio/procesareventos.php");
require_once(dirname(__FILE__)."../dominio/ReceptoraEventos.php");

class IUconsultar_eventos extends Aplicacion{

function IUconsultar_eventos(){
$this->proc_dat=new procesar_eventos();
}

function AgregarTablas($num_evento,$datos1,$datos6){
```

```

$Out.='<table width="100%" border="0">
    <input type="hidden" name="accion"
id="accion'.$num_evento.'" value="'.$num_evento.'"/>

<tr>
<td width="8%" align="center"><span class="Sub-Titulo-
cobos">'.$datos1[0]['tx_estado_cimor'].'</span></td>
<td width="20%" align="center"><span class="Sub-Titulo-
cobos">'.$datos1[0]['tx_titulo'].'</span></td>';
if($datos1[0]['tx_tipo_evento']=='ADICIONAL')
$Out.='<td width="20%" align="center"><span class="Sub-Titulo-
cobos">'.$datos1[0]['tx_informacion_adicional'].'</span></td>';
else if($datos1[0]['tx_tipo_evento']=='MANUAL')
$Out.='<td width="20%" align="center"><span class="Sub-Titulo-
cobos">'.$datos1[0]['tx_equipo'].'</span></td>';

$Out.='<td width="20%" align="center"><span class="Sub-Titulo-
cobos">'.$datos1[0]['tx_elemento'].'</span></td>
<td width="8%" align="center"><span class="Sub-Titulo-
cobos">'.$datos1[0]['fe_fecha_evento'].'</span></td>
<td width="8%" align="center">
<input type="image" src="./iu/imagenes-aplicacion/editar.jpg"
width="28" height="28" onclick="hacer(\'0\',
\'accion'.$num_evento.\');">
<input type="image" src="./iu/imagenes-aplicacion/eliminar.jpg"
width="25" height="25" onclick="hacer(\'1\',
\'accion'.$num_evento.\');">
<input type="image" src="./iu/imagenes-aplicacion/documentar.jpg"
width="32" height="32" onclick="hacer(\'2\',
\'accion'.$num_evento.\');">
</td>
<td width="5%" align="center">
<div class="Principio-Boton"> </div>

```

```



```

```

</tr>
<tr>

    <td align="center"><span class="Sub-Titulo-cobose">Tipo
Evento</span></td>
    <td align="center"><span class="Sub-Titulo-
cobos">'. $datos1[0]['tx_tipo_evento'] .'</span></td>
    <td align="center"><span class="Sub-Titulo-cobose">Tipo
Elemento:</span></td>
    <td align="center"><span class="Sub-Titulo-
cobos">'. $datos1[0]['tx_tipo_elemento'] .'</span></td>
</tr>
<tr>

    <td align="center"><span class="Sub-Titulo-
cobose">Localidad:</span></td>
    <td align="center"><span class="Sub-Titulo-
cobos">'. $datos1[0]['tx_localidad'] .'</span></td>
    <td align="center"><span class="Sub-Titulo-
cobose">Direccion:</span></td>
    <td align="center"><span class="Sub-Titulo-
cobos">'. $datos1[0]['tx_direccion'] .'</span></td>
</tr>
<tr>

    <td align="center"><span class="Sub-Titulo-
cobose">Disciplina:</span></td>
    <td align="center"><span class="Sub-Titulo-
cobos">'. $datos1[0]['tx_grupo'] .'</span></td>
    <td align="center"><span class="Sub-Titulo-
cobose">Tipificacion</span></td>
    <td align="center"><span class="Sub-Titulo-
cobos">'. $datos1[0]['tx_tipificacion'] .'</span></td>
</tr>
<tr>

```

```

        <td align="center"><a
href="javascript:mostrardiv(\'tabla_estado'. $datos1[0][\'tx_numero_ev
ento\'].\' \');" class="contenedor-opciones"><span class="Sub-Titulo-
cobose-link">Ver mas >>></span></a></td>
        <td></td>
        <td></td>
        <td></td>
</tr>
</table>
<div id="tabla_estado'. $datos1[0][\'tx_numero_evento\'].\' "
style="display: none">
<table class="estilo-tabla2" border="0">
<tr>
<td align="center"><span class="Sub-Titulo-cobose">Tipo Estado
Evento</span></td>
<td align="center"><span class="Sub-Titulo-cobose">Estado
Evento</span></td>
<td align="center"><span class="Sub-Titulo-cobose">Fecha
Estado Evento</span></td>
</tr>
';
for($i=0;$i<count($datos6);$i++){
    if($datos6[$i][\'tx_tipo_estado_evento\']== "NAGIOS" and
($datos6[$i][\'tx_estado_evento\']== "DOWN" ||
$datos6[$i][\'tx_estado_evento\']== "CRITICAL")){
        $Out.='<tr> <td align="center"><span class="Sub-Titulo-
cobos">'. $datos6[$i][\'tx_tipo_estado_evento\'].\'<span></td>
        <td bgcolor="#FF3300" align="center"><span class="Sub-Titulo-
cobos">'. $datos6[$i][\'tx_estado_evento\'].\'</span></td>
        <td align="center"><span class="Sub-Titulo-
cobos">'. $datos6[$i][\'fe_fecha_estado\'].\'</span></td>
        </tr>';}

```

```

        else if($datos6[$i]['tx_tipo_estado_evento']=="NAGIOS" and
($datos6[$i]['tx_estado_evento']=="UP" ||
$datos6[$i]['tx_estado_evento']=="OK")){
            $Out.='<tr> <td align="center"><span class="Sub-Titulo-
cobos">'. $datos6[$i]['tx_tipo_estado_evento'] .'</span></td>
            <td bgcolor="#66FF00" align="center"><span class="Sub-Titulo-
cobos">'. $datos6[$i]['tx_estado_evento'] .'</span></td>
            <td align="center"><span class="Sub-Titulo-
cobos">'. $datos6[$i]['fe_fecha_estado'] .'</span></td>
            </tr>';}

        else if($datos6[$i]['tx_tipo_estado_evento']=="NAGIOS" and
$datos6[$i]['tx_estado_evento']=="WARNING"){
            $Out.='<tr> <td align="center"><span class="Sub-Titulo-
cobos">'. $datos6[$i]['tx_tipo_estado_evento'] .'</span></td>
            <td bgcolor="#FFCC00" align="center"><span class="Sub-Titulo-
cobos">'. $datos6[$i]['tx_estado_evento'] .'</span></td>
            <td align="center"><span class="Sub-Titulo-
cobos">'. $datos6[$i]['fe_fecha_estado'] .'</span></td>
            </tr>';}

        else if($datos6[$i]['tx_tipo_estado_evento']=="CIMOR" ||
$datos6[$i]['tx_tipo_estado_evento']=="SIGA"){
            $Out.='<tr> <td align="center"><span class="Sub-Titulo-
cobos">'. $datos6[$i]['tx_tipo_estado_evento'] .'</span></td>
            <td align="center"><span class="Sub-Titulo-
cobos">'. $datos6[$i]['tx_estado_evento'] .'</span></td>
            <td align="center"><span class="Sub-Titulo-
cobos">'. $datos6[$i]['fe_fecha_estado'] .'</span></td>
            </tr>';}
    }
    $Out.='
</table>
</div>
</div>';

```

```

$this->todo.=$Out;
}
}??>

```

procesar_eventos

```

<?php
require_once(dirname(__FILE__)."../../bdd/ConsultasEventos.php");
require_once(dirname(__FILE__)."../../bdd/ConsultasReportesCimor.php")
;

class procesar_eventos{
function procesar_eventos(){
$this->event=new consultas_eventos();
    $this->report=new ConsultasReportesCimor();
}

function Buscar_eventos_no_pendientes(){
$eventos=$this->event->buscar_eventos();
$c=0;
for($i=0;$i<count($eventos);$i++){
    $eventos_eliminados=$this->event-
>buscar_eliminados($eventos[$i]['co_evento']);
    if(empty($eventos_eliminados[0]['co_evento'])){
        $eventos_nuevos12=$this->event-
>buscar_nuevos($eventos[$i]['co_evento']);//BUSCO LOS QUE NO ESTEN
CERRADOS
        if(empty($eventos_nuevos12[0]['co_evento'])) //pregunta si esta
vacío coloca el evento sino no lo coloca
        {$eventos_nuevos12=$this->event-
>buscar_no_pendientes($eventos[$i]['co_evento']);
            if(empty($eventos_nuevos12[0]['co_evento'])){
                $arre_eventos1[$c]=$eventos[$i]['co_evento'];
                $c++;}
        }
}
}
}

```



```

}

return $arre_eventos1;

}

function Buscar_eventos_pendientes(){
$eventos1=$this->event->buscar_eventos();
$k=0;
for($i=0;$i<count($eventos1);$i++){
    $eventos_eliminados=$this->event-
>buscar_eliminados($eventos1[$i]['co_evento']);
    if(empty($eventos_eliminados[0]['co_evento'])){
        $eventos_nuevos1=$this->event-
>buscar_nuevos($eventos1[$i]['co_evento']);
        if(empty($eventos_nuevos1[0]['co_evento'])){
            $pendientes_nuevos=$this->event-
>colocar_pendientes($eventos1[$i]['co_evento']);
            if(!empty($pendientes_nuevos[0]['co_evento']))

                {$arre_eventos_pendientes[$k]=$pendientes_nuevos[0]['co_evento'];
                    $k++;}
        }
    }
}
return $arre_eventos_pendientes;

}

}
?>

```

En la figura 5.4 mostrada a continuación, se visualiza la página consultar eventos cerrados por código. La pantalla muestra los datos del evento que coincidan con la consulta a realizar por el usuario.

©PDVSA 2003 - Última actualización Abril de 2008

Figura 5.4. Pantalla Consultar Eventos Cerrados por Código: Área Eventos.

Fuente: Elaboración Propia.

5.2.2.2. Código Fuente Consultar Eventos Cerrados por Código

IUeventos_cerrados

```
<?php
require_once("IUAplicacion.php");
require_once(dirname(__FILE__)."/../dominio/procesareventos.php");
require_once(dirname(__FILE__)."/../dominio/ReceptoraEventos.php");

class IUeventos_cerrados extends Aplicacion{

function IUeventos_cerrados(){
    $this->proc_dat=new procesar_eventos();
}
}
```

```

function MostrarEventosCerrados(){

$datos=$this->proc_dat->proc_eventos_nuevos($_POST['txtnum']);
$datos6=$this->proc_dat-
>proc_evento_estado_evento($_POST['txtnum']);
$fechafin=$this->proc_dat->proc_fecha_fin($_POST['txtnum']);
$info=$this->proc_dat->retornar_informacion($_POST['txtnum']);

$Out.='<span class="Titulo-Applicacion">EVENTO CERRADO</span>

<span class="Separador_Modulo"></span>
<br></br>
<br></br>
<br></br>
<br></br>
<br></br>
<br></br>
<br></br>
<form name="formeventoscerrados" id="formeventoscerrados"
method="POST">
<input type="hidden" id="nevento" name="nevento"
value="'. $_POST['txtnum'].'">
<input type="hidden" id="fechaf" name="fechaf"
value="'. $fechafin[0]['fe_fecha_estado'].'">

<table width="100%" border="0">
<tr>
<td width="4%" class="tabla" align="center"><span class="Sub-
Titulo-cobose">CIMOR</span></td>
<td width="14%" class="tabla" align="center"><span class="Sub-
Titulo-cobose">Titulo</span></td>
<td width="14%" class="tabla" align="center"><span class="Sub-
Titulo-cobose">Equipo</span></td>
<td width="14%" class="tabla" align="center"><span class="Sub-
Titulo-cobose">Elemento</span></td>

```

```

<td width="9%" class="tabla" align="center"><span class="Sub-
Titulo-cobose">Fecha Inicio</span></td>
<td width="9%" class="tabla" align="center"><span class="Sub-
Titulo-cobose">Fecha Fin</span></td>
<td width="8%" class="tabla" align="center"></td>
<td width="4%" class="tabla" align="center"></td>
</tr>

<tr>
<td align="center"><span class="Sub-Titulo-
cobos">'. $datos[0]['tx_estado_cimor'] .'</span></td>
<td align="center"><span class="Sub-Titulo-
cobos">'. $datos[0]['tx_titulo'] .'</span></td>';
if($datos[0]['tx_tipo_evento']=='ADICIONAL')
$Out.='<td align="center"><span class="Sub-Titulo-
cobos">'. $datos[0]['tx_informacion_adicional'] .'</span></td>';
else if($datos[0]['tx_tipo_evento']=='MANUAL')
$Out.='<td align="center"><span class="Sub-Titulo-
cobos">'. $datos[0]['tx_equipo'] .'</span></td>';

$Out.='<td align="center"><span class="Sub-Titulo-
cobos">'. $datos[0]['tx_elemento'] .'</span></td>
<td align="center"><span class="Sub-Titulo-
cobos">'. $datos[0]['fe_fecha_evento'] .'</span></td>
<td align="center"><span class="Sub-Titulo-
cobos">'. $fechafin[0]['fe_fecha_estado'] .'</span></td>
<td align="center">
<input type="image" src="./iu/imagenes-aplicacion/editar.jpg"
width="28" height="28" onclick="validarcerradas(\'0\');">
<a href="#" onclick="validarcerradas(\'1\');"></a>
<input type="image" src="./iu/imagenes-aplicacion/documentar.jpg"
width="32" height="32" onclick="validarcerradas(\'2\');">
</td>

```

```

<td align="center">
<div class="Principio-Boton"></div>
<input id="boton" name="boton" class="Boton" type="button"
value="+" onclick="mostrarOcultarTablas(\`divtabla\`,
\`tabla_estado\`,\`boton\`);"/>
<div class="Final-Boton"></div>
</td>
</tr>
</table>
<br></br>
<div id="divtabla" name="divtabla" style="display: none">
<table class="estilo-tabla1" border="0">
<tr>
<td align="center" ><span class="Sub-Titulo-
cobose">Mantenimiento</span></td>
<td align="center"><span class="Sub-Titulo-
cobos">'. $datos[0]['tx_mantenimiento']. '</span></td>
<td align="center" ><span class="Sub-Titulo-
cobose">Impacto</span></td>
<td align="center"><span class="Sub-Titulo-
cobos">'. $datos[0]['tx_impacto_evento']. '</span></td>
</tr>
<tr>
<td align="center" ><span class="Sub-Titulo-cobose">Tipo
Equipo</span></td>
<td align="center"><span class="Sub-Titulo-
cobos">'. $datos[0]['tx_tipo_equipo']. '</span></td>
<td align="center" ><span class="Sub-Titulo-
cobose">IP</span></td>
<td align="center"><span class="Sub-Titulo-
cobos">'. $datos[0]['tx_ip']. '</span></td>
</tr>
<tr>
<td align="center" ><span class="Sub-Titulo-cobose">Tipo
Evento</span></td>

```

```

        <td align="center" ><span class="Sub-Titulo-
cobos">'. $datos[0]['tx_tipo_evento']. '</span></td>
        <td align="center" ><span class="Sub-Titulo-cobose">Tipo
Elemento</span></td>
        <td align="center"><span class="Sub-Titulo-
cobos">'. $datos[0]['tx_tipo_elemento']. '</span></td>
    </tr>
    <tr>
        <td align="center" ><span class="Sub-Titulo-
cobose">Localidad</span></td>
        <td align="center" width="140"><span class="Sub-Titulo-
cobos">'. $datos[0]['tx_localidad']. '</span></td>
        <td align="center"><span class="Sub-Titulo-
cobose">Direccion</span></td>
        <td align="center"><span class="Sub-Titulo-
cobos">'. $datos[0]['tx_direccion']. '</span></td>
    </tr>
    <tr>
        <td align="center"><span class="Sub-Titulo-
cobose">Disciplina</span></td>
        <td align="center" ><span class="Sub-Titulo-
cobos">'. $datos[0]['tx_grupo']. '</span></td>
        <td align="center"><span class="Sub-Titulo-
cobose">Tipificacion</span></td>
        <td align="center"><span class="Sub-Titulo-
cobos">'. $datos[0]['tx_tipificacion']. '</span></td>
    </tr>
    <tr>
        <td align="center"><a
href="javascript:mostrardiv(\'tabla_estado\');" class="contenedor-
opciones"><span class="Sub-Titulo-cobose-link">Ver mas
>>></span></a></td>
        <td></td>
        <td></td>
        <td></td>

```

```

    </tr>
</table>
<div id="tabla_estado" name="tabla_estado" style="display: none">
<table class="estilo-tabla2">
  <tr>
    <td align="center"><span class="Sub-Titulo-cobose">Tipo Estado
Evento</span></td>
    <td align="center"><span class="Sub-Titulo-cobose">Estado
Evento</span></td>
    <td align="center"><span class="Sub-Titulo-cobose">Fecha Estado
Evento</span></td>
  </tr>';
  for($i=0;$i<count($datos6);$i++){
    if($datos6[$i]['tx_tipo_estado_evento']=="CIMOR" ||
$datos6[$i]['tx_tipo_estado_evento']=="SIGA"){
      $Out.='<tr> <td align="center"><span class="Sub-Titulo-
cobos">'. $datos6[$i]['tx_tipo_estado_evento'] .'</span></td>
      <td align="center"><span class="Sub-Titulo-
cobos">'. $datos6[$i]['tx_estado_evento'] .'</span></td>
      <td align="center"><span class="Sub-Titulo-
cobos">'. $datos6[$i]['fe_fecha_estado'] .'</span></td>
      </tr>';}
}

$Out.='</table>
</div>
</div>
</form>
<br></br>
<br></br>';
$this->todo.=$Out;
}
}
?>

```

5.2.3. Área Reportes

Durante el modelado de la aplicación se definió el área de reportes con sus respectivas páginas. Esta vista está disponible para todos los actores que interactúan con el sistema, siempre y cuando posean una cuenta de acceso al mismo. En la figura 5.7 mostrada a continuación se observa la página de reportes estándar la cual muestra toda la información referente a las alarmas y/o los eventos registrados en el sistema desde una fecha inicio hasta una fecha fin, seleccionada por el usuario.

# Caso	NAGIOS	CIMOR	Titulo	Equipo	Elemento	Fecha
<input checked="" type="checkbox"/>	UP	PROGRESO	ES-GASIFICACION-CORE: Falla General	ES-GASIFICACION-CORE	UNICO	2010-04-27 09:02:45
<input checked="" type="checkbox"/>	CRITICAL	PROGRESO	ER-MUNPETROCURAGUA03: Falla General	ER-MUNPETROCURAGUA03	PING	2010-04-27 07:35:33
<input checked="" type="checkbox"/>	UP	PROGRESO	MA1PCU01: Falla General	MA1PCU01	UNICO	2010-04-26 14:56:57
<input checked="" type="checkbox"/>	UP	PROGRESO	PTX-5810: Falla General	PTX-5810	UNICO	2010-04-26 13:24:49
<input checked="" type="checkbox"/>	UP	PROGRESO	MATPCU02: Falla General	MATPCU02	UNICO	2010-04-24 07:20:51
<input checked="" type="checkbox"/>	DOWN	PROGRESO	ES-STMALMAC02PB01: Falla General	ES-STMALMAC02PB01	UNICO	2010-04-23 14:06:47

Figura 5.7 Pantalla Reporte Estándar: Área Reportes.

Fuente: Elaboración Propia.

5.2.3.1. Código Fuente Reporte Estándar

IUReportes

```
<?php
require_once("IUAplicacion.php");
```



```

require_once(dirname(__FILE__)."/../dominio/procesaralarmas.php");
require_once(dirname(__FILE__)."/../dominio/RecepReportesCimor.php");
;
require_once(dirname(__FILE__)."/../dominio/RecepGenerarPDF.php");
require_once(dirname(__FILE__)."/../dominio/ProcesarArreglos.php");

class reportes_cimor extends Aplicacion{

function reportes_cimor(){
$this->proc_dat=new procesar_alarmas();
$this->obj_ProcesarArreglos=new ProcesarArreglos();
}
function Reporte_Estandar(){
$obj_RecepReportesCimor=new RecepReportesCimor();
$arre_reporte=$obj_RecepReportesCimor->BuscarReporteEstandar();
if($arre_reporte==0 || $arre_reporte==1)
{if($arre_reporte==0){
    $this->MostrarErrorFecha();
    }
    else if($arre_reporte==1){
        $this->MostrarRangoFechaErroneo();
    }
}
$this->todo.=$Out;
}

else{
$this->MostrarReporteEstandar($arre_reporte);
}

}

function MostrarReporteEstandar($arre_reporte){
$obj_RecepReportesCimor=new RecepReportesCimor();
$Out.='<span class="Titulo-Aplicacion">REPORTE ESTANDAR</span>
<span class="Separador_Modulo"></span>

```

```

<br></br>
<br></br>
<form name="formver" id="formver" method="POST">
<input type="hidden" name="fe_inicio" id="fe_inicio"
value="'.$_POST['E_fechainicio'].'/>
<input type="hidden" name="fe_fin" id="fe_fin"
value="'.$_POST['E_fechafin'].'/>
<input type="hidden" name="usuario" id="usuario" value="'. $this-
>usuario.'"/>
<br></br>
<br></br>
<br></br>
<span class="Sub-Titulo-cobose">Seleccione el formato que desea
generar:</span>
<br></br>
<br></br>
<table>
<tr>
<td></td>
<td><div class="Principio-Boton"></div>
<input name="enviarpdf" id="enviarpdf" class="Boton" type="button"
value="Generar .PDF" onclick="generarnuevopdf();"/>
<div class="Final-Boton"></div>
<div class="Principio-Boton"></div>
<input name="enviarods" id="enviarods" class="Boton" type="button"
value="Generar .ODS" onclick="generar_ods('\4\');"/>
<div class="Final-Boton"></div></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>

```

```
<td></td>
<td></td>
</tr>
<tr>
<td width="20"><span class="Sub-Titulo-cobose">Para</span></td>
<td width="250"><input class="estilo-nuevo" name="para" id="para"
type="text" /></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td width="20"><span class="Sub-Titulo-cobose">De</span></td>
<td width="250"><input class="estilo-nuevo" name="de" id="de"
type="text" value="Centro Integral de Monitoreo Oriente" /></td>
<td></td>
</tr>
<tr>
<td width="20"><span class="Sub-Titulo-cobose">Asunto</span></td>
<td width="250"><input class="estilo-nuevo" name="asunto"
id="asunto" type="text" /></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
```

```

<td></td>
</tr>
</table>
<br></br>

<table class="estilo-tabla-alarmas" border="0">
<tr>
<td width="20" class="tabla" align="center"></td>
<td width="80" class="tabla" align="center"><span class="Sub-
Titulo-cobose">NAGIOS</span></td>
<td width="80" class="tabla" align="center"><span class="Sub-
Titulo-cobose">CIMOR</span></td>
<td width="200" class="tabla" align="center"><span class="Sub-
Titulo-cobose">Titulo</span></td>
<td width="200" class="tabla" align="center"><span class="Sub-
Titulo-cobose">Equipo</span></td>
<td width="200" class="tabla" align="center"><span class="Sub-
Titulo-cobose">Elemento</span></td>
<td width="100" class="tabla" align="center"><span class="Sub-
Titulo-cobose">Fecha</span></td>
<td width="40" class="tabla" align="center"></td>
<td width="40" class="tabla" align="center"></td>
</tr>
</table>
<div class="estilo-divscroll">
<input type="hidden" name="nevento" id="nevento" value="" />';
$this->todo.=$Out;
for($i=0;$i<count($arre_reporte);$i++){
$equipo=$obj_RecepReportesCimor-
>buscar_adicional($arre_reporte[$i][tipo_evento],
$arre_reporte[$i][co_evento]);
if(!empty($equipo)){
$arre_reporte[$i][equipo]=$equipo;
}
}

```

```

$estado=$obj_RecepReportesCimor-
>buscar_estado($arre_reporte[$i][co_evento]);
$this->AgregarTablas($arre_reporte,$i,$estado);
}
$Out1.='</form>
</div>
<br></br>';

$this->todo.=$Out1;
}
}
?>

```

La figura 5.8 muestra la página reporte por impacto, a partir de la cual se puede visualizar la gráfica generada de acuerdo a los datos solicitados por el usuario del total de alarmas en función del impacto asignado, igualmente muestra un gráfico de tortas con el total en porcentaje generado para cada impacto, y se puede visualizar una tabla con el total en números de cada impacto por fecha, y una tabla con el detalle de todas las alarmas generadas en esa fecha, y el impacto de cada una.



Figura 5.8. Pantalla Reporte por Impacto: Área Reportes.

Fuente: Elaboración Propia.

5.2.3.2. Código Fuente Reporte por Impacto

IUReportes.php

```

<?php
require_once("IUAplicacion.php");
require_once(dirname(__FILE__)."/../dominio/procesaralarmas.php");
require_once(dirname(__FILE__)."/../dominio/RecepReportesCimor.php")
;
require_once(dirname(__FILE__)."/../dominio/RecepGenerarPDF.php");
require_once(dirname(__FILE__)."/../dominio/ProcesarArreglos.php");

class reportes_cimor extends Aplicacion{

function reportes_cimor(){
$this->proc_dat=new procesar_alarmas();
$this->obj_ProcesarArreglos=new ProcesarArreglos();
}
function Reporte_Por_Impacto(){
switch($_GET['opcion1']){
    case 0:$this->Reporte_impactoFechainiciofin();
    break;
    case 1:$this->Reporte_impactofecha_estandar();
    break;

}
}

function Reporte_impactofecha_estandar(){
$obj_RecepReportesCimor=new RecepReportesCimor();
$pos=$obj_RecepReportesCimor->Buscar_Impacto_FechaEstandar();
$impacto=$this->proc_dat->retornar_impacto_evento();
if($pos==0){
$this->MostrarErrorFecha();
}
}
}

```

```

    }
else{
<form      name="form_generar_impacto"      id="form_generar_impacto"
method="post">
<input  type="hidden"  name="usuario"  id="usuario"  value="'.$this->
usuario.'" />
<div class="Principio-Boton"> </div>
<input  type="button"  class="Boton"  value="Generar  .PDF"
onclick="generar_pdf('\0\')"/>
<div class="Final-Boton"> </div>
<div class="Principio-Boton"> </div>
<input  type="button"  class="Boton"  Value="Generar  .ODS"
onclick="generar_ods('\0\')"/>'.//reporte ods impacto
'<div class="Final-Boton"> </div>
<br></br>
<br></br>
<table class="estilo-tabla-reporte">
<tr><td></td>
<td></td>
<td></td></tr>
<tr>
<td width="20"><span class="Sub-Titulo-cobose">Para</span></td>
<td  width="200"><input  class="estilo-nuevo"  name="para"  id="para"
type="text" /></td>
<td></td>
</tr>
<tr><td></td><td></td><td></td></tr>
<tr>
<td width="20"><span class="Sub-Titulo-cobose">De</span></td>
<td  width="200"><input  class="estilo-nuevo"  name="de"  id="de"
type="text"  value="Centro Integral de Monitoreo Oriente"/></td>
<td></td>
</tr>
<tr>
<td width="20"><span class="Sub-Titulo-cobose">Asunto</span></td>

```



```

<td width="200"><input class="estilo-nuevo" name="asunto"
id="asunto" type="text"/></td>
<td></td>
</tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
</table><br></br><br></br>';

    if(!empty($pos[0][nombre_impacto])){
        $Out.='<div id="estilo-graficas"></div>
<br></br>
<br></br>
<input type="hidden" name="impacto"
value="'. $pos[0][nombre_impacto].'">
<div class="estilo-divscroll">
<br></br>
<table class="estilo-tabla-alarmas">
<tr>
<td class="col" bgcolor="#E8E8E8" align="center"><span class="Sub-
Titulo-cobose">FECHAS</span></td>
<td class="col" bgcolor="#E8E8E8" align="center"><span class="Sub-
Titulo-cobose">'. $pos[0][nombre_impacto]. '</span></td>
</tr>';
        for($i=0;$i<count($pos);$i++){
            $Out.='<tr><td class="col" align="center"><span class="Sub-
Titulo-cobos">'. $pos[$i][fecha_inicio]. "
". $pos[$i][fecha_fin]. '</span></td>
            <td class="col" align="center"><span class="Sub-Titulo-
cobos">'. $pos[$i][cantidad]. '</span></td></tr>';
        }
    }
    $Out.='</table><br></br><br></br>';
    $cantidad=count($pos)-1;

```

```

$datos_tabla=$obj_RecepReportesCimor-
>buscar_datos_tabla($pos[0][inicio],          $pos[$cantidad][fin],
$pos[0][disciplina], $pos[0][localidad], $pos[0][nombre_impacto]);

    $Out.='<table class="estilo-tabla-alarmas">
                <tr><td                class="col"                bgcolor="#E8E8E8"
align="center"><span class="Sub-Titulo-cobose">Caso</span></td>
                <td                class="col"                bgcolor="#E8E8E8"
align="center"><span
class="Sub-Titulo-
cobose">Descripcion</span></td>
                <td                class="col"                bgcolor="#E8E8E8"
align="center"><span class="Sub-Titulo-cobose">Localidad</span></td>
                <td                class="col"                bgcolor="#E8E8E8"
align="center"><span
class="Sub-Titulo-
cobose">Disciplina</span></td>
                <td                class="col"                bgcolor="#E8E8E8"
align="center"><span class="Sub-Titulo-cobose">Causas</span></td>
                <td                class="col"                bgcolor="#E8E8E8"
align="center"><span class="Sub-Titulo-cobose">Impacto</span></td>
                </tr>';

    for($i=0;$i<count($datos_tabla);$i++){
        $Out.='<tr><td class="col" align="center"><span class="Sub-
Titulo-cobos">' . $datos_tabla[$i][caso] . '</span></td>
                <td class="col" align="center"><span
class="Sub-Titulo-
cobos">' . $datos_tabla[$i][descripcion] . '</span></td>
                <td class="col" align="center"><span
class="Sub-Titulo-cobos">' . $datos_tabla[$i][localidad] . '</span></td>
                <td class="col" align="center"><span
class="Sub-Titulo-
cobos">' . $datos_tabla[$i][disciplina] . '</span></td>';
                if(empty($datos_tabla[$i][causa])){

```

```

                                $Out.='<td      class="col"   bgcolor="#66CDAA"
align="center"><span      class="Sub-Titulo-cobos">Caso      sin
documentar</span></td>' ;}
                                else if(!empty($datos_tabla[$i][causa])){
                                $Out.='<td      class="col"   align="center"><span
class="Sub-Titulo-cobos">' . $datos_tabla[$i][causa] . '</span></td>' ;}

                                $Out.='<td      class="col"   align="center"><span
class="Sub-Titulo-cobos">' . $datos_tabla[$i][impacto] . '</span></td>
                                </tr>' ;

                                }
                                $Out.='</table></div>' ;

                                }
else if (empty($pos[0][nombre_impacto])){

                                $Out.='<div                                id="estilo-graficas"></div>
                                <br></br>
                                <br></br>
                                <input type="hidden" name="impacto" value="todos">
                                <div class="estilo-divscroll">
                                <br></br>
                                <table class="estilo-tabla-alarmas">
                                for($i=0;$i<count($pos);$i++){
                                $SUMA=0;
                                $Out.='<tr><td      class="col"   align="center"><span      class="Sub-Titulo-
cobos">' . $pos[$i][fecha_inicial] . "
                                " . $pos[$i][fecha_final] . '</td></td>' ;
                                for($j=0;$j<count($impacto);$j++){
                                $nombre=strtolower($impacto[$j]['tx_impacto_evento']);
                                $Out.='<td      class="col"   align="center"><span      class="Sub-Titulo-
cobos">' . $pos[$i][$nombre] . '</span></td>' ;

```

```

$SUMA=$SUMA+$pos[$i][$nombre];
}
$Out.='<td class="col" align="center"><span class="Sub-Titulo-
cobos">'. $SUMA. '</span></td></tr>';
}

$Out.='</table><br></br><br></br>';
$cantidad=count($pos)-1;
$datos_tabla=$obj_RecepReportesCimor-
>buscar_datos_tabla_todos($pos[0][inicio], $pos[$cantidad][fin],
$pos[0][disciplina], $pos[0][localidad]);

$Out.='<table class="estilo-tabla-alarmas">
      <tr><td class="col" bgcolor="#E8E8E8"
align="center"><span class="Sub-Titulo-cobose">Caso</span></td>
      <td class="col" bgcolor="#E8E8E8"
align="center"><span class="Sub-Titulo-
cobose">Descripcion</span></td>
      <td class="col" bgcolor="#E8E8E8"
align="center"><span class="Sub-Titulo-cobose">Localidad</span></td>
      <td class="col" bgcolor="#E8E8E8"
align="center"><span class="Sub-Titulo-
cobose">Disciplina</span></td>
      <td class="col" bgcolor="#E8E8E8"
align="center"><span class="Sub-Titulo-cobose">Causas</span></td>
      <td class="col" bgcolor="#E8E8E8"
align="center"><span class="Sub-Titulo-cobose">Impacto</span></td>
      </tr>';

for($i=0;$i<count($datos_tabla);$i++){
    $Out.='<tr><td class="col" align="center"><span class="Sub-
Titulo-cobos">'. $datos_tabla[$i][caso]. '</span></td>
      <td class="col" align="center"><span
class="Sub-Titulo-
cobos">'. $datos_tabla[$i][descripcion]. '</span></td>

```

```

                <td class="col" align="center"><span
class="Sub-Titulo-cobos">' . $datos_tabla[$i][localidad]. '</span></td>
                <td class="col" align="center"><span
class="Sub-Titulo-
cobos">' . $datos_tabla[$i][disciplina]. '</span></td>' ;
                    if(empty($datos_tabla[$i][causa])){
                        $Out.='<td class="col" bgcolor="#66CDAA"
align="center"><span class="Sub-Titulo-cobos">Caso sin
documentar</span></td>' ;}
                    else if(!empty($datos_tabla[$i][causa])){
                        $Out.='<td class="col" align="center"><span
class="Sub-Titulo-cobos">' . $datos_tabla[$i][causa]. '</span></td>' ;}
                    if($datos_tabla[$i][impacto]=="ALTO")
                        $Out.='<td class="col" bgcolor="#FF0000"
align="center"><span class="Sub-Titulo-
cobos">' . $datos_tabla[$i][impacto]. '</span></td>' ;
                    else if($datos_tabla[$i][impacto]!="ALTO")
                        $Out.='<td class="col"
align="center"><span class="Sub-Titulo-
cobos">' . $datos_tabla[$i][impacto]. '</span></td>' ;
                        $Out.='</tr>' ;
                    }
                $Out.='</table></div>' ;
            }
$matriz_impacto=$this->obj_ProcesarArreglos->arrayToString($pos);
$matriz_datos=$this->obj_ProcesarArreglos-
>arrayToString($datos_tabla);
$Out.='<input type="hidden" name="txt_matriz"
value="' . $matriz_impacto. '">
<input type="hidden" name="txt_datos_tabla"
value="' . $matriz_datos. '">
</form>
}
$this->todo.=$Out;

```

```

}
}
?>

```

ReceptoraReportes

```

<?php
require_once(dirname(__FILE__)."../bdd/ConsultasReportesCimor.php")
;
require_once("PCHART/GenerarGraficas.php");

class RecepReportesCimor {

function RecepReportesCimor(){
$this->obj_consultas_reportes_cimor=new ConsultasReportesCimor();

}

function Buscar_Impacto_Fechainiciofin(){
$obj_GenerarGraficas=new GenerarGraficas();
$valores=$this->comparar_fechas($_POST['txt_fechainicio_impacto'],
$_POST['txt_fechafin_impacto']);//verificar si la fecha inicio es
mayor que la fecha fin
if($valores==1){return 1;}

else{
if($_POST['selectimpacto']!="todos"){
if($_POST['localidad']=="todos"){
if($_POST['grupo_solucionador']=="todos"){
$fechas=$this-
>verificar_fechas($_POST['txt_fechainicio_impacto'],
$_POST['txt_fechafin_impacto']);
if($fechas==0){return 1;}
else{
for($i=0;$i<count($fechas);$i++){

```

```

        $impacto_unico=$this->obj_consultas_reportes_cimor-
>buscar_reporte_impacto_unico($fechas[$i][inicio], $fechas[$i][fin],
$_POST['selectimpacto']);

$arre_impactos[$i][fecha_inicio]=$fechas[$i][fecha_inicio];
    $arre_impactos[$i][fecha_fin]=$fechas[$i][fecha_fin];
    $arre_impactos[$i][inicio]=$fechas[$i][inicio];
    $arre_impactos[$i][fin]=$fechas[$i][fin];

$arre_impactos[$i][disciplina]=$_POST['grupo_solucionador'];
    $arre_impactos[$i][localidad]=$_POST['localidad'];

$arre_impactos[$i][nombre_impacto]=$_POST['selectimpacto'];

$arre_impactos[$i][cantidad]=$impacto_unico[0]['cantidad'];
    $suma=$suma+$arre_impactos[$i][cantidad];
    }
    if($suma==0){return 0;}
    else{
        $obj_GenerarGraficas->graficas_impacto_unico($arre_impactos,
$_POST['selectimpacto'],
        $_POST['grupo_solucionador'],
$_POST['localidad']);
    }
}
}
?>

```

5.3. Flujo de Trabajo de Pruebas

El flujo de trabajo Pruebas, es una de las fases más importantes del ciclo de vida de desarrollo de software. Un producto de software que se desarrolla se debe entregar al cliente libre de defectos o de errores. La prueba es el proceso de ejecutar la aplicación con la intención de descubrir defectos en el programa.

En el ciclo de vida de desarrollo de la aplicación, la fase de prueba ocurre después de la fase de programación. En tal sentido se plantea la revisión del funcionamiento de los distintos módulos de la aplicación, sometiendo cada uno a una entrada, proceso y salida esperada, a fin de verificar la confiabilidad, seguridad e integridad del sistema. Las pruebas garantizan que el producto es de calidad.

5.3.1. Modelo de Pruebas

Un modelo de prueba describe como se evalúan los componentes ejecutables del sistema y se realiza con la finalidad de encontrar errores a tiempo para su corrección. Para ello se implementarán un tipo de prueba muy común, la cual representa un caso de prueba que detalla cómo probar un caso de uso o un escenario específico y que incluye la verificación del resultado de la interacción entre actores y el sistema, representa el caso de prueba del sistema conocido como “caja negra”. Es importante especificar que esta prueba posee dos tipos: caja negra en lo pequeño y caja negra en lo grande.

5.3.2. Casos de Pruebas

Un caso de prueba especifica una forma de probar el sistema, incluyendo la entrada resultado con la que se ha de probar y las condiciones bajo las cuales ha de probarse. Existen dos casos de prueba comunes, prueba de “caja negra” y prueba de “caja blanca”.

Las pruebas de caja negra se concentran en los requisitos funcionales del software, es decir, permiten derivar un conjunto de condiciones de entrada que ejercitarán por completo todos los requisitos funcionales de un programa. La prueba de caja negra es un enfoque complementario al enfoque de caja blanca, debido a que provee la posibilidad de descubrir una clase diferente de errores.

En el caso del sistema desarrollado se trabajó realizando el caso de prueba especificado como caja negra, y se realizaron sólo para los casos de uso: Administrar Usuarios, Gestionar Servicios y Gestionar Alarmas.

5.3.2.2. Caso de prueba del caso de uso Administrar Usuarios

Entrada: Datos del usuario a ingresar.

Procedimiento de prueba:

1. Se ingresan los datos solicitados en la interfaz referente al nuevo usuario.
2. Se presiona el botón Aceptar.

Resultado: Se introdujo algunos campos sin completar el formulario, de manera de comprobar que realmente se evita que el actor deje campos vacíos, o inclusive ingrese datos errados para algún atributo, y que estos sean guardados en el almacén de datos del sistema.

Se verifico también que la interfaz funciona de forma correcta, y una vez se ejecuta el proceso de registro, se muestra en pantalla el mensaje de operación satisfactoria, y muestra la opción de volver a la pantalla anterior. La tabla 5.1 muestra las clases de equivalencias para la operación ingresar usuario.

5.3.2.3. Caso de prueba del caso de uso Gestionar Servicios

Entrada: Datos del servicio a ingresar.

Procedimiento de prueba:

1. Se ingresan los datos solicitados en la interfaz referente al nuevo servicio.
2. Se presiona el botón Aceptar.

3. En caso de teclear de forma errada algún dato se presiona el botón limpiar.

Resultado: Se introdujo algunos campos sin completar el formulario, de manera de comprobar que realmente se evita que el actor deje campos vacíos, o inclusive ingrese datos errados para algún atributo, y que estos sean guardados en el almacén de datos del sistema.

Luego de introducir estos datos y presionar el botón aceptar se muestra un mensaje de conclusión satisfactoria y una opción para regresar a la página anterior. La tabla 5.2 muestra la clase de equivalencia para la operación ingresar servicio.

Tabla 5.1. Clase de Equivalencia para Operación “Ingresar Usuario”.

Nº	Campo	Caso de Prueba	Salida
1	Indicador	Nulo	“Debe ingresar el campo Indicador”
2	Nombre	Nulo	“Debe ingresar el campo Nombre”
3	Nombre	José23678	“El campo Nombre debe estar compuesto por letras”
4	Nombre	José	“Datos correctos”
5	Apellido	Nulo	“Debe ingresar el campo apellido”
6	Clave	Nulo	“Debe crearle una clave al nuevo usuario”

Fuente: Elaboración Propia.

Tabla 5.2. Clase de Equivalencia para Operación “Ingresar Servicio”.

Nº	Campo	Caso de Prueba	Salida
1	Nombre	Nulo	“Debe ingresar el campo Nombre”
2	Fecha	Nulo	“Debe seleccionar una fecha inicio”
3	Elementos	Seleccione	“Debe seleccionar por lo menos un elemento de la lista”
4	Nombre	Internet	“Datos correctos”

Fuente: Elaboración Propia.

5.3.2.4. Caso de prueba del caso de uso Gestionar Eventos

Entrada: Datos generales, Datos documentación.

Procedimiento de prueba:

1. Se selecciona el tipo de evento que se desea ingresar.
2. Se presiona el botón Ingresar.
3. Se ingresan los datos solicitados en la interfaz referentes al tipo de evento a ingresar, y su respectiva documentación.
4. Se presiona el botón Aceptar.

Resultado: Se debe visualizar un mensaje que indique que ha sido almacenado el evento, y si falta por llenar algún campo de carácter obligatorio, se debe arrojar un mensaje especificando cual es el campo que falta por llenar.

La tabla 5.3 muestra la clase de equivalencia para la operación ingresar eventos.

Tabla 5.3. Clase de Equivalencia para Operación “Ingresar Servicio”.

Nº	Campo	Caso de Prueba	Salida
1	Fecha Inicio	Nulo	“Debe seleccionar una fecha inicio”
2	Título	Nulo	“Debe ingresar el campo título”
3	Título	Falla del equipo PLCGUA03	“Datos correctos”
4	Equipo	Seleccione	“Debe seleccionar un equipo”
5	Elemento	Seleccione	“Debe seleccionar un elemento”
6	Descripción	Nulo	“Debe ingresar el campo descripción”

Fuente: Elaboración Propia.

5.3.2.5. Caso de prueba del caso de uso Gestionar alarmas

Entrada: Código de la alarma cerrada.

Procedimiento de prueba:

1. Se ingresa el código de la alarma cerrada a consultar.
2. Se presiona el botón buscar.

Resultado: Al presionar el botón buscar se muestra una interfaz con los datos correspondientes a la información de la alarma, en caso de ingresar algún código errado el sistema mostrará un mensaje de error indicando que la alarma no ha sido cerrada, o bien que el código no corresponde a una alarma, o que esta ha sido eliminada.

La tabla 5.4 muestra la clase de equivalencia para la operación buscar alarma cerrada por código.

Tabla 5.4. Clase de Equivalencia para Operación “Buscar Alarma Cerrada por Código”.

Nº	Campo	Caso de Prueba	Salida
1	Número caso	Nulo	“Debe ingresar el número de caso de la alarma”
2	Númerocaso	4567	“El numero ingresado no corresponde a una alarma”
3	Número caso	“abcd”	“Formato incorrecto”
4	Número caso	3456	“La alarma no se ha cerrado”
5	Número caso	3455	“Alarma encontrada”

Fuente: Elaboración Propia.

5.3.3. Pruebas de Integración

Las pruebas de integración se utilizan para verificar que los componentes interaccionan entre sí de la forma apropiada después de haber sido integrados.

El sistema está basado en un conjunto de archivos HTML, PHP, PostgreSQL, Javascript y CSS, entrelazados entre si, por tal motivo, las pruebas de integración se realizaron probando todos y cada uno de los enlaces y simuladores presentes en el software.

Se navegó por todas las unidades haciendo link en cada una de las partes y procesos automatizados probando así su enlace y respuesta con las demás páginas, de igual manera se probaron los enlaces entre las páginas por medio de los botones *Adelante* y *Atrás de los navegadores internet explorer y firefox* y de los enlaces y botones *de volver y salir del menú de la aplicación*.

Después de haber realizado todas las pruebas, tanto de unidad como de integración, se hizo una revisión de los resultados de éstas y se determinó que el software ha superado las pruebas del sistema y que ha alcanzado la capacidad operativa inicial, por lo tanto no es necesario incluir otra iteración.

CAPÍTULO VI

FASE DE TRANSICIÓN

6.1. Introducción

El objetivo de la fase de transición es asegurar que los requisitos del cliente se hayan cumplido, ya que se centra en implantar el producto en su entorno de operación. Una vez que el proyecto entra en esta fase, significa que ha alcanzado la capacidad operativa inicial, aunque no sea necesariamente perfecto, puesto que aun en esta etapa se pueden evidenciar fallas que no se detectaron en la fase de construcción.

Los objetivos básicos de esta fase son:

- Cumplir los requisitos establecidos en las fases anteriores, hasta la satisfacción de todos los usuarios.
- Gestionar todos los aspectos relativos a la operación en el entorno del usuario, incluyendo la corrección de los defectos remitidos por los usuarios de la versión beta o por los encargados de las pruebas de aceptación.

En pocas palabras, el proyecto recibirá información de los usuarios para:

- Determinar si el sistema hace lo que demandan sus usuarios y el negocio.
- Descubrir riesgos inesperados.
- Anotar problemas no resueltos.
- Encontrar fallos.
- Eliminar ambigüedades y algunas en la documentación de usuario.

- Centrarse en áreas en las que los usuarios muestren deficiencias y necesiten información o formación.

6.2. Flujo de Trabajo de Implementación

Durante esta fase se lleva a cabo el trabajo de implementación, para tratar defectos tardíos como los encontrados en la versión beta del sistema.

6.2.1. Preparación de la versión beta

Para preparar la primera versión de SIDENCA a entregar, se seleccionaron sólo empleados de la Superintendencia de Mantenimiento de la Plataforma, entregándoles el manual de usuario del sistema y entrenándolos para su uso con la exposición del sistema y reuniones para aclarar dudas, de manera que los usuarios al familiarizarse con la aplicación fueran capaces de interactuar con ella sin ningún problema.

6.2.2. Instalación de la versión beta

El sistema se realizó en los servidores de desarrollo de la gerencia AIT del Edificio Sede Guaraguao en Puerto la Cruz, sin embargo para la necesaria aprobación del mismo debe pasar por los servidores de prueba y producción, donde personal de AIT serán los encargados de verificar y dar la aprobación definitiva del sistema para su posterior subida e implementación a los servidores de producción, todo esto con la respectiva autorización de la empresa.

6.3. Flujo de Trabajo de Pruebas

6.3.1. Pruebas a partir de la versión beta

A pesar de las normas a seguir por parte de la empresa para su posterior implementación, se hicieron un conjunto de pruebas con el personal seleccionado de la Superintendencia de Mantenimiento de la Plataforma con la versión realizada en

los servidores de desarrollo, puesto que esta Superintendencia fue la que solicitó el desarrollo del sistema.

Estas pruebas se llevaron a cabo tomando como base los mismos casos de pruebas realizados en la fase de construcción.

6.3.1.1. Reacción a los resultados de las pruebas

En la aplicación desarrollada se encontraron las siguientes fallas:

- En la pantalla Consultar Alarmas no se mostraba la cantidad de eventos en progreso y pendientes existentes, se estableció que debía colocarse una pequeña bandera (Notificación) que indicará al analista la cantidad de eventos en progreso y pendientes existentes, esto como recordatorio para el personal encargado.
- En el reporte estándar y búsqueda rápida se estableció que se debía colocar un campo adicional donde se muestre el código de cada alarma y evento.
- En el área de reporte por impacto se estableció que debía colocarse un campo adicional donde se muestren las causas que originaron la falla, es decir la descripción de la misma.
- Estas fallas fueron reportadas por personal seleccionado para las pruebas y fueron corregidas. El usuario quedó satisfecho con el sistema.

CONCLUSIONES

- Los requisitos funcionales y adicionales considerados en el proceso de desarrollo del sistema se implementaron de forma eficiente, superando las expectativas de las pautas establecidas por parte de la gerencia, gracias a las mejoras planteadas e implementadas en la fase de transición y construcción respectivamente, además de la interacción constante con los usuarios finales de la aplicación.
- El Proceso Unificado de Desarrollo de Software, y su herramienta WEBML permitió la realización de un software robusto, flexible y escalable, debido a que ofrece un modo estándar de visualizar, especificar, construir, documentar y comunicar los artefactos de un sistema.
- El poder tener todo lo referente a la conexión con la base de datos separado del resto del código permitió que el desarrollo de la aplicación fuera más fácil, incluyendo el mantenimiento de la misma. Se determinó el diseño de una Base de Datos relacional para el control y gestión de los datos, utilizando el modelo de datos de la especificación de WEBML.
- La utilización de software libre, cumpliendo con el decreto presidencial 3.390, es de gran importancia, ya que se obtiene software de calidad sin necesidad de comprar licencias. Además la utilización de PHP como lenguaje de programación para páginas Web y el manejador de bases de datos PostgreSQL, permitió el desarrollo de la aplicación de una manera más simple y efectiva, debido a que estas tecnologías son de fácil entendimiento e implementación.
- SIDENCA es un novedoso sistema que permite almacenar, visualizar, entre otras operaciones, las alarmas generadas por un sistema de monitoreo de equipos

utilizado por la gerencia de AIT Servicios Comunes Oriente de PDVSA Puerto la Cruz, el cual permitirá reducir el tiempo para la toma de decisiones, y una mayor agilidad de las actividades de mantenimiento a realizar por los diferentes grupos solucionadores.

RECOMENDACIONES

- Desarrollar módulos adicionales que integren el sistema realizado con el sistema automatizado de gestión de casos (SIGA), debido a que ciertas operaciones realizadas en ambos sistemas van entrelazadas, y de manera que la información en común se cargue y se mantenga en una sola base de datos.
- Realizar periódicamente un respaldo de la base de datos del sistema, así como también el mantenimiento respectivo, debido a que con este software se maneja gran cantidad de información
- Se recomienda que, en caso de realizarse la actualización de SIDENCA (una versión 2.0), se modifique los manuales del mismo.
- Incentivar el desarrollo e implementación de sistemas, para la automatización de las demás Superintendencias, que permitan incrementar la eficiencia de sus operaciones y minimizar los tiempos de respuestas en sus actividades diarias.

BIBLIOGRAFÍA

- Abad, J. (2008). Desarrollo de un sistema de monitoreo que permita la detección de fallas del sistema de información de tiempo real Petrocedeño. Trabajo de Grado no publicado. Universidad de Oriente, Anzoátegui, Venezuela.
- Alonso, S. (2006). Creación de sitios web con PHP 5. (1ª ed.). Barcelona, España: Creative Commons.
- Apache Software Foundation. (2009). [Página web en línea]. Disponible: <http://httpd.apache.org>.
- Barrientos, I, y Beites, J. (2006, Marzo). Nagios un Sistema de Monitorización de Servicios de Red. Ponencia presentada en la VI Jornada de software libre. Oviedo (Asturias).
- Binder, R. (2000). Testing Object-Oriented Systems: Models, Patterns and Tools. USA: Addison Wesley.
- Booch, G. (1999). Software Architecture and the UML. USA: Rational Software Corporation.
- Booch, G.; Jacobson, I y Rumbaugh, L. (1997). The UML Specification Documents. Santa Clara: Rational Software Corporation.
- Eguíluz, J. (2008). Introducción a XHTML. (1ª ed.). Barcelona, España: Creative Commons.

- Eguíluz, J. (2008). Introducción a Ajax. (1ª ed). Barcelona, España.
- Elmasri, R y Navathe, S. (1997). Sistemas de Base de datos. (2ª ed.). España: Addison Wesley.
- Garcia, G.; Villarroel, J. y Pernía, M. (2008). Manual de Adiestramiento Servicios Nivel 1. Universidad de Oriente, Sucre, Venezuela.
- Gobierno Bolivariano de Venezuela (2007). Ministerio del poder popular para las Telecomunicaciones y la informática. [Página web en línea]. Disponible en: <http://www.softwarelibre.gob.ve/index.php>
- Gutierrez, C. (2008). Como Funciona la Web. (1ª ed.). Chile.
- Laudon, K y Laudon, J. (1999). Administración de los Sistemas de Información, Organización y Tecnología. (3ª ed.). Barcelona, España.
- Medina, J. (2007). Desarrollo de un sistema basado en Aplicaciones Web para la automatización del control de pedidos asociados al proceso de ventas de una empresa cafetalera. Trabajo de Grado no publicado. Universidad de Oriente, Anzoátegui, Venezuela.
- Medina N, Nucette G, Seneco K, Yañes M, y Gomes, H. (2007). Confiabilidad Integral: Un Enfoque Practico. (1ª ed.). Caracas, Venezuela.
- Núñez, L. (2007). Desarrollo de una Aplicación Web para la visualización en tiempo real de los parámetros operacionales de producción de la empresa PDVSA. Trabajo de Grado no publicado. Universidad de Oriente, Anzoátegui, Venezuela.

- Ordaz, A. (2007). Desarrollo de un Software para la automatización y seguimiento de las actividades que se realizan en el almacén de un taller de mantenimiento aeronáutico. Trabajo de Grado no publicado. Universidad de Oriente, Anzoátegui, Venezuela.
- Politecnico Di Milano. Webml Site. (2003). [Página web en línea]. Disponible: <http://www.webml.org>.
- Salazar, P. (2007). Desarrollo de un Software para la automatización de reportes y consultas de archivos históricos del tráfico de conexiones de red realizadas por la Superintendencia de Seguridad Lógica de una empresa petrolera utilizando tecnología web. Trabajo de Grado no publicado. Universidad de Oriente, Anzoátegui, Venezuela.
- Schach, S. (2005). Análisis y Diseño Orientado a Objetos con UML y el Proceso Unificado. (1^{ra} ed.). Madrid, España: Mc Graw Hill.
- Sitio Oficial de PostgreSQL. (2009). [Página web en línea]. Disponible: <http://www.postgresql.org/about>.
- Stallman, R. (2004). Software libre para una sociedad libre. (1^{ra} ed.). Madrid, España.
- Tenias, J. (2007). Desarrollo de un Software basado en Aplicaciones Web para el monitoreo de dispositivos de la plataforma de Telecomunicaciones de PDVSA-GAS. Trabajo de Grado no publicado. Universidad de Oriente, Anzoátegui, Venezuela.

ANEXOS

A.1. CASO DE USO PROCESAR DATOS ALARMAS

Caso de Uso Procesar Datos Alarmas

En la figura mostrada a continuación se visualiza el caso de uso Gestionar Alarmas mostrado en la figura 3.2 del capítulo 3; se observa el caso de uso procesar datos alarmas, el cual se encarga de gestionar los datos de las alarmas generadas por el actor NAGIOS que en este caso representa la herramienta de monitoreo y se comporta como un actor activo, ya que por medio de este las alarmas, luego de ser validadas y procesadas por el caso de uso, son enviadas al actor BDD SIDENCA el cual actúa como un actor pasivo ya que es receptor de los datos y representa la base de datos del sistema. Los actores administrador del sistema y analista de monitoreo activan el caso de uso gestionar alarmas para visualizar las alarmas generadas. Luego que los datos son almacenados en la base de datos, el caso de uso gestionar alarmas se encarga de consultar la base de datos del sistema, cada cierto tiempo para verificar si existen alarmas nuevas y mostrarlas al usuario.

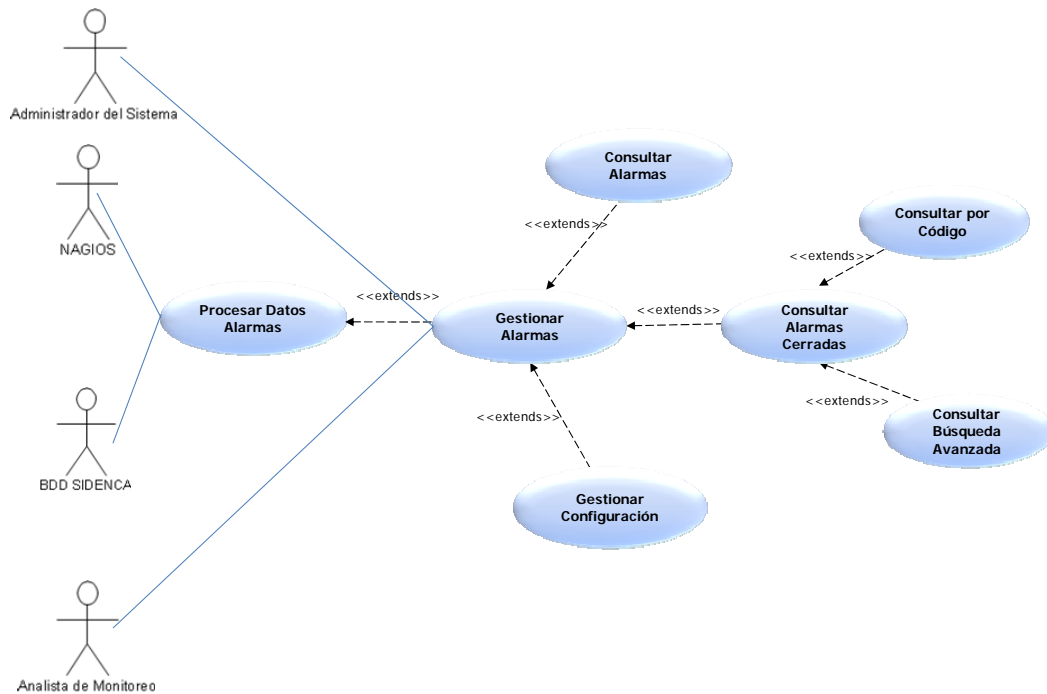


Figura 1. Caso de Uso Procesar Datos Alarmas

Fuente: Elaboración Propia.

Diagrama de Clase de Analisis para el Caso de Uso Procesar Datos Alarmas

La figura mostrada a continuación representa el diagrama de clase de analisis para el caso de uso procesar datos alarmas. El actor NAGIOS, representa a la herramienta de monitoreo encargada de generar los datos de las alarmas, estos datos son enviados via url al gestor procesar datos, el cual se encarga de verificar los datos enviados y de tomar la desición de mostrar la alarma o descartarla, las alarmas procesadas son enviadas al gestor procesar alarmas encargado de recibir los datos y transferirlos a la clase entidad alarmas la cual se encarga de realizar las operaciones necesarias para almacenar los datos en las respectivas tablas de la base de datos del sistema representada por el actor BDD SIDENCA.



Figura 2. Diagrama de Clase de Analisis para el Caso de Uso Procesar Datos Alarmas.

Fuente: Elaboración Propia.

A.2. MANUAL DE USUARIO SIDENCA

Contenido

Visión general del sistema Integrado de Notificación y comunicación de Alarmas

¿Qué es SIDENCA?

¿Qué módulos conforma SIDENCA?

Aspectos básicos de SIDENCA

Roles dentro de la aplicación

Acceso a SIDENCA

Sistema

Ingresar Usuario

Buscar Usuario

Modificar Usuario

Eliminar Usuario

Listar Usuarios

Alarmas

Consultar Alarmas

Editar Alarmas

Documentar Alarmas

Mostrar Alarmas Pendientes

Alarmas Cerradas

Búsqueda por Código

Búsqueda Avanzada

Editar Alarma Cerrada

Documentar Alarmas Cerrada

Configuración

Eventos

Ingresar Eventos

Evento Manual

Evento Adicional

Consultar Eventos

Editar Eventos

Documentar Eventos

Mostrar Eventos Pendientes

Eventos Cerrados

Búsqueda por Código

Búsqueda Avanzada

Editar Evento Cerrado

Documentar Evento Cerrado

Servicio

Ingresar Servicio

Consultar Servicio

Eliminar Servicio

Reportes

Estándar

Impacto

Mantenimiento

Equipo

Búsqueda Rápida

Visión general del sistema Integrado de Notificación y comunicación de Alarmas (SIDENCA)

¿Qué es SIDENCA?

El Sistema Integrado de Notificación y comunicación de Alarmas, es una aplicación desarrollada en software libre, cuya funcionalidad consiste en obtener las alarmas generadas por la herramienta de monitoreo NAGIOS en tiempo real, para su visualización y documentación por parte de los analistas que conforman el área del Centro de Monitoreo Oriente, Puerto la Cruz.

¿Qué Módulos Conforman SIDENCA?

El Sistema Integrado de Notificación y comunicación de Alarmas, está conformado por los siguientes módulos:

- Sistema
- Alarmas
- Eventos
- Servicios
- Reportes
- Ayuda

Aspectos básicos de SIDENCA

La aplicación SIDENCA tiene un modelo de flujo de trabajo, desde que es generada la alarma hasta el momento en que es documentada por primera vez, hasta que es resuelta y se puede cerrar el caso. A continuación explicaremos algunos aspectos básicos:

Roles dentro de la aplicación

Los usuarios directos de la aplicación SIDENCA son los Analistas del Centro Integral de Monitoreo Oriente (CIMOR), Administradores y Gerentes del área de Mantenimiento de la Plataforma (MAP).

Los roles identificados se describen a continuación:

Analistas del Centro Integral de Monitoreo Oriente (CIMOR): Es el grupo encargado de velar porque cada una de las alarmas sean atendidas por los diferentes grupos solucionadores y documentarlas antes de cerrarlas. Igualmente son los encargados de ingresar los eventos que se hayan generado en cualquier localidad, y documentarlos. Este grupo puede acceder a todos los módulos de la aplicación a excepción del módulo de Sistema.

Gerentes: Son las personas encargadas de generar informes y reportes diarios, semanales, mensuales o anuales de toda la información referente a las alarmas y los eventos.

Administradores: Son las personas encargadas de administrar SIDENCA. Entre sus funciones está la creación de cuentas a nuevos usuarios, y el acceso a todos los módulos de la aplicación

Acceso a SIDENCA

Si ya tiene clave de acceso, solo deberá llenar los campos que se muestran en la pantalla, y luego pulsar el botón Entrar



Pantalla de validación para ingresar al sistema

Sistema

Este modulo es manejado por los administradores de la aplicación. Permite gestionar el acceso a los usuarios del sistema.

Ingresar Usuario

Permite ingresar todos aquellos usuarios que pueden acceder a la aplicación incluyendo el rol de cada uno. En la figura mostrada a continuación se puede visualizar todos los campos que se deben llenar.

Buscar Usuario

Permite buscar los usuarios que se encuentran registrados en el sistema, ingresando el indicador de dicha persona, y visualizar todos sus datos, como: nombre, apellido, estatus, nivel, validación y observación.

©PDVSA 2003 - Última actualización Abril de 2008

Pantalla Ingresar Usuario

Modificar Usuario

Permite editar los datos de los usuarios que se encuentran registrados en el sistema ingresando el indicador del usuario, a excepción del indicador los campos que se pueden editar son los siguientes: Nombre, Apellido, Estatus Usuario, Tipo Validación, Nivel Usuario, Observación, y el campo Nueva Clave, está en caso de que el usuario quiera cambiar su clave de acceso por una nueva, en caso contrario el campo debe permanecer vacío.

Eliminar Usuario

Permite eliminar usuarios del sistema, ingresando el indicador y presionando el boton eliminar como se muestra en la pantalla siguiente. Si el usuario esta registrado la aplicación mostrara un mensaje de éxito.

Listar Usuarios

Permite visualizar todos los usuarios registrados en el sistema, así como la información referente a ellos. La siguiente figura muestra la pantalla listar usuarios.

Pantalla Modificar Usuario.

Alarmas

Este modulo es manejado por los Administradores y los Analistas del Centro Integral de Monitoreo Oriente (CIMOR), permite consultar las alarmas generadas por la herramienta de monitoreo NAGIOS en tiempo real, documentar, editar y eliminar. Es el modulo principal de SIDENCA y está diseñado para llevar un control de la documentación de todas las fallas generadas por los equipos que monitorea NAGIOS, y así poder generar reportes de las mismas.

Consultar Alarmas

Permite visualizar las alarmas generadas por NAGIOS, así como editar, documentar y eliminar. Las alarmas están ordenadas desde la más reciente hasta la fecha más antigua como se muestra en la siguiente figura.

STATUS	SEVER	Titulo	Equipo	Elemento	Fecha
OK	NUEVO	MANTMUI2: Falta General	MANTMUI2	DISCO D.	2010-05-06 10:14:13
OK	PROGRESO	MATPCU02: Falta General	MATPCU02	UPTIME	2010-05-05 21:00:25
OK	NUEVO	ER-PLCGUAS: Falta General	ER-PLCGUAS	ENLACE GUARADUARO - CE COLON PROYECTO ACF3	2010-05-05 06:19:42
OK	NUEVO	ER-PLCGUAS: Falta General	ER-PLCGUAS	ENLACE PLC - PETROGUARICO SERIAL11010	2010-05-04 12:33:57
NUEVO	NUEVO	PRIDE-319: Falta General	PRIDE-319	UNICO	2010-05-04 10:18:48
NUEVO	NUEVO	PRIDE-319: Falta General	PRIDE-319	UNICO	2010-05-04 10:18:46
NUEVO	NUEVO	PRIDE-319: Falta General	PRIDE-319	UNICO	2010-05-04 10:18:46

Pantalla Consultar Alarmas

Editar Alarmas

Permite modificar los datos de las alarmas, bien sea porque existe algún dato erróneo o por la falta de información de importante relevancia. Esta operación debe realizarla cada analista para verificar que todos los datos sean correctos antes de que la alarma sea cerrada. El usuario ingresa los datos que desee modificar y presiona el boton aceptar, en caso de que prefiera cancelar la operación presiona el boton cancelar y regresa a la pantalla consultar alarmas. La siguiente figura muestra la pantalla editar alarma.

Como se puede observar en la siguiente figura los campos que se pueden editar son los marcados por el asterisco (*).

The screenshot shows a web application interface for editing alarms. At the top, there is a header with the PDVSA logo and the text 'Sistema Integrado de Notificación y Comunicación de Alarmas'. Below the header, there is a navigation bar with tabs for SISTEMA, ALARMAS, EVENTOS, SERVICIO, REPORTES, AYUDA, and CERRAR SESION. The main content area is titled 'EDITAR ALARMAS' and contains a form with the following fields:

- Evento: 4933
- Fecha: 2010-05-06 15:16:13
- Titulo: MANTMUS: Falla General
- Descripcion Equipo: (empty)
- Descripcion Elemento: (empty)
- Equipo: MANTMUS
- Tipo Equipo: NO_DEFINIDO
- Impacto Evento: MEDIO
- Elemento: DISCO D
- Tipo Elemento: NO_DEFINIDO
- Mantenimiento: A CONDICION
- Localidad: ANACO
- Direccion: (empty)
- Disciplina: SERVIDORES
- Tipificacion: (empty)

Pantalla Editar Alarmas.

Los campos que no estén marcados no se pueden editar ya que son datos únicos provenientes de NAGIOS y no pueden ser modificados por reglamento de la empresa.

Los campos dirección y tipificación no se pueden editar ya que al seleccionar una localidad automáticamente la aplicación mostrara la respectiva dirección correspondiente a la misma, igualmente ocurre para el campo disciplina al seleccionar una de ellas se mostrara la tipificación correspondiente.

Documentar Alarmas

Permite al analista llevar un informe detallado del seguimiento que se realiza a cada falla, incluyendo el analista que atendió la falla, la descripción, el impacto y las acciones que se ejecutaron para solucionarla. De igual manera permite asignarle un estado a la alarma dependiendo de la situación en la cual se encuentre en ese momento.

Mostrar Alarmas Pendientes

Permite visualizar las alarmas que se encuentran en estado pendiente, ya que estas no se pueden ver en la tabla principal de alarmas. De igual manera permite editar y documentar cada una de las fallas.

Alarmas Cerradas

Permite visualizar las alarmas que han sido cerradas, esto para verificar si falta algún dato en la documentación, de ser así el analista puede volver a documentar, pero sin cambiar el estado de la alarma, ya que ha sido cerrada con anterioridad.

Búsqueda por Código

La búsqueda por código permite al usuario ingresar el número de caso de alarma, presiona el boton aceptar y el sistema se encargara de consultar los datos de dicha alarma y mostrarlos junto con las diferentes acciones que se pueden realizar, como editar, eliminar o documentar la alarma solucionada.

Estado	Código	Título	Equipo	Elemento	Fecha Inicio	Fecha Fin
CERRADO		SIPRE, Falla General	SIPRE	MEMORIA SIMP	2010-04-27 08:15:15	2010-05-14 17:08:11

Tipo Estado Evento

CERRADO
NADOS
CMOR
NADOS
CMOR

Fecha Estado Evento

2010-05-14 17:08:11
2010-04-27 12:08:30
2010-04-27 08:15:15
2010-04-27 08:15:15

Pantalla Búsqueda por código.

Editar Alarmas Cerradas

Permite modificar los datos de la alarma cerrada, ingresando los datos mostrados en la siguiente figura y presionando el boton aceptar.

Documentar Alarma Cerrada

Permite seguir documentando una falla después que se ha cerrado. Esta documentación se realiza en caso de que se haya hecho algún cambio luego de solucionada la falla. No permite cambiar el estado de la alarma, ya que el último estado de una alarma es cerrado.

Configuración

Permite gestionar las opciones localidad, disciplina, tipo equipo y tipo elemento, que corresponden a unos de los datos de las alarmas. La siguiente figura muestra la pagina ingresar opciones, a partir de la cual se puede editar, eliminar o ingresar una opción.

PDVSA Sistema Integrado de Notificación y Comunicación de Alarmas

Usuario: ESTHER MARIA CORO RAMIREZ Jueves 05 de Agosto de 2010

ADMINISTRADOR

SISTEMA ALARMAS EVENTOS SERVICIO REPORTES AYUDA CERRAR SESION

INGRESAR OPCIONES

Seleccione la opción que desea ingresar

Localidad
SATURIN

Disciplina
NO_DEFINIDO

Tipo Equipo
NO_DEFINIDO

Tipo Elemento
NO_DEFINIDO

© PDVSA 2003 - Última actualización Abril de 2008

Pantalla Ingresar Opciones.

Eventos

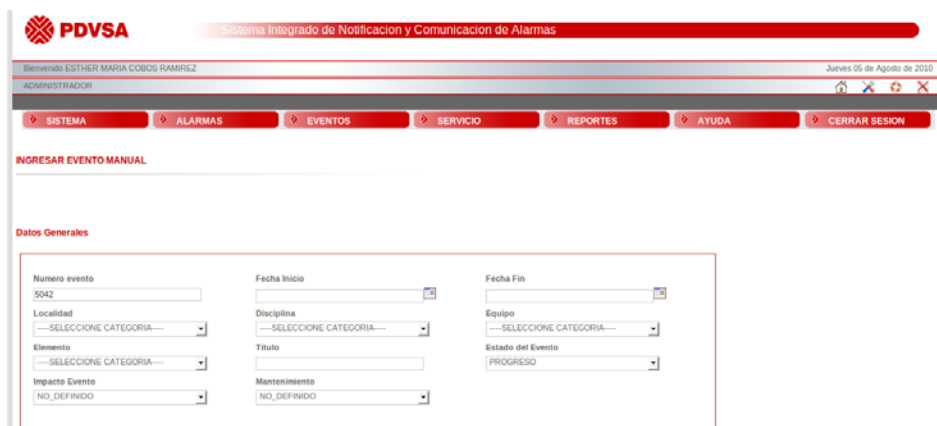
Este modulo es manejado por los Administradores y los Analistas del Centro Integral de Monitoreo Oriente (CIMOR), permite consultar los eventos ingresados por los analista. Estos eventos corresponden a fallas que no reporta NAGIOS debido a que algunos equipos no están registrados por no poseer IP, o porque simplemente es una falla que no ha sido detectada por la herramienta. Al igual que en el modulo de alarmas es posible documentar, editar y eliminar. Permite llevar un control de la documentación de todas las fallas ingresadas, y así poder generar reportes de las mismas.

Ingresar Eventos

Permite ingresar fallas no reportadas por NAGIOS. Los eventos ingresados pueden ser manuales o adicionales.

Evento Manual

Un evento manual es aquel que corresponde a una falla que no ha sido reportada por NAGIOS.

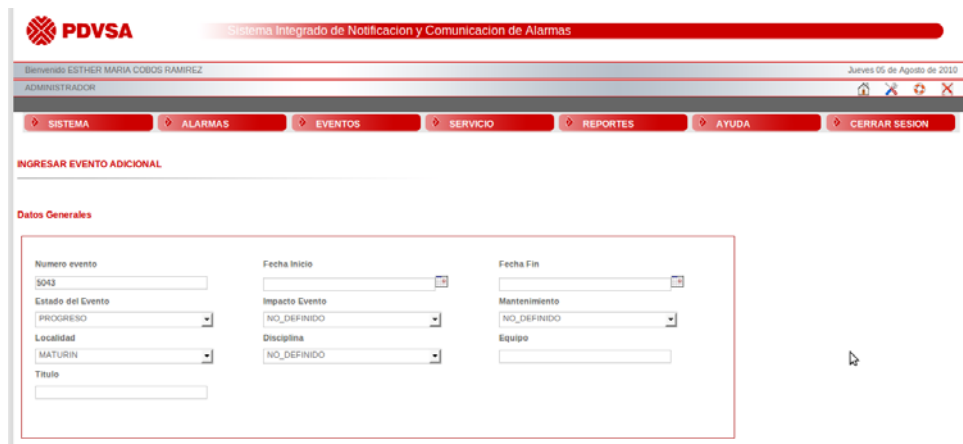


The screenshot shows a web application interface for PDVSA. At the top, there is a red header with the PDVSA logo and the text 'Sistema Integrado de Notificación y Comunicación de Alarmas'. Below the header, a navigation bar contains several menu items: SISTEMA, ALARMAS, EVENTOS, SERVICIO, REPORTES, AYUDA, and CERRAR SESION. The main content area is titled 'INGRESAR EVENTO MANUAL' and contains a form labeled 'Datos Generales'. The form has several fields: 'Numero evento' (text input with '5042'), 'Fecha Inicio' (calendar icon), 'Fecha Fin' (calendar icon), 'Localidad' (dropdown menu), 'Disciplina' (dropdown menu), 'Equipo' (dropdown menu), 'Elemento' (dropdown menu), 'Titulo' (text input), 'Estado del Evento' (dropdown menu with 'PROGRESO'), 'Impacto Evento' (dropdown menu with 'NO_DEFINIDO'), and 'Mantenimiento' (dropdown menu with 'NO_DEFINIDO').

Pantalla Ingresar Evento Manual

Evento Adicional

Un evento adicional es aquel que corresponde a una falla generada por un equipo que no se encuentra registrado en la herramienta de monitoreo, en este caso NAGIOS.



The screenshot shows a web application interface for PDVSA. At the top, there is a red header with the PDVSA logo and the text 'Sistema Integrado de Notificación y Comunicación de Alarmas'. Below the header, the user's name 'Esther ESTHER MARIA CORDO RAMIREZ' and the role 'ADMINISTRADOR' are displayed. A navigation menu contains buttons for 'SISTEMA', 'ALARMAS', 'EVENTOS', 'SERVICIO', 'REPORTES', 'AYUDA', and 'CERRAR SESION'. The main content area is titled 'INGRESAR EVENTO ADICIONAL' and contains a section for 'Datos Generales' with a form. The form includes the following fields: 'Numero evento' (text input with value '3043'), 'Fecha Inicio' (calendar icon), 'Fecha Fin' (calendar icon), 'Estado del Evento' (dropdown menu with 'PROGRESO' selected), 'Impacto Evento' (dropdown menu with 'NO_DEFINIDO' selected), 'Mantenimiento' (dropdown menu with 'NO_DEFINIDO' selected), 'Localidad' (dropdown menu with 'MATURIN' selected), 'Disciplina' (dropdown menu with 'NO_DEFINIDO' selected), 'Equipo' (text input), and 'Titulo' (text input).

Pantalla Ingresar Evento Adicional

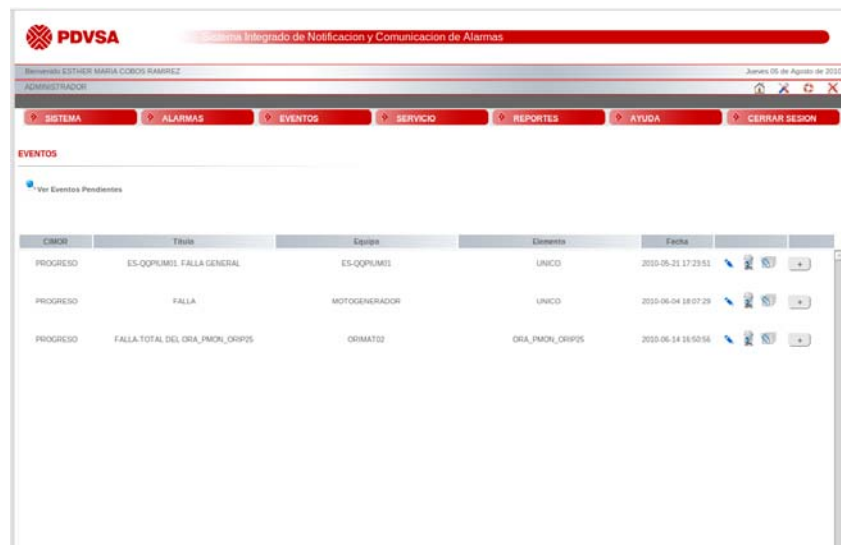
Consultar Eventos

Permite visualizar los eventos ingresados por los analistas, y realizar distintas operaciones como editar, documentar, eliminar y verificar más información referente al evento como la disciplina, localidad, tipo de equipo, tipo de elemento, los diferentes estados por los cuales a pasado el evento entre otros. Los eventos están ordenados desde los más recientes hasta la fecha más antigua.

Editar Eventos

Permite modificar los datos de los eventos, bien sea porque existe algún dato erróneo o por la falta de algún dato de importante relevancia. Esta operación debe realizarla cada analista para verificar que todos los datos sean correctos antes de que el evento sea cerrado, el usuario ingresa los datos a editar y presiona el botón aceptar en caso contrario puede presionar el botón cancelar y regresar a la pantalla consultar

eventos. A diferencia de editar alarmas, este caso requiere de dos tipos de formatos ya que existe tanto el evento de tipo manual como el tipo adicional, por lo tanto los campos a modificar no seran los mismos en los dos casos.



CMBOS	Titulo	Equipo	Elemento	Fecha			
PROGRESO	ES-QOPLUM1 FALLA GENERAL	ES-QOPLUM1	UNICO	2020-05-21 17:28:51			
PROGRESO	FALLA	MOTOGENERADOR	UNICO	2020-06-04 18:07:29			
PROGRESO	FALLA TOTAL DEL ORA_PMON_ORIP5	ORIMAT02	ORA_PMON_ORIP5	2020-06-14 16:50:54			

Pantalla Consultar Eventos.

Documentar Eventos

Permite al analista llevar un informe detallado del seguimiento que se realiza a cada falla, incluyendo el analista que atendió la falla, la descripción, el impacto y las acciones que se ejecutaron para solucionarla. De igual manera permite asignarle un estado al evento dependiendo de la situación en la cual se encuentre en ese momento.

Mostrar Eventos Pendientes


Permite visualizar los eventos que se encuentran en estado pendiente pulsando, ya que estos no se pueden ver en la tabla principal de eventos. De igual manera permite editar y documentar cada una de las fallas. La figura que se muestra a continuación permite visualizar la pantalla correspondiente a los eventos pendientes.

Eventos Cerrados

Permite visualizar los eventos que han sido cerrados, esto para verificar si falta algún dato en la documentación, de ser así el analista puede volver a documentar, pero sin cambiar el estado del evento, ya que ha sido cerrado con anterioridad.

Búsqueda por código

Para consultar los eventos el analista debe ingresar el numero de caso de la falla y pulsar el botón Buscar. La siguiente figura muestra los datos del evento consultado.



The screenshot shows the PDVSA system interface. At the top, there is a navigation menu with options: SISTEMA, ALARMAS, EVENTOS, SERVICIO, REPORTE, AYUDA, and CERRAR SESION. The main content area displays the details of a closed event (EVENTO CERRADO) in a table format.

CMOR	Título	Equipo	Elemento	Fecha Inicio	Fecha Fin
CERRADO	SABANA LARGA- FALLA GENERAL	SABANA LARGA	UNICO	2010-05-14 17:26:50	2010-05-24 10:53:51
Mantenimiento		A CONDICION	Impacto		MEDIO
Tipo Equipo		ADICIONAL	IP		127.0.0.1
Tipo Evento		MORICHAL	Tipo Elemento		NO_DEFINIDO
Localidad		APLICACIONES	Direccion		
Disciplina			Tipificacion		
Ver mas >>>					
	Tipo Estado Evento	Estado Evento	Fecha Estado Evento		
	CMOR	CERRADO	2010-05-24 10:53:51		
	CMOR	PROGRESO	2010-05-14 17:26:50		
	CMOR	NUOVO	2010-05-14 17:26:50		

©PDVSA 2003 - Última actualización Abril de 2008

Pantalla Búsqueda por código

Búsqueda Avanzada

Permite visualizar los últimos diez eventos cerrados desde el más reciente hasta el más antiguo, al igual que la búsqueda por código esta búsqueda también permite editar los eventos, documentarlos, o bien eliminarlos.

CERRAR	Título	Equipo	Elemento	Fecha Inicio	Fecha Fin	
CERRADO	SABANA LARGA- FALLA GENERAL	SABANA LARGA	UNICO	2010-05-14 17:26:50	2010-05-24 10:53:51	[Iconos de edición y eliminación]
CERRADO	MOTOGENERADOR 2-FALLA GENERAL	MOTOGENERADOR 2	UNICO	2010-05-08 11:56:29	2010-05-08 13:47:28	[Iconos de edición y eliminación]

Pantalla Búsqueda Avanzada

Editar Evento Cerrado

Permite modificar los datos del evento luego de haber sido solucionado.

Documentar Evento Cerrado

Permite seguir documentando una falla después que se ha cerrado. Esta documentación se realiza en caso de que se haya hecho algún cambio luego de solucionada la falla. No permite cambiar el estado del evento, ya que el último estado de un evento es cerrado.

Servicio

Este modulo permite gestionar todos los servicios utilizados por la Superintendencia de Mantenimiento de la Plataforma (MAP), por ejemplo el servicio de internet, correo electrónico entre otros. Está diseñado para verificar a futuro los servicios que han sido afectados por las fallas, para este caso solo se podrán ingresar, consultar, modificar y eliminar un servicio.

Ingresar Servicio

Permite ingresar un determinado servicio y los elementos que lo conforman. La siguiente figura muestra la pantalla con los campos que conforman la interfaz ingresar servicio.

Consultar Servicio

Permite consultar los servicios ingresados en el sistema, colocando el nombre del servicio a buscar.

Si el nombre ingresado corresponde a algún servicio almacenado en el sistema entonces el sistema mostrara los datos del servicio, como se muestra en la siguiente figura.

Elementos	Criticidad del Servicio
<input type="checkbox"/> /	CRITICO
<input type="checkbox"/> 45455454	CRITICO
<input type="checkbox"/> AB1784KT.EXE	CRITICO
<input type="checkbox"/> ACTUALIZASSP.EXE	CRITICO
<input type="checkbox"/> AC-VOLTAJE	CRITICO
<input type="checkbox"/> APACHE	CRITICO
<input type="checkbox"/> APACHE2	CRITICO
<input type="checkbox"/> APACHE.EXE	CRITICO
<input type="checkbox"/> ASPLMD.EXE	CRITICO
<input type="checkbox"/> ASYNCDLGP.EXE	CRITICO
<input type="checkbox"/> BPNBTD.EXE	CRITICO
<input type="checkbox"/> CIM-IO MANAGER	CRITICO

© PDVSA 2003 - Última actualización Abril de 2008

Pantalla Ingresar Servicio



Pantalla Ingresar Nombre: Consultar Servicio

Modificar Servicio

Permite editar un servicio almacenado en el sistema, incluyendo los elementos que integran dicho servicio. La figura siguiente muestra la pantalla con los datos de un servicio ya consultado.

Los campos que se pueden modificar son los elementos del servicio, la criticidad del elemento y la criticidad del servicio. Para cambiar algún elemento solo se debe deshabilitar el cuadro del elemento en cuestión y habilitar el cuadro que corresponda al elemento que se desea seleccionar.

Eliminar Servicio

Permite eliminar un servicio del sistema. Para eliminar un servicio solo debe ingresar el nombre y pulsar el botón Eliminar, si el nombre está asociado a algún servicio entonces se eliminara del sistema todos los datos que correspondan a dicho servicio.

Reportes

Este Modulo permite generar estadísticas, informes detallados y gráficas sobre la información de todas las alarmas y los eventos ingresados en el sistema.

Estándar

Este tipo de reporte muestra las alarmas y los eventos generados, de acuerdo a su estado CIMOR, entre una fecha inicio y fin ingresada por el usuario. Permite mostrar los datos en formato PDF y hojas de cálculo.

Para generar un reporte estándar solo debe ingresar una fecha inicio y una fecha fin, y seleccionar una opción del campo Tipo Caso y Estado del Caso.

Entre las opciones del campo Tipo Caso están:

- **Todas:** Permite mostrar todas las alarmas y los eventos adicionales y manuales.
- **Alarmas:** Permite mostrar las alarmas generadas por NAGIOS.
- **Eventos:** Permite mostrar todos los eventos.

Entre las opciones del campo Estado del Caso están:

- **Todos:** Muestra la alarmas y/o los eventos de todos los estados, ya sea nuevo, progreso, pendiente y cerrado.
- **Nuevo:** Muestra las alarmas que estén en estado nuevo.
- **Progreso:** Muestra las alarmas y/o los eventos que estén en estado progreso.
- **Pendiente:** Muestra las alarmas y/o los eventos que estén en estado pendiente.
- **Cerrado:** Muestra las alarmas y/o los eventos que estén en estado Cerrado.

La siguiente figura muestra la pantalla con los datos del reporte.

Para generar el reporte en formato pdf solo se deben seleccionar los casos que se requieran que aparezcan en la hoja, y descartar los que son irrelevantes, el usuario debe ingresar los campos para, de y asunto o de lo contrario el sistema mostrara un error indicando que se requieren dichos campos para generar el reporte, luego debe

presionar el boton generar .pdf y seguir las intrucciones para abrir o guardar el reporte.

PDVSA Sistema Integrado de Notificación y Comunicación de Alarmas

Bienvenido ESTHER MARIA CORCOB RAMIREZ Viernes 13 de Agosto de 2010

ADMINISTRADOR

SISTEMA ALARMAS EVENTOS SERVICIO REPORTES AYUDA CERRAR SESION

REPORTE ESTANDAR

Seleccione el formato que desea generar:

Generar .PDF Generar .ODS Cancelar

Para:

De: Centro Integral de Monitoreo Oriente

Asunto:

# Caja	NAGIOS	CIMOR	Titulo	Equipo	Elemento	Fecha		
<input checked="" type="checkbox"/>	3987	UP	PROGRESO	ES-GASIFICACION-CORE: Falta General	ES-GASIFICACION-CORE	UNICO	2010-04-27 09:02:45	
<input checked="" type="checkbox"/>	3977	CRITICAL	PROGRESO	ER-MUNPETROCURAGUA03: Falta General	ER-MUNPETROCURAGUA03	PING	2010-04-27 07:35:33	
<input checked="" type="checkbox"/>	3946	UP	PROGRESO	MATPCU01: Falta General	MATPCU01	UNICO	2010-04-26 14:56:57	
<input checked="" type="checkbox"/>	3943	UP	PROGRESO	PTX-5010: Falta General	PTX-5010	UNICO	2010-04-26 13:24:49	
<input checked="" type="checkbox"/>	3844	UP	PROGRESO	MATPCU02: Falta General	MATPCU02	UNICO	2010-04-24 07:20:51	
<input checked="" type="checkbox"/>	3766	DOWN	PROGRESO	ES-STMALMAC02PB01: Falta General	ES-STMALMAC02PB01	UNICO	2010-04-23 14:06:47	

Pantalla Mostrar Datos: Reporte Estándar

En la figura mostrada a continuación se puede visualizar el reporte en formato PDF.

Para generar el reporte en formato ods se debe seguir las mismas instrucciones para generar el formato pdf con la diferencia de que no es necesario llenar los campos para, de y asunto y el boton que se debe presionar es generar .ods.



COBOSE
FECHA: 16/04/2010 - 22/05/2010
HORA: 15:18:18 - 15:18:28

PARA: DOUGLAS ARAUJO
DE: Centro Integral de Monitoreo Oriente
ASUNTO: REPORTE ESTANDAR

ES-GASIFICACION-CORE: Falla General
ES-GASIFICACION-CORE - Caída de UNICO

3987 - ANACO
FECHA/HORA INICIO: 2010-04-27 09:02:45
ESTADO CIMGR: PROGRESO

DESCRIPCION

Se anexa evidencia:
PING CRITICAL - Packet loss = 100% 2010-04-27 09:02:45

ER-MUNPETROCURAGUA03: Falla General
ER-MUNPETROCURAGUA03 - Caída de PING

3977 - PETROCURAGUA
FECHA/HORA INICIO: 2010-04-27 07:35:33
ESTADO CIMGR: PROGRESO

DESCRIPCION

Se anexa evidencia:
PING CRITICAL - Packet loss = 66%, RTA = 1691.13 ms 2010-04-27 07:35:33

MAIPCU01: Falla General
MAIPCU01 - Caída de UNICO

3946 - PETROCURAGUA
FECHA/HORA INICIO: 2010-04-26 14:56:57
ESTADO CIMGR: PROGRESO

DESCRIPCION

Se anexa evidencia:
CRITICAL - Time to live exceeded (10.42.72.11) 2010-04-26 14:56:57

PTX-5810: Falla General
PTX-5810 - Caída de UNICO

Page 1/5

Pantalla PDF: Reporte Estándar

La siguiente figura muestra el reporte en formato Open Calc.

	B	C	D	E	F	G	H	
	EVENTO	ESTADO NAGIOS	ESTADO CIMOB	TITULO	EQUIPO	ELEMENTO	FECHA	TIPO EVENTO
2	3987	UP	PROGRESO	ES-GASIFICACION-COBE: Falla General	ES-GASIFICACION-COBE	UNICO	2010-04-27 09:02:45	AUTOMATICO
3	3977	CRITICAL	PROGRESO	ER-MUNPETROCURAGUA03: Falla General	ER-MUNPETROCURAGUA03	PING	2010-04-27 07:35:33	AUTOMATICO
4	3946	UP	PROGRESO	MATPCU01: Falla General	MATPCU01	UNICO	2010-04-26 14:56:57	AUTOMATICO
5	3943	UP	PROGRESO	PTX-5010: Falla General	PTX-5010	UNICO	2010-04-26 13:24:49	AUTOMATICO
6	3844	UP	PROGRESO	MATPCU02: Falla General	MATPCU02	UNICO	2010-04-24 07:20:51	AUTOMATICO
7	3766	DOWN	PROGRESO	ES-STMALMAC02PB01: Falla General	ES-STMALMAC02PB01	UNICO	2010-04-23 14:06:47	AUTOMATICO
8	3765	DOWN	PROGRESO	EVDR-MATERIALES1Y2: Falla General	EVDR-MATERIALES1Y2	UNICO	2010-04-23 14:05:20	AUTOMATICO
9	3716	UP	PROGRESO	ES-STMCOODEPE10: Falla General	ES-STMCOODEPE10	UNICO	2010-04-23 09:31:31	AUTOMATICO
10	3545	DOWN	PROGRESO	GUARAGUAO-VEPICA: Falla General	GUARAGUAO-VEPICA	UNICO	2010-04-21 15:11:07	AUTOMATICO
11	3484	DOWN	PROGRESO	EFLMUSI_SCA1S2: Falla General	EFLMUSI_SCA1S2	UNICO	2010-04-19 06:48:57	AUTOMATICO
12	3432	DOWN	PROGRESO	STOLNA02: Falla General	STOLNA02	UNICO	2010-04-17 04:37:40	AUTOMATICO
13	3400	DOWN	PROGRESO	SSAPBW01: Falla General	SSAPBW01	UNICO	2010-04-16 09:07:57	AUTOMATICO
14	3372	UP	PROGRESO	EHTV-ACOMGUAASANTA: Falla General	EHTV-ACOMGUAASANTA	UNICO	2010-04-15 21:21:19	AUTOMATICO
15	3440	DOWN	PROGRESO	PLC0543: Falla General	PLC0543	UNICO	2010-04-12 15:41:22	AUTOMATICO
16	3399	CRITICAL	PROGRESO	MOROWS2: Falla General	MOROWS2	MCSHIELD.EXE	2010-04-16 08:08:32	AUTOMATICO
17	3561	CRITICAL	PROGRESO	MNDOWS8: Falla General	MNDOWS8	MCSHIELD.EXE	2010-04-21 17:20:55	AUTOMATICO
18								
19								
20								
21								
22								
23								
24								
25								
26								
27								
28								
29								
30								
31								
32								
33								

Pantalla Hoja de Cálculo: Reporte Estándar

Impacto

Permite generar un reporte detallado, y gráficas de todos los casos ya sean alarmas y eventos con impacto y sin impacto, seleccionando una localidad y disciplina o bien todas las localidades y todas las disciplina. El usuario puede seleccionar entre una fecha inicio fin o una fecha específica. La diferencia entre estos dos tipos de fechas es que la fecha inicio fin se selecciona a partir de un calendario, y la fecha específica se selecciona a partir de tres opciones como son: el último mes, la ultima semana y las últimas 24 horas. La siguiente figura muestra la pantalla para generar un reporte por impacto partiendo de una fecha inicio y una fecha fin.

PDVSA Sistema Integrado de Notificación y Comunicación de Alarmas

Bienvenido ESTHER MARÍA COBOS RAMÍREZ Viernes 13 de Agosto de 2010

ADMINISTRADOR

SISTEMA ALARMAS EVENTOS SERVICIO REPORTE AYUDA CERRAR SESION

REPORTE POR IMPACTO

Seleccione el tipo de fecha que desea:

Fecha Inicio Fecha Fin
 Fecha Especifica

Fecha Inicio Fecha Fin Impacto

Localidad Disciplina

TODAS TODAS TODOS

Generar Reporte

©PDVSA 2003 - Última actualización Abril de 2008

Pantalla Fecha Inicio Fecha Fin: Reporte Por Impacto

De acuerdo a la figura anterior es posible seleccionar una fecha inicio y una fecha fin, de allí se desplegará un calendario del cual se podrá escoger la fecha completa incluyendo la hora. Entre los diferentes impactos que se pueden escoger están: ALTO, MEDIO, BAJO, NO_DEFINIDO y TODOS. El campo localidad y disciplina permite seleccionar una categoría de entre las que se encuentran almacenadas en el sistema, incluyendo TODAS.

A diferencia de la opción fecha inicio fecha fin la opción fecha específica permite seleccionar diferentes rangos de fechas ya definidos por el sistema como: última semana, último mes, últimas 24 horas, estos rangos de fecha se toman desde el día que está corriendo actualmente hasta los días que ya pasaron dependiendo de la opción que se escoja por ejemplo si la opción es las últimas 24 horas y el día actual es el 24/05/2010 12:00:00, la fecha inicio sería 23/05/2010 12:00:00 y la fecha fin 24/05/2010 12:00:00.

La siguiente figura muestra la gráfica de todos los impactos, todas las localidades, todas las disciplinas desde el 16/04/2010 – 21/05/2010. La línea de color morado muestra el total de todos los impactos por fecha. El eje de las X representa las fechas y el eje de las Y representa la cantidad de cada impacto incluyendo el total.

La tabla mostrada en la parte superior de la pantalla siguiente indica el resultado de la gráfica de la figura pantalla grafica, indicando la fecha por cada impacto y el total. La tabla que se encuentra debajo muestra todos los casos generados en esa fecha incluyendo su número de caso, descripción, localidad, disciplina, causa de la falla y el impacto, las celdas marcadas en color rojo indican que el impacto es ALTO. En el campo causa, las celdas marcadas con color verde muestran que aun el caso no ha sido documentado.

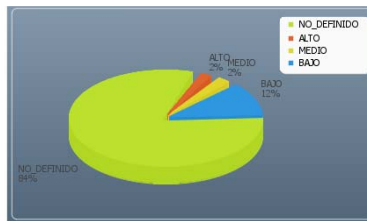
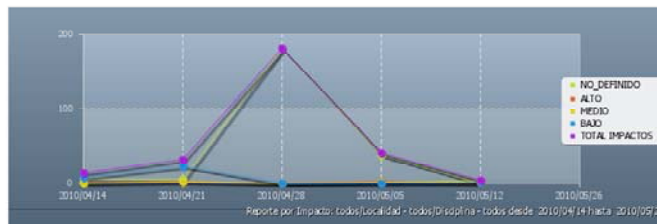
Al igual que el reporte estándar, este reporte permite generar los datos en formato PDF y hojas de cálculo. Para generar el reporte en formato pdf, el usuario debe llenar los campos para, de y asunto y presionar el boton generar pdf, en caso de no ingresar los campos antes mencionados, el sistema generara un mensaje de error indicando que dichos campos deben llenarse. Para generar el reporte en formato ods (Open Calc) solo se debe presionar el boton generar ods. Las figuras mostradas a continuación presentan el reporte en formato pdf y ods.

REPORTE POR IMPACTO

Seleccione el formato que desea generar:

Generar .PDF Generar .ODS Cancelar

Para:
De: Centro Integral de Monitoreo Oriente
Asunto:



FECHAS	NO_DEFINIDO	ALTO	MEDIO	BAJO	TOTAL IMPACTOS
2010/04/14 - 2010/04/21	5	1	0	8	14
2010/04/21 - 2010/04/28	5	2	2	23	32
2010/04/28 - 2010/05/05	181	0	0	0	181
2010/05/05 - 2010/05/12	38	2	1	0	41
2010/05/12 - 2010/05/26	0	0	3	1	4

Caso	Descripcion	Localidad	Disciplina	Causas	Impacto
3287	ES_TREBOL_01: Falta General	CUMANA	REDES	Se visualiza alarma critica del switch es-trebol-01 IP 10.170.32.1 y de igual manera el sensor de temperatura eht-onaedftrbol01 IP 10.170.34.131. se anexa evidencia: 14/04/2010 05:58:34 PM Critical es-trebol-01 10.170.32.1 DEVICE HAS STOPPED RESPONDING TO POLLS 14/04/2010 05:58:02 PM Critical eht-onaedftrbol01 10.170.34.131 DEVICE HAS STOPPED RESPONDING TO POLLS Se llama a la analista Karina Rengel AIT Cumana para ponerla al tanto.	NO_DEFINIDO
3378	WGB350_30ABE0_OROC4: Falta General	MATURIN	TRANSPORTE	Se visualiza alarma critica del enlace de radio ESEM -ORO, se anexa evidencia: 16/04/2010 02:46:52 AM Critical WGB350_462824 170.179.247.138 DEVICE HAS STOPPED RESPONDING TO POLLS 16/04/2010 02:46:43 AM Critical WGB350_30ABE0_OROC4 170.179.247.138 DEVICE HAS STOPPED RESPONDING TO POLLS 16/04/2010 02:47:20 AM Critical ESEM - ORO WIDE-AREA LINK CONTACT LOST 16/04/2010 02:47:20 AM Critical ORO-ESEM WIDE-AREA LINK CONTACT LOST Se llama al Sr. Angel Villanueva Coord AIT FUR para ponerlo al tanto e informara que llamara a al personal de telecom.	ALTO
3418	GW-63: Falta General	NO_DEFINIDO	NO_DEFINIDO	Caso sin documentar	NO_DEFINIDO
3533	GW65: Falta General	NO_DEFINIDO	NO_DEFINIDO	Caso sin documentar	NO_DEFINIDO
3532	SDS-97: Falta General	PUERTO LA CRUZ	TRANSPORTE	Caso sin documentar	NO_DEFINIDO
3630	EHTV-CCMT-TCOMSUCRE: Falta General	PETROSUCRE	INFRAESTRUCTURA	Se observa alta temperatura en sensor ehtv-comt-tomsucre con IP:10.60.132.5. Se anexa Evidencia tecnica: Temperatura CRITICAL 22-04-2010 13:00:10 del 08:43m 27% 22 Temperatura CRITICAL - 24° Grados Se llamo al personal de guardia Petrosucre Adielia Morales quedo en llamar al personal de mantenimiento para que revisara. Fecha y Inicio:22-04-2010 12:17 >>>> bernandezmtx 22/04/2010 13:47 Se llamo a Adielia Morales (Petrosucre) ella informa que el aire acondicionado de esa Area no esta entrando bien y el llamar esta	BAJO

Pantalla Reporte Por Impacto.

Mantenimiento

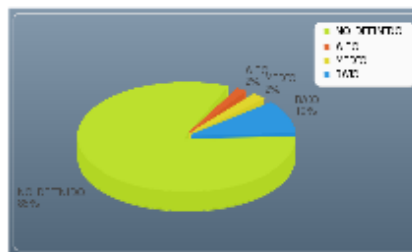
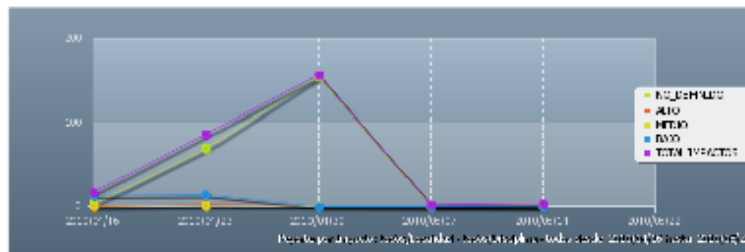
Permite generar estadísticas, y gráficas de todos los casos ya sean alarmas y eventos de acuerdo a el tipo de mantenimiento que se le haya realizado o que se deba realizar a una falla, seleccionando una localidad y todas las disciplina o bien todas las localidades y una disciplina. El usuario puede seleccionar entre una fecha inicio fin o una fecha específica.

	A	B	C	D	E	F	G
	FECHA	NO. DEFINIDO	ALTO	MEDIO	BAJO	TOTAL IMPACTOS	
2	2010/04/15 - 2010/04/23	3	0	1	13	17	
3	2010/04/23 - 2010/04/30	69	2	1	14	86	
4	2010/04/30 - 2010/05/07	154	1	1	0	156	
5	2010/05/07 - 2010/05/14	1	1	0	0	2	
6	2010/05/14 - 2010/05/22	0	0	3	0	3	
7							
8							
9							
10							
11							
12							
13	CASO	DESCRIPCION	LOCALIDAD	DISCIPLINA	CAUSAS	IMPACTO	
14	3418	GW-63: Falla General	NO. DEFINIDO	NO. DEFINIDO	Caso sin documentar	NO. DEFINIDO	
15	3533	GW65: Falla General	NO. DEFINIDO	NO. DEFINIDO	Caso sin documentar	NO. DEFINIDO	
16	3532	SEPS-97: Falla General	PUERTO LA CRUZ	TRANSPORTE	Caso sin documentar	NO. DEFINIDO	
17	3630	HTV-CC-MT-TC-GOMSU-RE: Falla General	PETROSUCRE	INFRAESTRUCTURA	Faltas al personal de mantenimiento para que revise	BAJO	
18	3618	PZS-POSUB07-01: Falla General	PETROQUATA	REDES	Evidencia técnica: PING CRITICAL: 22-04-2010 09:09:00	BAJO	
19	3637	ES-STMDESURBP01: Falla General	SAN TOME	REDES	Evidencia técnica: PING CRITICAL: 22-04-2010 14:09:00	BAJO	
20	3616	ES-SALAGUAMACHE: Falla General	GUAMACHE	REDES	C. CRITICAL: 22-04-2010 09:13:09 0d 1h 32m 7s 100ms	BAJO	
21	3640	ER-MUNMEM01: Falla General	MATURIN	REDES	WUR y se habla con el analista Armando Rivas, que	MEDIO	
22	3819	FR-PDW-ENS069: Falla General	PETROSUCRE	REDES	8:05:50 0d 1h 17m 75s CRITICAL: Serial0:1:0: PC	ALTO	
23	3432	STORNA02: Falla General	PETROBTUPANO	SERVIDORES	Caso sin documentar	BAJO	
24	3484	EFLMUN-SC-A1S2: Falla General	PUNTA DE MATA	AUTOMATIZACION	Caso sin documentar	BAJO	
25	3545	GUARAGUAO-VEPCA: Falla General	PUERTO LA CRUZ	TRANSPORTE	Caso sin documentar	BAJO	
26	3716	ES-STMCOODEPE10: Falla General	SAN TOME	REDES	Caso sin documentar	BAJO	
27	3765	EVDOR-MATERIALES1Y2: Falla General	SAN TOME	PCP	Caso sin documentar	BAJO	
28	3766	ES-STMALMACIO2PB01: Falla General	SAN TOME	REDFS	Caso sin documentar	BAJO	
29	3844	MATPCU02: Falla General	PETROCRAGUA	SERVIDORES	Caso sin documentar	BAJO	
30	3943	PTX-5810: Falla General	PUERTO LA CRUZ	TRANSPORTE	Caso sin documentar	BAJO	
31	3946	MATPCU01: Falla General	PETROCRAGUA	SERVIDORES	Caso sin documentar	BAJO	
32	3977	MUNPETROCRAGUA03: Falla General	PETROCRAGUA	REDES	Caso sin documentar	BAJO	
33	3981	SIPRE: Falla General	PUERTO LA CRUZ	SERVIDORES	Caso sin documentar	MEDIO	

Pantalla OpenCalc: Reporte por Impacto



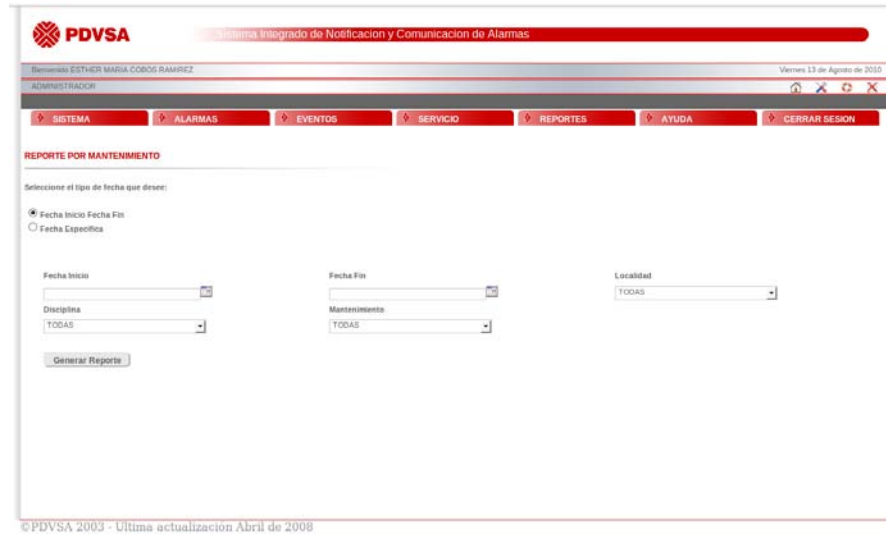
PARA: DOUGLAS ARAUJO
DE: Centro Integral de Monitoreo Oriente
ASUNTO: REPORTE POR IMPACTO



FECHA	NO_DEFINIDO	ALTO	MEDIO	BAJO	TOTAL IMPACTO
20100416 - 20100423	3	0	1	0	4
20100423 - 20100430	49	2	1	14	66
20100430 - 20100507	204	1	1	0	206
20100507 - 20100514	1	1	0	0	2
20100514 - 20100522	0	0	3	0	3

Pantalla PDF: Reporte Por Impacto

Si el usuario selecciona la opción fecha inicio fecha fin la siguiente figura muestra la pantalla con los datos referentes a esta opción.



Pantalla Fecha Inicio Fecha Fin: Reporte Por Mantenimiento

La siguiente figura muestra una gráfica de mantenimiento correctivo, de la localidad de puerto la cruz y todas las disciplinas, las barras que muestra la gráfica representan todas las disciplinas de la localidad de puerto la cruz, el eje de las X representa las fechas y el eje de las Y representa la cantidad de mantenimiento para cada disciplina, en este caso cuantos mantenimientos correctivos a realizado cada disciplina.

La figura mostrada a continuación indica el resultado de la gráfica, indicando la fecha por cada mantenimiento y el total por cada disciplina.

PDVSA Sistema Integrado de Notificación y Comunicación de Alarmas

Bienvenido ESTHER MARIA COBOS RAMIREZ
ADMINISTRADOR
Viernes 13 de Agosto de 2009

SISTEMA ALARMAS EVENTOS SERVICIO REPORTES AYUDA CERRAR SESION

REPORTE POR MANTENIMIENTO

Seleccione el formato que desea generar:

Generar PDF Generar ODS Cancelar

Para:

De: Centro Integral de Monitoreo Olefina

Asunto:

FECHA	MATORIN	REGIONAL	SAN TOME
2009-08-01 - 2009-08-01	CORRECTIVO	CORRECTIVO	CORRECTIVO
2009-08-01 - 2009-08-02	0	0	0
2009-08-03 - 2009-08-03	0	0	0
2009-08-04 - 2009-08-04	0	0	0
2009-08-05 - 2009-08-05	0	0	0
2009-08-06 - 2009-08-06	0	0	0
2009-08-07 - 2009-08-07	0	0	0
2009-08-08 - 2009-08-08	0	0	0
2009-08-09 - 2009-08-09	0	0	0
2009-08-10 - 2009-08-10	0	0	0
2009-08-11 - 2009-08-11	0	0	0
2009-08-12 - 2009-08-12	0	0	0
2009-08-13 - 2009-08-13	0	0	0

FECHA	PUNTA DE MATA	PETROCURAGUA	PETROGUARICO
2009-08-01 - 2009-08-01	CORRECTIVO	CORRECTIVO	CORRECTIVO
2009-08-01 - 2009-08-02	0	0	0
2009-08-03 - 2009-08-03	0	0	0
2009-08-04 - 2009-08-04	0	0	0
2009-08-05 - 2009-08-05	0	0	0
2009-08-06 - 2009-08-06	0	0	0
2009-08-07 - 2009-08-07	0	0	0
2009-08-08 - 2009-08-08	0	0	0
2009-08-09 - 2009-08-09	0	0	0
2009-08-10 - 2009-08-10	0	0	0
2009-08-11 - 2009-08-11	0	0	0
2009-08-12 - 2009-08-12	0	0	0
2009-08-13 - 2009-08-13	0	0	0

FECHA	PETROLAGO	PETROPAR	PETROQUESE
2009-08-01 - 2009-08-01	CORRECTIVO	CORRECTIVO	CORRECTIVO
2009-08-01 - 2009-08-02	0	0	0
2009-08-03 - 2009-08-03	0	0	0
2009-08-04 - 2009-08-04	0	0	0
2009-08-05 - 2009-08-05	0	0	0
2009-08-06 - 2009-08-06	0	0	0
2009-08-07 - 2009-08-07	0	0	0
2009-08-08 - 2009-08-08	0	0	0
2009-08-09 - 2009-08-09	0	0	0
2009-08-10 - 2009-08-10	0	0	0
2009-08-11 - 2009-08-11	0	0	0
2009-08-12 - 2009-08-12	0	0	0
2009-08-13 - 2009-08-13	0	0	0

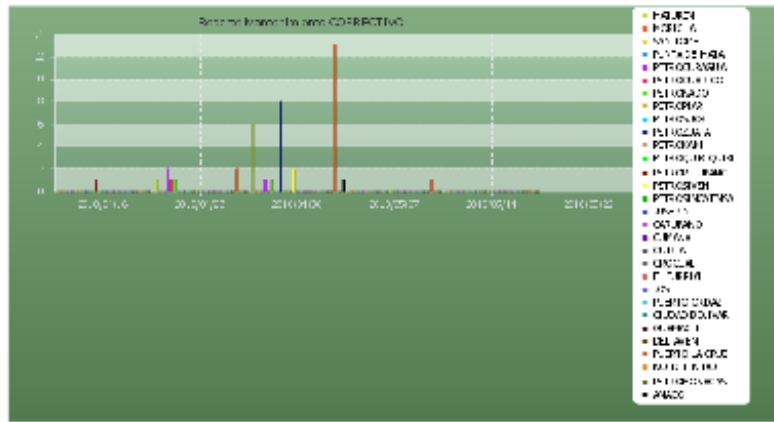
© PDVSA 2003 - Última actualización Abril de 2008

Pantalla Grafica: Reporte Por Mantenimiento

Al igual que el reporte estándar, este reporte permite generar los datos en formato PDF y Open Calc.



PARA: DOUGLAS ARAUJO
 DE: Centro Integral de Monitoreo Oriente
 ASUNTO: REPORTE POR MANTENIMIENTO



FECHA	MANTEN	SERVICIO	SANTIDAD
	CORRECTIVO	CORRECTIVO	CORRECTIVO
20100416-20100423	0	0	0
20100423-20100430	1	0	0
20100430-20100507	0	0	0
20100507-20100514	0	0	0
20100514-20100522	0	0	0

FECHA	PUNTA DE SANTA	PETROBRASIA	PETROBRASIA
	CORRECTIVO	CORRECTIVO	CORRECTIVO
20100416-20100423	0	0	0

A	B	C	D	E
FECHA	MATURIN	MORICHAL	SANTOME	PUNTA DE MATA
	CORRECTIVO	CORRECTIVO	CORRECTIVO	CORRECTIVO
2010-04-16 - 2010-04-23	0	0	0	0
2010-04-23 - 2010-04-30	1	0	0	0
2010-04-30 - 2010-05-07	6	0	0	0
2010-05-07 - 2010-05-14	0	0	0	0
2010-05-14 - 2010-05-22	0	0	0	0
FECHA	PETROURAGUA	PETROGLARICO	PETRONADO	PETROPAR
	CORRECTIVO	CORRECTIVO	CORRECTIVO	CORRECTIVO
2010-04-16 - 2010-04-23	0	0	0	0
2010-04-23 - 2010-04-30	2	1	1	0
2010-04-30 - 2010-05-07	1	0	1	0
2010-05-07 - 2010-05-14	0	0	0	0
2010-05-14 - 2010-05-22	0	0	0	0
FECHA	PETROSUCRE	PETROZUATA	PETROCAKI	PETROQUIRUIRE
	CORRECTIVO	CORRECTIVO	CORRECTIVO	CORRECTIVO
2010-04-16 - 2010-04-23	0	0	0	0
2010-04-23 - 2010-04-30	0	0	0	0
2010-04-30 - 2010-05-07	0	0	0	0
2010-05-07 - 2010-05-14	0	0	0	0
2010-05-14 - 2010-05-22	0	0	0	0

Pantalla OpenCalc: Reporte por Mantenimiento

Equipo

Permite generar estadísticas, y gráficas de todos los casos ya sean alarmas y/o eventos, por equipo. El usuario puede seleccionar entre una fecha inicio fin o una fecha específica.

Para generar un reporte por equipo el usuario debe seleccionar primeramente una categoría del campo localidad, luego se desplegaran todas las disciplinas, y de estas se seleccionara una categoría. De acuerdo a la localidad y disciplina seleccionada se desplegaran todos los equipos asociados a esas categorías y por último se selecciona el reporte que se desea generar. Entre los tipos de reportes por equipo están: impactos, mantenimientos, alarmas, eventos y alarmas-eventos.

La figura mostrada a continuación corresponde a la opción fecha inicio fecha fin.

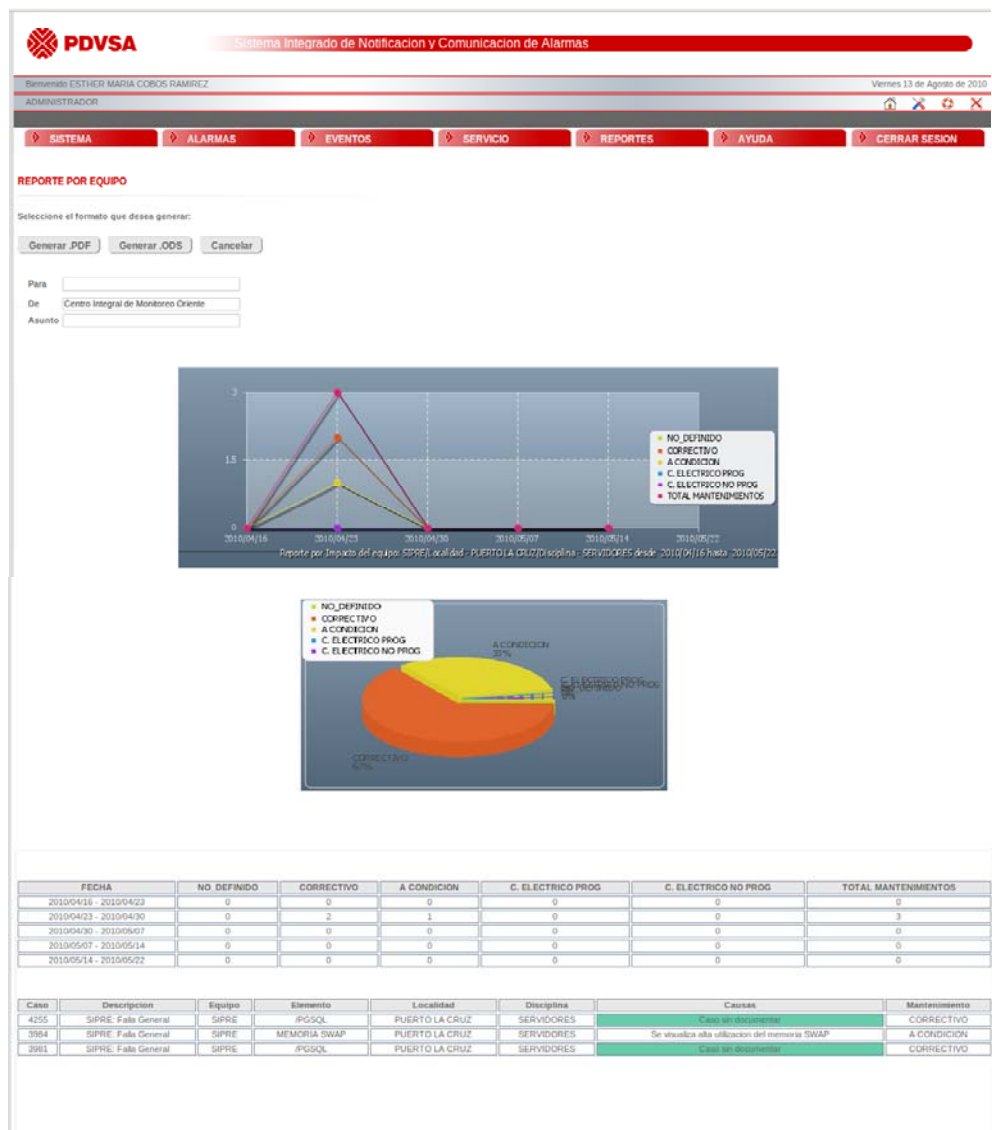
Pantalla Fecha Inicio Fecha Fin: Reporte Por Equipo

A diferencia de la opción fecha inicio fecha fin esta opción permite seleccionar diferentes rangos de fechas ya definidos por el sistema como: última semana, último mes, últimas 24 horas, estos rangos de fecha se toman desde el día que está corriendo actualmente hasta los días que ya pasaron dependiendo de la opción que se escoja por ejemplo si la opción es las últimas 24 horas y el día actual es el 24/05/2010 12:00:00, la fecha inicio seria 23/05/2010 12:00:00 y la fecha fin 24/05/2010 12:00:00.

La siguiente figura muestra la gráfica de todos los mantenimientos por equipo en este caso SIPRE, localidad puerto la cruz y disciplina servidores, desde el 07/04/2010 - 21/05/2010. La línea de color morado muestra el total de todos los mantenimientos por fecha. El eje de las X representa las fechas y el eje de las Y representa la cantidad de cada mantenimiento incluyendo el total.

La tabla mostrada en la siguiente figura indica el resultado de la gráfica, indicando la fecha por cada mantenimiento y el total. La tabla que se encuentra

debajo muestra todos los casos generados en esa fecha por el equipo SIPRE, incluyendo su número de caso, descripción, localidad, disciplina, causa de la falla y el tipo de mantenimiento. En el campo causa las celdas marcadas con color verde muestran que aun el caso no ha sido documentado.



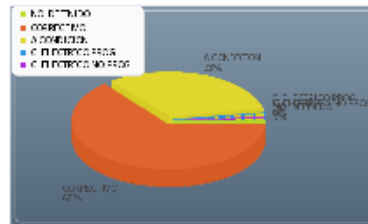
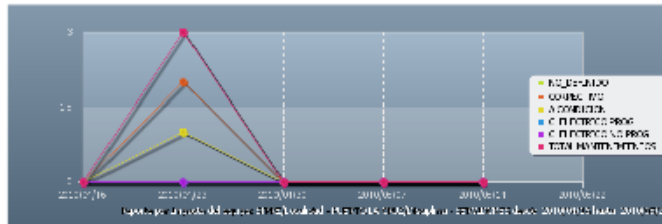
Pantalla Reporte Por Equipo

Al igual que el reporte estándar, este reporte permite generar los datos en formato PDF y hojas de cálculo.



COBOSE
FECHA: 2010-04-16 - 2010-05-22
HORA: 15:41:23 - 15:41:31

PARA: DOUGLAS ARAUJO
DE: Centro Integral de Monitoreo Oriente
ASUNTO: REPORTE EQUIPO POR MANTENIMIENTO



FECHA	NO. EQUIPO	CORRECTIVO	A CORRECCION	C. ELECTRICOS PERO	C. ELECTRICOS NO PERO	TOTAL
20100416-20100421	0	0	0	0	0	0
20100422-20100426	0	2	1	0	0	3
20100427-20100507	0	0	0	0	0	0
20100508-20100514	0	0	0	0	0	0
20100515-20100522	0	0	0	0	0	0

A	B	C	D	E	F	G	H	
1	FECHA	NO. DEFINIDO	COBRECTIVO	A CONDECION	C. ELECTRICO PROG.	C. ELECTRICO NO PROG.	TOTAL MANTENIMIENTOS	
2	2010/04/16 - 2010/04/23	0	0	0	0	0	0	
3	2010/04/23 - 2010/04/30	0	2	1	0	0	3	
4	2010/04/30 - 2010/05/07	0	0	0	0	0	0	
5	2010/05/07 - 2010/05/14	0	0	0	0	0	0	
6	2010/05/14 - 2010/05/22	0	0	0	0	0	0	
7								
8								
9								
10								
11								
12								
13	CASO	DESCRIPCION	EQUIPO	ELEMENTO	LOCALIDAD	DISCIPLINA	CAUSAS	MANTENIMIENTO
14	4255	SIPRE: Falla General	SIPRE	PGSQL	PUERTO LA CRUZ	SERVIDORES	Caso sin documentar	CORRECTIVO
15	3984	SIPRE: Falla General	SIPRE	MEMORIA SWAP	PUERTO LA CRUZ	SERVIDORES	Alta alta utilizacion del memo	A CONDECION
16	3981	SIPRE: Falla General	SIPRE	PGSQL	PUERTO LA CRUZ	SERVIDORES	Caso sin documentar	CORRECTIVO
17								
18								
19								
20								
21								
22								
23								
24								
25								
26								
27								
28								
29								
30								
31								
32								
33								

Pantalla OpenCalc: Reporte por Equipo

Búsqueda Rápida

Permite generar informes de todos los casos ya sean alarmas y eventos por IP o por nombre de equipo. El usuario puede seleccionar entre una fecha inicio fin o una fecha específica.

Para generar un reporte de búsqueda rápida primero se debe escoger que tipo de fecha se desea, luego se elije que tipo de búsqueda se realizara si es por IP o por equipo. Si es por IP, se debe ingresar una IP correcta estándar, no tiene que ser exacta. Si es por equipo, se debe ingresar bien sea el nombre del equipo completo, o una cadena.

PDVSA Sistema Integrado de Notificación y Comunicación de Alarmas

Bienvenido ESTHER MARIA COROS RAMIREZ Miércoles 22 de Septiembre de 2010

ADMINISTRADOR

[SISTEMA](#)
[ALARMAS](#)
[EVENTOS](#)
[SERVICIO](#)
[REPORTES](#)
[AYUDA](#)
[CERRAR SESION](#)

REPORTE DE BÚSQUEDA RÁPIDA

Seleccione el formato que desea generar:

Para:

De: Centro Integral de Montareo Oriente

Asunto:

# Casa	NAGROS	CMOR	Título	Equipo	Elemento	Fecha	
<input checked="" type="checkbox"/>	4849	OK	NUEVO	ER-PLCGUAS: Falla General	ER-PLCGUAS	ENLACE GUARAGUIMO - CE COLON PROYECTO ACFJ	2010-05-05 06:19:42
<input checked="" type="checkbox"/>	4767	OK	NUEVO	ER-PLCGUAS: Falla General	ER-PLCGUAS	ENLACE PLC - PETROGUARICO SERIALIUDG	2010-05-04 12:33:57
<input checked="" type="checkbox"/>	3973	OK	CERRADO	ER-PLCGUAS: Falla General	ER-PLCGUAS	ENLACE GUARAGUIMO - CE COLON PROYECTO ACFJ	2010-04-27 06:21:44

Pantalla Grafica: Reporte de Búsqueda Rápida

Hoja1 - Libro1 - OpenOffice.org Calc

Archivo Editar Ver Insertar Formato Herramientas Datos Ventana Ayuda

HTML tables fei Σ = EVENTO

A	B	C	D	E	F	G	H
EVENTO	ESTADO NAGROS	ESTADO CMOR	TITULO	EQUIPO	ELEMENTO	FECHA	TIPO EVENTO
2	4849	OK	NUEVO	ER-PLCGUAS: Falla General	ER-PLCGUAS	ENLACE GUARAGUIMO - CE COLON	AUTOMATICO
3	4767	OK	NUEVO	ER-PLCGUAS: Falla General	ER-PLCGUAS	ENLACE PLC - PETROGUARICO	AUTOMATICO
4	3973	OK	CERRADO	ER-PLCGUAS: Falla General	ER-PLCGUAS	ENLACE GUARAGUIMO - CE COLON	AUTOMATICO
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							
28							
29							
30							
31							
32							
33							

Hoja 1 / 1 Predeterminado STD Suma=0 100%

Pantalla OpenCalc: Reporte de Búsqueda Rápida.



COBOSE
FECHA: 2010-04-16 - 2010-05-22
HORA: 15:46:52 - 15:47:11

PARA: DOUGLAS ARAUJO
DE: Centro Integral de Monitoreo Oriente
ASUNTO: REPORTE DE BUSQUEDA RAPIDA

ER-PLCGUAS: Falla General
ER-PLCGUAS - Caída de ENLACE GUARAGUAO - CE COLON PROYECTO ACEJ

4849 - PUERTO LA CRUZ
FECHA/HORA INICIO: 2010-05-05 06:19:42
ESTADO C INGR: NUEVO

DESCRIPCION

Se anexa evidencia:
CRITICAL : Serial1/1/1:0:DOWN: 0 UP: 1 int not UP 2010-05-05 06:19:42

ER-PLCGUAS: Falla General
ER-PLCGUAS - Caída de ENLACE PLC - PETROGUARICO SERIAL1/1/0

4767 - PUERTO LA CRUZ
FECHA/HORA INICIO: 2010-05-04 12:33:57
ESTADO C INGR: NUEVO

DESCRIPCION

Se anexa evidencia:
CRITICAL : Serial1/1/0:0:DOWN: 0 UP: 1 int not UP 2010-05-04 12:33:57

ER-PLCGUAS: Falla General
ER-PLCGUAS - Caída de ENLACE GUARAGUAO - CE COLON PROYECTO ACEJ

3973 - PUERTO LA CRUZ
FECHA/HORA INICIO: 2010-04-27 06:21:44
ESTADO C INGR: CERRADO

DESCRIPCION

Se anexa evidencia:
CRITICAL : Serial1/1/1:0:DOWN: 0 UP: 1 int not UP 2010-04-27 06:21:44

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

TÍTULO	DESARROLLO DE UNA APLICACIÓN WEB BAJO SOFTWARE LIBRE PARA LA AUTOMATIZACIÓN DE LA GESTIÓN DE LAS ALARMAS GENERADAS POR UNA HERRAMIENTA DE MONITOREO DE EQUIPOS PARA LA SUPERINTENDENCIA DE MANTENIMIENTO DE LA PLATAFORMA DE LA GERENCIA DE AUTOMATIZACIÓN INFORMATICA Y TELECOMUNICACIONES DE PDVSA PUERTO LA CRUZ.
SUBTÍTULO	

AUTOR (ES):

APELLIDOS Y NOMBRES	CÓDIGO CULAC / E MAIL
Cobos R., Esther M.	CVLAC: V-16.927.294 E MAIL: cobosem360@hotmail.com
	CVLAC: E MAIL:
	CVLAC: E MAIL:
	CVLAC: E MAIL:

PALÁBRAS O FRASES CLAVES:

MAP, AIT, CIMOR, Equipos, Fallas, Alarmas, Predictivas, Correctivas, PHP, PostgreSQL, UML, WEBML, SoftwareLibre.

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

AREA	SUBAREA
Ingeniería y Ciencias Aplicadas	Ingeniería en Computación

RESUMEN (ABSTRACT):

La Superintendencia de Mantenimiento de la Plataforma (MAP), se encarga de establecer políticas y lineamientos de mantenimiento para la gerencia de Automatización, Informática y Telecomunicaciones (AIT), y poder lograr de ésta manera la prestación de un mejor servicio. El Centro Integral de Monitoreo Oriente (CIMOR) se encarga de garantizar la calidad y disponibilidad de la Plataforma de AIT, mediante el análisis y comprobación de los activos, esto se realiza a través de herramientas que se encargan de monitorear los equipos de servicios comunes oriente. Actualmente estas herramientas no permiten a los analistas realizar la respectiva documentación de las fallas reportadas, lo que impide generar acciones predictivas y correctivas de manera remota y asegurar la localización y notificación oportuna de eventos al personal de soporte de mantenimiento. Por ello la Superintendencia de Mantenimiento de la Plataforma se vio en la necesidad de solicitar a la gerencia de AIT el desarrollo de un software para la automatización de las alarmas generadas por una herramienta de monitoreo, para obtener beneficios de seguridad de la información, disminución de esfuerzo y tiempo, y rapidez en cuanto a la toma de decisiones. El diseño y desarrollo de la aplicación fue realizado según las fases del Proceso Unificado Racional (RUP). Entre las técnicas y tecnologías usadas para el diseño se encuentran el Lenguaje Unificado de Modelado (UML) y su extensión WEBML. Para la implementación se utilizaron las herramientas de PHP y PostgreSQL cumpliendo con las normas de la empresa, en cuanto al desarrollo de aplicaciones con software libre de acuerdo al decreto presidencial 3.390.

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:**CONTRIBUIDORES:**

APELLIDOS Y NOMBRES	ROL / CÓDIGO CVLAC / E_MAIL				
Veracierta T., Gabriela M.	ROL	CA	AS X	TU	JU
	CVLAC:	14.616.683			
	E_MAIL	gveracierta@hotmail.com			
	E_MAIL				
Pinto R., Carlimar D.	ROL	CA	AS	TU X	JU
	CVLAC:	V-12.576.006			
	E_MAIL	carlipinto@hotmail.com			
	E_MAIL				
Cortínez N., Claudio A.	ROL	CA	AS	TU	JU X
	CVLAC:	V-12.155.334			
	E_MAIL	clcortinez@cantv.net			
	E_MAIL				
Rodríguez M., Rhonald E.	ROL	CA	AS	TU	JU X
	CVLAC:	V-14.077.185			
	E_MAIL	rerodriguez@anz.udo.edu.ve			
	E_MAIL				

FECHA DE DISCUSIÓN Y APROBACIÓN:

2010	12	01
AÑO	MES	DÍA

LENGUAJE.SPA

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:**ARCHIVO (S):**

NOMBRE DE ARCHIVO	TIPO MIME
TesisAplicacionweb.doc	Application/msword

CARACTERES EN LOS NOMBRES DE LOS ARCHIVOS: A B C D E F G H
 I J K L M N O P Q R S T U V W X Y Z. a b c d e f g h i j k l m n o p q r s t u
 v w x y z. 0 1 2 3 4 5 6 7 8 9.

ALCANCE

ESPACIAL: _____ (OPCIONAL)

TEMPORAL: _____ (OPCIONAL)

TÍTULO O GRADO ASOCIADO CON EL TRABAJO:

___Ingeniero en computación_____

NIVEL ASOCIADO CON EL TRABAJO:

___Pre-Grado_____

ÁREA DE ESTUDIO:

___Departamento de Computación y Sistemas_____

INSTITUCIÓN:

___Universidad de Oriente - Núcleo Anzoátegui_____

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:**DERECHOS**

_____ De acuerdo con el artículo 41 del reglamento de trabajo de grado: _____
"los trabajos de grado son de exclusiva propiedad de la Universidad de Oriente y solo podran ser utilizados a otros fines con el consentimiento del consejo de núcleo respectivo, quien debera participarlos previamente al Consejo Universitario para su autorización"._____

Cobos R., Esther M.

AUTOR

AUTOR

AUTOR

Veracierta T., Gabriela M. Cortínez N., Claudio A. Rodríguez M., Rhonald E.

TUTOR

JURADO

JURADO

Bastardo, José L.

POR LA COMISION DE TESIS