

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



“ DESARROLLO DE UNA APLICACIÓN WEB BASADA EN TECNOLOGÍA
HELPDESK PARA OFRECER SERVICIOS DE SOPORTE TÉCNICO E
INVENTARIO EN LA GERENCIA DE INFORMÁTICA DE LA EMPRESA C.A.
HIDROLÓGICA DEL CENTRO, EN VALENCIA ESTADO CARABOBO ”

Realizado por:
José Miguel Suniaga Salazar
C.I.: 16.827.771

Trabajo de grado presentado como requisito parcial para optar al título de
INGENIERO EN COMPUTACIÓN

Barcelona, Octubre de 2009

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



“ DESARROLLO DE UNA APLICACIÓN WEB BASADA EN TECNOLOGÍA
HELPDESK PARA OFRECER SERVICIOS DE SOPORTE TÉCNICO E
INVENTARIO EN LA GERENCIA DE INFORMÁTICA DE LA EMPRESA C.A.
HIDROLÓGICA DEL CENTRO, EN VALENCIA ESTADO CARABOBO ”

Asesores:

Ing. Zulirais García
Asesor Académico

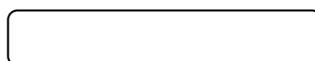
Ing. Eduardo Primera
Asesor Industrial

Barcelona, Octubre de 2009.

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



“ DESARROLLO DE UNA APLICACIÓN WEB BASADA EN TECNOLOGÍA
HELPDESK PARA OFRECER SERVICIOS DE SOPORTE TÉCNICO E
INVENTARIO EN LA GERENCIA DE INFORMÁTICA DE LA EMPRESA C.A.
HIDROLÓGICA DEL CENTRO, EN VALENCIA ESTADO CARABOBO ”



Jurado Calificador

Ing. Zulirais García
Asesor Académico

Ing. Rhonald Rodríguez
Jurado Principal

Ing. Gabriela Veracierta
Jurado Principal

Barcelona, Octubre de 2009

RESOLUCIÓN

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 sólo podrán ser utilizados a otros fines con el consentimiento del consejo de núcleo respectivo, quien lo participará al Consejo Universitario”.

DEDICATORIA

A mi padre, mi madre y mi hermana, por estar siempre conmigo en los momentos más difíciles, ustedes han sido mi fortaleza a lo largo de este recorrido.

AGRADECIMIENTOS

A mis padres, Jesús y Lumen, ejemplos de humildad y honradez dignos de admirar, sin su esfuerzo y sacrificio, no hubiese podido alcanzar esta meta.

A mi hermana Asunción, con quien siempre he podido contar cuando he necesitado, gracias por estar ahí manita, formas parte de este triunfo.

A mis tíos, Aquiles y Rebecca, por abrirme las puertas de su casa mi etapa de pasantía y por ayudarme hasta en el último momento de ésta.

A mis padrinos, Rómulo y Panchita, gracias por estar pendiente de mi, ambos han sido como unos segundos padres.

A mi familia, por su apoyo incondicional a lo largo de este recorrido.

A Daniel, Miky, Luis, Rosa y Migde, mis panas del alma, siempre compartiendo cada experiencia, cada alegría y cada tristeza.

A la gerencia de informática de HIDROCENTRO por confiar en mí y entregarme la responsabilidad de la elaboración de este proyecto

A mi tutora académica, la Prof. Zulirais García, por transmitirme siempre esas buenas vibras y por brindarme sus valiosos conocimientos.

Y a todas aquellas personas que de una u otra manera hicieron posible la realización de este proyecto, a todos mis más sinceros agradecimientos.

ÍNDICE GENERAL

RESOLUCIÓN	IV
DEDICATORIA	V
AGRADECIMIENTOS	VI
ÍNDICE GENERAL	VII
ÍNDICE DE FIGURAS	XII
ÍNDICE DE TABLAS	XVI
RESUMEN	XIX
CAPÍTULO I : ASPECTOS GENERALES	20
1.1. Descripción de la Empresa	20
1.2. Planteamiento del Problema	20
1.3. Objetivos del Proyecto	23
1.3.1. Objetivo General	23
1.3.2. Objetivos Específicos	23
CAPÍTULO II : MARCO TEÓRICO	25
2.1. Antecedentes de la Investigación	25

2.2. Resumen de Conocimientos Previos	26
2.2.1. Tecnología de Servicios Helpdesk	26
2.2.1.1. Mecanismo de funcionamiento del Helpdesk	26
2.2.1.2. Beneficios del Helpdesk	26
2.2.2. Aplicación	27
2.2.3. Servidor	27
2.2.4. Servidor Web	27
2.2.5. Entorno Cliente-Servidor	28
2.2.6. Navegador o Browser	28
2.2.7. Aplicación Web	28
2.2.7.1. Ventaja del uso de Aplicaciones Web	28
2.2.7.2. Tecnologías para el desarrollo de aplicaciones Web	29
2.2.8. Lenguaje de Programación Java	30
2.2.8.1. Características básicas Java	30
2.2.8.2. Entorno de Desarrollo Java	31
2.2.9. NetBeans	31
2.2.10. Sistema Gestor de Bases de Datos PostgreSQL	32
2.2.10.1. Características del PostgreSQL	32
2.2.10.2. Entorno Gráfico PgAdmin III	33
2.2.11. Lenguaje Unificado de Modelado (UML)	33
2.2.12. Metodología del Proceso Unificado de Rational (RUP)	33
CAPÍTULO III : FASE DE INICIO	37
3.1. Alcance del Proyecto	38
3.2. Modelo de Dominio	39
3.2.1. Identificación de las Clases Conceptuales	39
3.2.2. Asociaciones de las Clases Conceptuales	41
3.2.3. Diagrama del Modelo de Dominio	42
3.2.4. Glosario de Términos	43
3.3. Riesgos del Sistema	46

3.3.1. Falta de información inicial	47
3.3.2. Falta de experiencia con las herramientas utilizadas	47
3.3.3. Cambios en los requisitos	48
3.3.4. Diseño erróneo	49
3.3.5. Pérdida de datos	50
3.3.6. Falla de operatividad	50
3.3.7. Hardware y/o Software inadecuados	50
3.4. Requerimientos del Sistema	51
3.4.1. Requisitos funcionales	51
3.4.2. Requisitos no funcionales	52
3.5. Modelo de Casos de Uso	53
3.5.1. Identificación de Actores	53
3.5.2. Identificación de los Casos de Uso	55
3.5.3. Descripción de los Casos de Uso	56
3.5.4. Diagrama de Casos de Uso	66
3.5. Análisis	68
3.5.1. Análisis de los Casos de Uso	68
3.5.1.1. Identificación de las Clases de Análisis	68
3.5.1.2. Diagramas de las Clases de Análisis	75
3.5.1.3. Diagramas de las Colaboración	77
3.5.2. Análisis de la Arquitectura	81
3.5.2.1. Identificación de los Paquetes de Análisis	81
3.6. Evaluación de la Fase de Inicio	85
CAPÍTULO IV : FASE DE ELABORACIÓN	86
4.1. Requerimientos del Sistema	87
4.1.1. Requisitos adicionales	87
4.1.2. Interfaz de Inicio de SAIME	87
4.1.3. Interfaz de Usuario	88

4.2. Modelo de Casos de Uso Actualizado	90
4.2.1. Identificación de actor adicional	90
4.2.2. Identificación del Caso de Uso adicional	90
4.2.3. Descripción del Caso de Uso adicional	91
4.2.4. Diagrama de Casos de Uso Actualizado	93
4.3. Análisis	93
4.3.1. Identificación de las Clases de Análisis	95
4.3.2. Diagrama de las Clases de Análisis	95
4.3.3. Diagrama de Colaboración	96
4.4. Diseño	97
4.4.1. Diagrama de Clase de Diseño	97
4.4.2. Diagramas de Secuencia	101
4.4.3. Diagrama de Capas	102
4.4.4. Diagrama de Base de Datos	104
4.5. Implementación	105
4.5.1 Diagrama de Componentes	107
4.5.2. Implementación de los componentes asociados a la solicitud de un nuevo ticket	108
4.6. Evaluación de la Fase de Elaboración	114
 CAPÍTULO V : FASE DE CONSTRUCCIÓN	 115
5.1 Implementación	115
5.1.1. Escogencia del Lenguaje de Programación	116
5.1.2. Escogencia del Gestor de Base de Datos	117
5.1.3. Diagrama de Componentes Total	118
5.2. Pruebas	119
5.2.1. Pruebas por Unidad	120
5.2.2. Pruebas de Integración	121

5.2.2.1. Pruebas de Integración de gestionar Departamento	123
5.3. Evaluación de la Fase de Construcción	128
CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES	129
6.1. Conclusiones	129
6.2. Recomendaciones	130
BIBLIOGRAFÍA	132

ÍNDICE DE FIGURAS

Figura 1.1. Estructura Organizacional de la C.A. HIDROCENTRO.	21
Figura 2.1. Estructura de RUP.	36
Figura 3.1. Esfuerzo en flujos de trabajo de la fase de inicio.	37
Figura 3.2. Clases Conceptuales.	41
Figura 3.3. Diagrama del Modelo de Dominio	44
Figura 3.4. Diagrama de Casos de Uso General de SAIME	67
Figura 3.5. Prototipo de Clase de Interfaz	69
Figura 3.6. Prototipo de Clase de Control	71
Figura 3.7. Prototipo de Clase de Entidad	73
Figura 3.8. Diagrama de Clase de Análisis para el Caso de Uso Gestionar Inventario	75
Figura 3.9. Diagrama de Clase de Análisis para los Casos de Uso Asignar Encargado de Ticket y Consultar Asignaciones de Tickets	76
Figura 3.10. Diagrama de Clase de Análisis para los Casos de Uso Procesar Ticket, Consultar Ticket y Emitir Comentario de Respuesta de Solicitud	77
Figura 3.11. Diagrama de Colaboración para el Caso de Uso Gestionar Inventario	78
Figura 3.12. Diagrama de Colaboración para los Casos de Uso Asignar Encargado de Ticket y Consultar Asignaciones de Tickets.	79

Figura 3.13. Diagrama de Colaboración para los Casos de Uso Procesar Nuevo Ticket, Consultar Ticket y Emitir Comentario Ticket de Respuesta de Solicitud	80
Figura 3.14. Paquetes de Análisis: Ticket	81
Figura 3.15. Paquetes de Análisis: Departamentos.	82
Figura 3.16. Paquetes de Análisis: Usuarios.	82
Figura 3.18. Paquetes de Análisis: Consultas DB.	83
Figura 3.19. Paquetes de Análisis: Inventario.	83
Figura 3.20. Paquetes de Análisis: Reportes Estadísticos.	84
Figura 3.21. Diagrama de Paquetes de Análisis del Sistema SAIME.	84
Figura 4.1. Esfuerzo en flujos de trabajo de la fase de elaboración.	87
Figura 4.2. Interfaz de Inicio (autenticación de usuario)	88
Figura 4.3. Prototipo de Interfaz de Usuario	89
Figura 4.4. Diagrama de Casos de Uso General de SAIME	94
Figura 4.5. Diagrama de Clases de Análisis para el Caso de Uso Consultar Estadísticas	96
Figura 4.6. Diagrama de Colaboración para el Caso de Uso Consultar Estadísticas (actor Administrador de Sistema)	96
Figura 4.7. Diagrama de Colaboración para el Caso de Uso Consultar Estadísticas (actor Supervisor)	97
Figura 4.8. Diagrama de Clases de Diseño del SAIME	100

Figura 4.9. Diagrama de Clases de Secuencia del Caso de Uso Procesar Nuevo Ticket	101
Figura 4.10. Diagrama de Clases de Secuencia del Caso de Uso Gestionar Inventario	103
Figura 4.11. Diagrama de Capas	104
Figura 4.12. Modelo Entidad – Relacion del SAIME [Fuente: Elaboración Propia]	106
Figura 4.13. Diagrama parcial de Componentes (CU Procesar Nuevo Ticket)	107
Figura 4.14. Interfaz de inicio del Sistema (autenticación de usuario)	108
Figura 4.15. Interfaz de Usuario Nuevo Ticket	110
Figura 4.16. Ventana de confirmación para el almacenamiento del Ticket	112
Figura 5.1. Esfuerzo en flujos de trabajo de la fase de construcción.	115
Figura 5.2. Interfaz del entorno de desarrollo JAVA	117
Figura 5.3. Interfaz del entorno grafico de PostgreSQL	118
Figura 5.4. Diagrama de Componentes Total	119
Figura 5.5. Diagrama de Componentes Total por fase de integración de SAIME	122
Figura 5.6. Interfaz de Gestión de Departamentos	124
Figura 5.7. Introducción de datos en la ventana de Departamentos	124
Figura 5.8. Consulta de datos en la ventana de Departamentos	125

ÍNDICE DE TABLAS

Tabla 3.1. Lista de Categoría de Conceptos. (1/2)	39
Tabla 3.1. Lista de Categoría de Conceptos. (2/2)	40
Figura 3.2. Clases Conceptuales.	41
Tabla 3.2. Asociaciones de las Clases Conceptuales. (1/2)	41
Tabla 3.2. Asociaciones de las Clases Conceptuales. (2/2)	42
Tabla 3.3. Glosario de Términos. (1/3)	43
Tabla 3.3. Glosario de Términos. (2/3)	45
Tabla 3.3. Glosario de Términos. (3/3)	46
Tabla 3.4. Descripción de Actores. (1/2)	54
Tabla 3.4. Descripción de Actores. (2/2)	55
Tabla 3.5. Definición de Casos de Uso. (1/2)	55
Tabla 3.5. Definición de Casos de Uso. (2/2)	56
Tabla 3.6. Gestionar Inventario. (1/2)	57
Tabla 3.6. Gestionar Inventario. (2/2)	58
Tabla 3.7. Gestionar Departamento. (1/2)	58
Tabla 3.7. Gestionar Departamento. (2/2)	59
Tabla 3.8. Procesar Usuarios. (1/3)	59

Tabla 3.8. Procesar Usuarios. (2/3)	60
Tabla 3.8. Procesar Usuarios. (3/3)	61
Tabla 3.9. Consultar Asignaciones de Tickets. (1/2)	61
Tabla 3.9. Consultar Asignaciones de Tickets. (2/2)	62
Tabla 3.10. Asignar Encargado de Tickets. (1/2)	62
Tabla 3.10. Asignar Encargado de Tickets. (2/2)	63
Tabla 3.11. Emitir Comentarios de Respuesta de Solicitud. (1/2)	63
Tabla 3.11. Emitir Comentarios de Respuesta de Solicitud. (2/2)	64
Tabla 3.12. Consultar Tickets.	64
Tabla 3.13. Procesar Nuevo Ticket.	65
Tabla 3.14. Clases de Interfaz (1/3)	69
Tabla 3.14. Clases de Interfaz (2/3)	70
Tabla 3.14. Clases de Interfaz (3/3)	71
Tabla 3.15. Clases de Control (1/2)	72
Tabla 3.15. Clases de Control (2/2)	73
Tabla 3.16. Clase de Entidad.	74
Tabla 4.1. Descripción de actor adicional.	90
Tabla 4.2. Descripción de los Casos de Uso adicionales	91
Tabla 4.3. Procesar Consultas Estadísticas. (1/2)	91
Tabla 4.3. Procesar Consultas Estadísticas. (2/2)	92

Tabla 4.4. Consultar Estadísticas. (1/2)	92
Tabla 4.4. Consultar Estadísticas. (2/2)	93
Tabla 4.5. Identificación de las Clases de Análisis para el Caso de Uso Consultar Estadísticas.	95
Tabla 5.1. Clase de equivalencia para el componente departamentos.java	120
Tabla 5.2. Casos de prueba de caja negra para el componente departamentos.java	121
Tabla 5.3. Datos de entrada para probar la integración de Configuración de Señales	123

RESUMEN

La Gerencia de Informática de la empresa C.A. Hidrológica del Centro, HIDROCENTRO tiene como objetivo ofrecer a los empleados de la empresa servicios de calidad en el área de información; desarrollando sistemas óptimos y dando soporte a los recursos informáticos con eficiencia, en el menor tiempo posible. En tal sentido, la Gerencia de Informática debe: mantener un control del inventario de los recursos informáticos de la empresa así como de los servicios que se prestan a estos, de manera que se pueda gestionar los equipos que estén activos y vigilar el desempeño de los servicio de soporte técnico a cargo de los empleados de la gerencia. Actualmente, la gerencia lleva este control de manera manual, es por esa razón que se requiere de un proyecto que permita la automatización de dichos procesos. En este informe de trabajo de grado se presenta el diseño de un Sistema de Administración de Inventario y Mantenimiento de Equipos (SAIME) que apoyará los procesos descritos anteriormente, el cual está modelado y documentado bajo el Lenguaje Unificado de Modelado (UML), siguiendo la metodología RUP, implantado bajo la plataforma Windows, programado con el lenguaje de programación Java y cuyos datos son almacenados en una base de datos PostgreSQL.

CAPÍTULO I

ASPECTOS GENERALES

1.1. Descripción de la Empresa

La empresa C.A. Hidrológica del Centro, HIDROCENTRO es un ente gubernamental venezolano que tiene por objeto la administración, operación, mantenimiento ampliación y reconstrucción de los sistemas de distribución de agua potable y de los sistemas de recolección, tratamiento y disposición de aguas residuales en los estados Aragua, Carabobo y Cojedes.

1.2. Planteamiento del Problema

C.A. HIDROCENTRO, está compuesta por un conjunto de gerencias supeditadas entre dos vicepresidencias (ver Figura 1.1) como son la vicepresidencia de negocios y la vicepresidencia de servicios; ambas vicepresidencias son dependencias de la presidencia.

Las vicepresidencias son las encargadas de supervisar la correcta gestión de las gerencias, en el caso de la Gerencia de Informática subordinada por la vicepresidencia de servicios su delegación es la de coordinar todos lo referente a los sistemas que automatizan los activos de la empresa.

La Gerencia de Informática de HIDROCENTRO se encarga de proporcionar servicios en la administración de las tecnologías de información, utilizadas en los diferentes departamentos de la organización. También se encarga de proponer y gestionar desarrollos, normas y procedimientos, con la finalidad de contribuir a las mejoras continuas. Su labor principal es velar por la disponibilidad y funcionalidad de los recursos informáticos como por ejemplo: computadores, impresoras, escaners, software, entre otros.

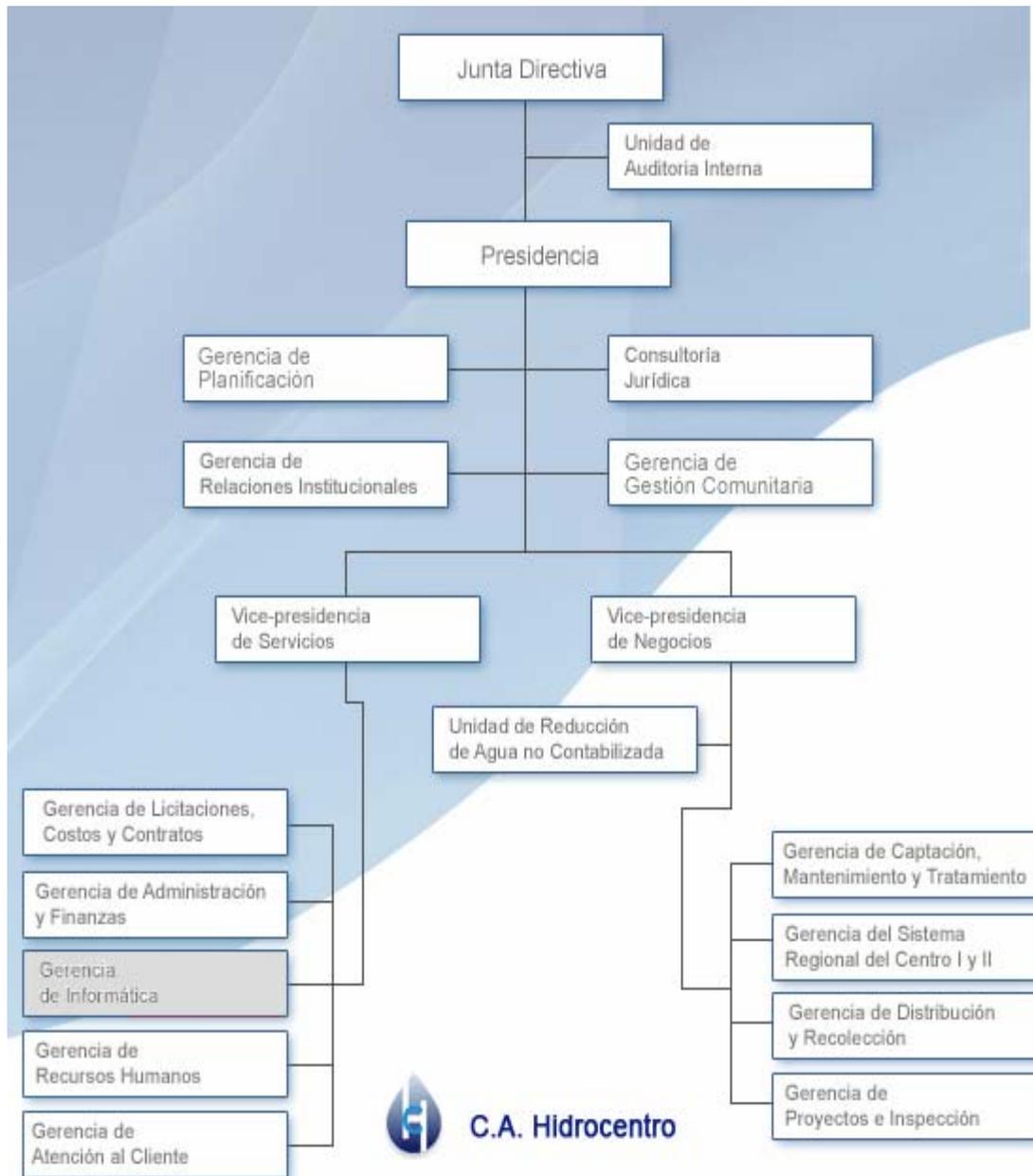


Figura 1.1. Estructura Organizacional de la C.A. HIDROCENTRO.

[Fuente: <http://www.hidrocentro.gob.ve/hc/estructuraOrganizativa/>]

La política de la Gerencia de Informática, está enfocada en la calidad de sus servicios ofrecidos a las gerencias, departamentos y agencias de la empresa, para

asegurar el rendimiento óptimo de sus recursos informáticos y así la satisfacción de sus empleados.

En la actualidad, existen procedimientos manuales a través del uso de formato o planillas que permiten ejercer la gestión de las actividades de soporte y mantenimiento, hasta el momento los requerimientos o solicitudes son realizadas bajo llamadas telefónicas o el uso de correo electrónico, estos mecanismos dificultan la aplicación de un seguimiento, control y certificación adecuada por lo que incide directamente sobre el estatus de la plataforma tecnológica (Hardware y Software). El control manual presenta debilidades y deficiencias que afectan las medidas o variables de control dentro de la gestión de la Gerencia de Informática y sus departamentos, por lo que se plantea la necesidad de implementar una solución de automatización que permita brindar los correctivos necesarios y brinde la posibilidad de tener un solución de control, gestión y seguimientos de las actividades de soporte e inventario de la plataforma tecnológica.

En este sentido, se propone el diseño de una aplicación web basada en la filosofía helpdesk que permita automatizar la información de los recursos informáticos para solventar las debilidades y deficiencias que presentan los mecanismos actuales. Las principales funcionalidades del programa estarán articuladas sobre los siguientes ejes:

- El inventario preciso de todos los recursos informáticos, y el software existente, cuyas características se almacenan en bases de datos.
- Gestión de las diferentes labores de mantenimiento llevadas a cabo sobre esos recursos informáticos.
- Flujo de trabajo incorporado para asignar requerimientos a los técnicos del departamento de soporte técnico de forma automática.
- Manejo estadístico y control de tiempos de atención de las actividades de soporte y sus requerimientos.
- Certificación de actividades por parte de los usuarios y cierre de casos abiertos.

- Consolas de administración, seguimiento y control para los supervisores.
- Tablas de información, que incluye las especificaciones de los incidentes y a quienes están asignados.

La aplicación a desarrollar, permitirá la administración completa de incidentes y capacidad de resolución inmediata. Asimismo, podrá ser utilizado como consulta de inventario y de labores de mantenimiento de recursos informáticos por el personal administrador. De igual manera, facilitará a las jefaturas de sistemas y soporte, el control y la evaluación del desempeño del personal a su cargo, y a la vez permitirá a la Gerencia de Informática tomar las decisiones que conlleven a un manejo óptimo de las operaciones de mantenimiento.

Además de esto el uso de aplicación producirá ventajas inherentes tales como la reducción de costos, la optimización de los recursos y la rigurosa gestión de licencias.

1.3. Objetivos del Proyecto

1.3.1. Objetivo General

Desarrollar una aplicación Web basada en tecnología helpdesk para ofrecer servicios de soporte técnico e inventario en la gerencia de informática de la empresa C.A. Hidrológica del Centro, en Valencia Estado Carabobo.

1.3.2. Objetivos Específicos

- Realizar la recopilación de los requisitos que permita cumplir con los servicios que se ofrecen en cada una de las jefaturas dentro de la gerencia.
- Desarrollar un módulo de solicitud de tickets de mantenimiento de recursos informáticos.
- Desarrollar un módulo que permita monitorear los servicios atendidos por el personal de la gerencia.
- Desarrollar un módulo de inventario de recursos informáticos.
- Desarrollar un módulo de reportes estadísticos de los datos recolectados por el sistema.

- Integración de los módulos para depurar y corregir las fallas que puedan ser encontradas en el proceso.

CAPÍTULO II

MARCO TEÓRICO

2.1. Antecedentes de la Investigación

Cabrera (2002) llevo a cabo un estudio, titulado: “Desarrollo de un software educativo e interactivo como apoyo en el proceso enseñanza – aprendizaje de la asignatura matemáticas de séptimo grado de Educación Básica, utilizando tecnología World Wide Web”. Con este estudio el autor pretende conseguir una solución educativa con ayuda del computador que supere las limitaciones de los entornos educativos convencionales. Al final concluye, que para la realización de un software robusto, flexible y escalable, es necesario la utilización de una buena metodología; las metodologías basadas en UML ofrecen un modo estándar de visualizar, especificar, construir, documentar y comunicar los artefactos del sistema. [1]

Fermín (2005) desarrollo un proyecto de grado, titulado: “Diseño de un Sistema de Información para el Monitoreo de la Red LAN y de Apoyo al Departamento de Redes y Comunicaciones de una Empresa Siderúrgica”. Mediante este diseño el autor plantea un sistema capaz de obtener y almacenar la información histórica del rendimiento de los dispositivos de red. [2]

Velasco y Gotilla (2006) realizaron una investigación con el propósito de diseñar un sistema de información bajo licencia de software libre para el manejo de las actividades administrativas de los módulos compra y venta en una pequeña y/o mediana empresa. El autor concluyó que: el software libre es un software que una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. Asimismo expone que la licencia GNU GPL posibilita la modificación, redistribución del software, pero únicamente bajo esa misma licencia. [3]

Obando y Sales (2008) desarrollaron un software con el propósito de minimizar los procedimientos manuales en el Banco de Sangre del Hospital Universitario Dr.

Luis Razetti de Barcelona, Estado Anzoátegui. Este trabajo se fundamentó en la visión de aumentar la productividad del personal del hospital para así brindar un servicio eficaz y eficiente para la atención de usuario donante. Al concluir, los autores expresan que la aplicación permitirá reducir los tiempos de generación de reportes y adicionalmente es una fuente de información para la consulta y toma de decisiones referentes al control y seguimiento de los hemodadores. [4]

2.2. Resumen de Conocimientos Previos

2.2.1. Tecnología de Servicios Helpdesk

En términos generales, un helpdesk es un recurso de información y asistencia que soluciona problemas con computadores o productos similares.

2.2.1.1. Mecanismo de funcionamiento del Helpdesk [5]

Como su nombre lo dice, es una “Mesa de Ayuda”, tiene varias funciones:

Proporciona a los usuarios un punto central para recibir ayuda en diversos temas.

El servicio de ayuda normalmente gestiona sus solicitudes a través de un software helpdesk, así como un sistema de seguimiento de incidentes, que les permite rastrear las solicitudes de usuarios con un único número de solicitud llamado ticket.

Por medio del software el usuario notifica su problema, y este le emite un ticket que tiene los detalles del problema. Si el apoyo técnico es capaz de resolver el problema en cuestión, el ticket queda cerrado y con la documentación de esa solución permite que otros técnicos de asistencia a los usuarios para hacer referencia en el futuro.

2.2.1.2. Beneficios del Helpdesk [5]

La asistencia a los usuarios del software puede ser una herramienta muy beneficiosa cuando se utiliza para encontrar, analizar y eliminar los problemas comunes en una organización del entorno informático.

El Helpdesk ayuda a incrementar la productividad y aumenta la satisfacción de los usuarios internos y externos.

2.2.2. Aplicación [6]

En informática, una aplicación es un tipo de programa informático diseñado para facilitar al usuario la realización de un determinado tipo de trabajo. A diferencia de otros tipos de programas como los sistemas operativos, las utilidades, y los lenguajes de programación, que realizan tareas no pertinentes al usuario común.

2.2.3. Servidor [7]:

En informática, un servidor es un tipo de software que realiza ciertas tareas en nombre de los usuarios. El término servidor también se utiliza para referirse al equipo físico en el cual funciona ese software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos.

2.2.4. Servidor Web [7]:

En la Web, un servidor Web es un computador que usa el protocolo http para enviar páginas Web al computador de un usuario cuando el usuario las solicita. Los servidores Web, servidores de correo y servidores de bases de datos son a lo que tiene acceso la mayoría de la gente al usar Internet.

Alternativamente, el servidor Web podría referirse al software, como el servidor de http de Apache, que funciona en la máquina y maneja la entrega de los componentes de los páginas Web como respuesta a peticiones de los navegadores de los clientes.

2.2.5. Entorno Cliente-Servidor:

La arquitectura cliente-servidor se creó para manejar los nuevos entornos de computo en lo que un gran número de computadores personales, estaciones de trabajo, servidores de archivos, impresoras y otros equipos a través de una red. [8]

La idea primaria de un entorno cliente-servidor es que debe haber un sitio donde se centraliza la información, que se desea distribuir bajo demanda a un conjunto de personas o máquinas. La clave de este concepto radica en que si se produce un cambio en la información del sistema central, inmediatamente es propagada a los receptores de la información, a la parte cliente.

2.2.6. Navegador o Browser [9]:

Es una aplicación informática que permite el usuario visualizar documentos de hipertexto. Dentro de sus funciones están la petición de páginas Web, la representación correcta de sus contenidos y la gestión de los posibles errores que se puedan producir.

2.2.7. Aplicación Web [10]:

Una aplicación Web es cualquier aplicación al que un usuario puede utilizar accediendo a un servidor Web a través de una red ya sea Internet o intranet mediante un navegador.

2.2.7.1. Ventaja del uso de Aplicaciones Web:

El uso de aplicaciones Web genera una cierta cantidad de beneficios tanto para los analistas, desarrolladores y administradores que operan en la parte del servidor como para los usuarios que la manipulan en la parte del cliente. La facilidad de comunicación que proporciona Internet combinada con la necesidad de acceso remoto a aplicaciones sin necesidad de instalaciones en la máquina del usuario hace evolucionar este concepto.

La comunicación ya no se basa simplemente en la carga de una página estática, sino que esta puede ser el resultado de la ejecución en el servidor de alguna lógica de programación, es decir, interacción dinámica entre usuario y servidor. [10]

La facilidad de una administración centralizada las hace ideales tanto para su despliegue en Internet como en intranets corporativas.

Otro hecho a tener en cuenta es que una vez realizada una aplicación Web para uso interno de una empresa, por ejemplo en una intranet, el poner esa funcionalidad, o incluso funcionalidades nuevas, a disposición de empleados o el público general tiene un coste mínimo a la vez que una potencial proyección mundial.[11]

2.2.7.2. Tecnologías para el desarrollo de aplicaciones Web [10]:

Para el desarrollo de aplicaciones Web se han generado múltiples tecnologías entre las cuales se encuentran:

- CGI: Common Gateway Interface, fue la primera técnica utilizada para que el contenido de las páginas Web se generara de manera dinámica. En resumen el CGI es un mecanismo de comunicación entre servidor Web y una aplicación externa. La mayoría de estas aplicaciones CGIs se encuentran desarrolladas en PERL.

- Fast-CGI: es una solución similar al CGI solo que propone la creación de un solo proceso persistente por cada programa en lugar de por cada solicitud del cliente.

- Páginas dinámicas en servidor: Este enfoque consiste en insertar pequeños fragmentos de lógica de programación en el código HTML de la página. En este sentido se conocen alternativas, como PHP, ASP, JSP entre otros.

- Java: Es un lenguaje de programación construido a partir de lenguajes orientado a objetos como C++, pretendiendo ir mucho mas lejos de estos posee características como la recolección de basura, programación multihilos y el manejo de memoria a cargo del lenguaje.

- JDBC: Java DataBase Connectivity, Consiste en un conjunto de clases e interfaces escritas en Java que provee comunicación con bases de datos.

- Servlets: puede considerarse como una evolución de los CGI's como parte de la tecnología Java. Son programas Java que proveen la funcionalidad de generar dinámicamente contenidos Web.

- XSL: eXtensible Stylesheet Language, es una especificación desarrollada para aplicar formato a los documentos XML de forma estandarizada. La idea es asociar al documento XML con una hoja de estilo y a partir de esto visualizar el documento XML en cualquier plataforma.

- Applets de Java: Los applets java son pequeños programas que se descargan del servidor Web y se ejecutan en la Java Virtual Machine (JVM) del navegador

2.2.8. Lenguaje de Programación Java:

El lenguaje Java fue desarrollado en 1991 por un grupo de ingenieros de Sun Microsystems con el fin de desarrollar software para el control de pequeños dispositivos electrónicos (TV interactiva, microondas, tostadoras, etc.), lo que llevo a desarrollar un lenguaje sencillo capaz de generar código de tamaño reducido. Debido a la existencia de diferentes tipos de electrodoméstico surgió la necesidad de desarrollar un código neutro independiente del tipo de electrodoméstico. De esta idea nació el JVM o Java Virtual Machine la cual toma ese código neutro y lo convierte en código particular de la plataforma en que se ejecuta. [11]

El concepto de la máquina virtual llegó a las computadores personales para dar inicio a una nueva era en el desarrollo de aplicaciones capaces de ejecutarse en diferentes sistemas operativos.

En la actualidad, Java se utiliza para desarrollar aplicaciones empresariales a gran escala, para mejorar la funcionalidad de servidores Web, para proporcionar aplicaciones para los dispositivos domésticos y para muchos otros propósitos. [12]

2.2.8.1. Características básicas Java [11]:

Entre todas las características de Java, sus 3 más básicas son:

- Java es pequeño: Los programas Java son rápidos de descargar desde una página Web.

- Java es seguro: Prohíbe cualquier tipo de operación que pueda afectar la integridad del ambiente en que se ejecutan sus programas.

- Java es portable: Permite ser ejecutado en Windows, MacIntosh y otras plataformas sin modificación alguna.

2.2.8.2. Entorno de Desarrollo Java [11]:

Para desarrollar código Java se requiere algún paquete de programación Java. La compañía Sun Microsystems, creadora de Java distribuye gratuitamente el Java Development Kit (JDK), la cual se trata de un conjunto de programas y librerías que permiten desarrollar, compilar y ejecutar programas en Java.

Otra alternativa son los IDE (Integrated Development Environment), por sus siglas en ingles, son entornos de desarrollo integrado, que ofrecen un ambiente grafico en los que se tiene acceso a las herramientas del JDK por lo que es posible escribir, compilar y ejecutar código Java.

Ejemplo de algunos IDEs son:

- NetBeans Open-Source
- Eclipse Open-Source
- Forte de Sun
- JBuilder de Borland
- JCreator de Xinox
- JDeveloper de Oracle

2.2.9. NetBeans[13]:

NetBeans es un IDE (Entorno de Desarrollo Integrado) para el lenguaje de programación Java principalmente, esto significa que precisa que tenga instalado Java en su equipo de trabajo, por lo que hay que tener instalado en el equipo de desarrollo

el JDK que incluye la máquina Virtual. Las últimas versiones de NetBeans IDE cuentan también con soporte para otros lenguajes como C++, PHP y Ruby.

2.2.10. Sistema Gestor de Bases de Datos PostgreSQL[14]:

PostgreSQL es un gestor de bases de datos orientadas a objetos muy conocido y usado en entornos de software libre porque cumple los estándares SQL92 y SQL99, y también por el conjunto de funcionalidades avanzadas que soporta, lo que lo sitúa al mismo o a un mejor nivel que muchos Sistemas Gestores de Bases de Datos (SGBD) comerciales.

2.2.10.1. Características del PostgreSQL [14]:

- La API de acceso al SGBD se encuentra disponible en C, C++, Java, PERL, PHP, Python y TCL, entre otros.
- Cuenta con un rico conjunto de tipos de datos, permitiendo además su extensión mediante tipos y operadores definidos y programados por el usuario.
- Su administración se basa en usuarios y privilegios.
- Sus opciones de conectividad abarcan TCP/IP, sockets Unix y sockets NT, además de soportar completamente ODBC.
- Los mensajes de error pueden estar en español y hacer ordenaciones correctas con palabras acentuadas o con la letra 'ñ'.
- Es altamente confiable en cuanto a estabilidad se refiere.
- Puede extenderse con librerías externas para soportar encriptación, búsquedas por similitud fonética (soundex), etc.
- Control de concurrencia multi-versión, que mejora sensiblemente las operaciones de bloqueo y transacciones en sistemas multi-usuario.
- Soporte para vistas, claves foráneas, integridad referencial, disparadores, procedimientos almacenados, subconsultas y casi todos los tipos y operadores soportados en SQL92 y SQL99.

- Implementación de algunas extensiones de orientación a objetos. En PostgreSQL es posible definir un nuevo tipo de tabla a partir de otra previamente definida.

2.2.10.2. Entorno Gráfico PgAdmin III [14]:

Es el máximo exponente de cliente gráfico de PostgreSQL. En pgAdmin3 se puede ver y trabajar con casi todos los objetos de la base de datos, examinar sus propiedades y realizar tareas administrativas. También incorpora funcionalidades para realizar consultas, examinar su ejecución (como el comando explain) y trabajar con los datos.

2.2.11. Lenguaje Unificado de Modelado (UML):

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. [15]

Cabe destacar, que el UML es un lenguaje para especificar y no un método o un proceso, se utiliza para definir un sistema de software, para detallar los artefactos en el sistema para documentar y construir.

2.2.12. Metodología del Proceso Unificado de Rational (RUP):

Según Krutchen [16], "RUP es un proceso de ingeniería de software que provee un enfoque disciplinado para la asignación de tareas y responsabilidades dentro de una organización desarrolladora de software"

Explícitamente, RUP es una guía de cómo usar UML de la forma más efectiva, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

La metodología RUP cumple un ciclo de vida que consta de cuatro fases [18]:

- Iniciación o inicio, El Objetivo en esta etapa es determinar la visión del proyecto.
- Elaboración, En esta etapa el objetivo es determinar la arquitectura óptima.
- Construcción, En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
- Transición, El objetivo es llegar a obtener el release del proyecto.

El ciclo de vida es llevado cabo bajo 2 de flujos de trabajo, como son: Los flujos de trabajo del proceso y los flujos de trabajo de soporte.

Los flujos de control de proceso, son los necesarios para la realización de un proyecto de software, aunque para proyectos no muy grandes se pueden omitir algunas.

Los flujos de control de apoyo, son los que como su nombre lo indica sirven de apoyo a las del proceso y especifican otras características en la realización de un proyecto de software.

Flujos de Control de Proceso [17]

- Modelado de Negocios: tiene como objetivos, comprender la estructura y la dinámica de la organización, comprender problemas actuales e identificar posibles mejoras, comprender los procesos de negocio.
- Requerimientos: tiene como objetivos establecer lo que el sistema debe hacer, definir los límites del sistema, y una interfaz de usuario, realizar una estimación del costo y tiempo de desarrollo.
- Análisis y Diseño: define la arquitectura del sistema y tiene como objetivos trasladar requisitos en especificaciones de implementación
- Implementación: tiene como objetivos disciplina tiene como objetivos implementar las clases de diseño como componentes, asignar los componentes a los nodos, probar los componentes individualmente, integrar los componentes en un sistema ejecutable.

- Prueba: tiene como objetivos verificar la integración de los componentes, que todo lo solicitado esta presente y que los defectos hayan sido resueltos antes de la distribución.

- Despliegue: tiene como objetivos asegurar que el producto está preparado para el cliente, proceder a su entrega y recepción por el cliente.

Flujos de Control de Apoyo [17]

- Configuración y manejo del Cambio: es necesario para evitar confusiones costosas asegurando que los resultados de los artefactos no entren en conflicto con algunos de los siguientes tipos de problemas: Actualización simultánea y Versiones múltiples.

- Administración del Proyecto: Su objetivo es equilibrar los objetivos competitivos, administrar el riesgo, y superar restricciones para entregar un producto que satisface las necesidades de ambos clientes con éxito y los usuarios.

- Entorno: se enfoca sobre las actividades necesarias para configurar el proceso que engloba el desarrollo de un proyecto y describe las actividades requeridas para el desarrollo de las pautas que apoyan un proyecto.

La Figura 2.1, refleja el esfuerzo de los flujos de control en cada una de las fases.

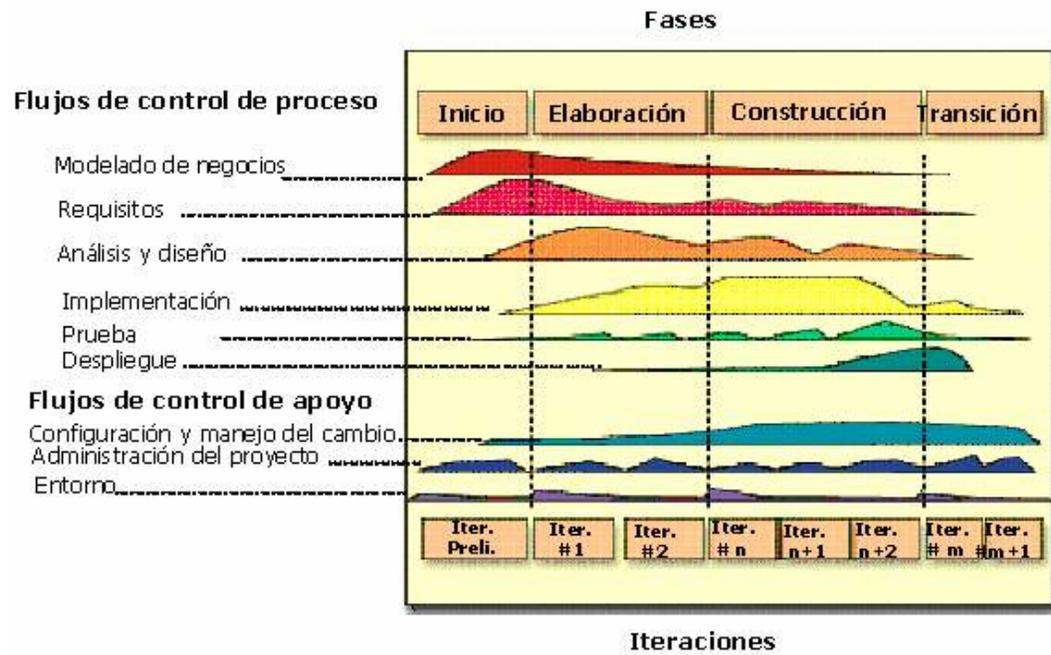


Figura 2.1. Estructura de RUP.

[Fuente: <http://rrivera334.blogspot.es/img/rup.jpg>]

CAPÍTULO III

FASE DE INICIO

En este capítulo correspondiente a la fase de inicio se origina la primera visión aproximada de la situación a resolver, mediante la identificación y descripción del dominio para el cual se desarrolla la aplicación Web. En esta fase se determinarán las necesidades de información y automatización de los procesos de negocio que tienen los usuarios de la aplicación Web en desarrollo. Se identificarán todos los requisitos del proyecto y se elaborarán los modelos iniciales que capturen la trayectoria y desenvolvimiento del mismo, considerando los recursos disponibles y partiendo de diagramas que bosquejan los requisitos, los cuales son la base para la futura definición de la arquitectura del software.

La fase de inicio tendrá como objetivos, establecer el alcance del proyecto, realizar el modelado del dominio del negocio, identificar la terminología clave del dominio, estimar los riesgos y las fuentes de incertidumbre, capturar los requisitos, modelar los casos de uso, mostrar al menos una arquitectura candidata y finalmente indicar los elementos básicos para el diseño de la interfaces de usuario.

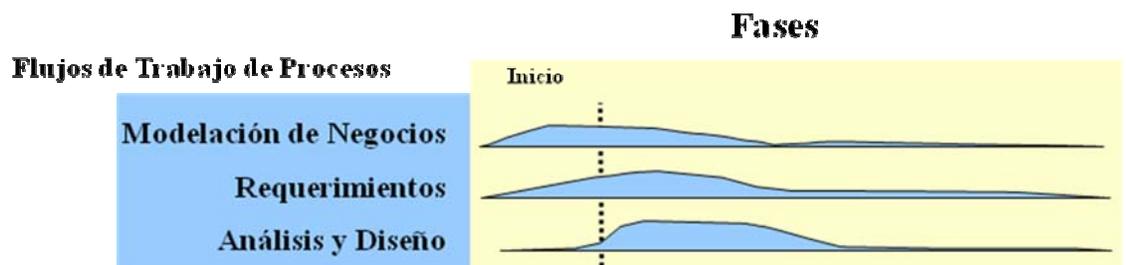


Figura 3.1. Esfuerzo en flujos de trabajo de la fase de inicio.

[Fuente: Elaboración Propia]

Para llevar a cabo la fase de inicio se abarcaran los flujos de trabajo, modelación de negocio, requerimientos y análisis (ver Figura 3.1), comenzando con una visión general del proyecto para poder realizar la captura de requisitos mediante el conocimiento del contexto del sistema, dicho contexto se logra a través del establecimiento del diagrama de domino del sistema, además se identificarán los riesgos que serán mitigados con la realización de las fases en el desarrollo del proyecto. A continuación se realizará la captura de requisitos y se identificarán los actores que interactúan con el sistema por medio del modelado de los diagramas de casos de uso. El siguiente flujo de trabajo es el de análisis, que permite modelar mediante los diagramas de clase de análisis para la comprensión de la información obtenida del flujo anterior. Por último, se realizarán los diagramas de colaboración para representar las interacciones entre los objetos, el diagrama de paquete de análisis para encapsular los casos de usos que fueron definidos al realizar el análisis del sistema.

3.1. Alcance del Proyecto

En este proyecto se propone desarrollar un Sistema de Administración de Inventario y Mantenimiento de Equipos (SAIME) para que sea la solución de automatización que permitirá a los empleados de la empresa reportar daños en sus equipos informáticos a través de una interfaz sencilla donde podrán interactuar directamente con el técnico encargado de resolver su caso, además que permitirá a los Jefes de Departamentos de la Gerencia de Informática supervisar el trabajo de sus empleados y a través de reportes estadísticos de las incidencias, también podrán realizar análisis que sirvan de apoyo en la toma de decisiones que conlleven a un manejo optimo de las operaciones de mantenimiento.

Una vez estudiado el alcance del proyecto, se procede a modelar todas sus actividades, para efectos del sistema SAIME se representará el contexto a través del Modelo de Dominio, el cual describe los conceptos más importantes del contexto,

como las clases conceptuales del dominio y enlaza unas con otras, con la finalidad de comprender los aspectos más importantes que se deben cubrir en este proyecto.

3.2. Modelo de Dominio

Un Modelo de Dominio es una representación de conceptos del Dominio de un problema. Los conceptos del Dominio representan “las cosas” que existen o los eventos que suceden para así comprender y describir sus elementos relevantes.

El diagrama del Modelo de Dominio se representa en UML con un Diagrama de Clases en lo que se muestra:

- Conceptos u objetos del dominio del problema: clases conceptuales.
- Asociaciones entre las clases conceptuales.
- Atributos de clases conceptuales.

3.2.1. Identificación de las Clases Conceptuales:

La creación de un Modelo de Dominio se comienza preparando una lista de conceptos idóneos a partir de la siguiente lista, la cual contiene muchas categorías comunes a cualquier dominio a representar. Los ejemplos sugeridos se tomaron del dominio de la tecnología helpdesk:

Tabla 3.1. Lista de Categoría de Conceptos. (1/2)

CATEGORÍAS	EJEMPLOS
Objetos físicos o tangibles	RecursoInformatico
Especificaciones, diseño o descripciones de cosas	DescripcionTicket, RespuestaMantenimiento, CierreTicket, StatusTicket
Lugares	Empresa, Gerencia, Departamento, Agencia

[Fuente: Elaboración Propia]

Tabla 3.1. Lista de Categoría de Conceptos. (2/2)

CATEGORÍAS	EJEMPLOS
Transacciones	TicketMantenimiento
Línea o renglón de elemento de transacciones	RecursoInformatico
Rol de las personas	Empleado, Técnico, Supervisor, Administrador
Contenedoras de otras cosas	Empresa, Departamento, Agencia, Inventario
Cosas dentro de un contenedor	Departamento RecursoInformatico
Otros sistemas de computo o electromecánicos externos al sistema	<<No Aplica>>
Conceptos de nombres abstractos	Conocimiento, Dedicación
Organizaciones	Departamento
Eventos	TicketMantenimiento
Procesos (a menudo no están representados como conceptos)	AsignacionTicket, RespuestaMantenimiento,
Reglas y políticas	PoliticodeEntradaySalidadeRecurso
Catálogos	CatalogodeProveedores
Registros de finanzas, de trabajo, de contratos de asuntos legales	HistorialdeTickets
Instrumentos y servicios financieros	ReportesEstadisticos
Manuales, libros	ManualdeUsuario, ManualdeReparaciones

[Fuente: Elaboración Propia]

A partir de la lista de categorías se genera la lista de conceptos adecuados para incluirlos al Diagrama de Modelo de Dominio del SAIME, representados gráficamente en la notación del diagrama de estructura estática de UML:

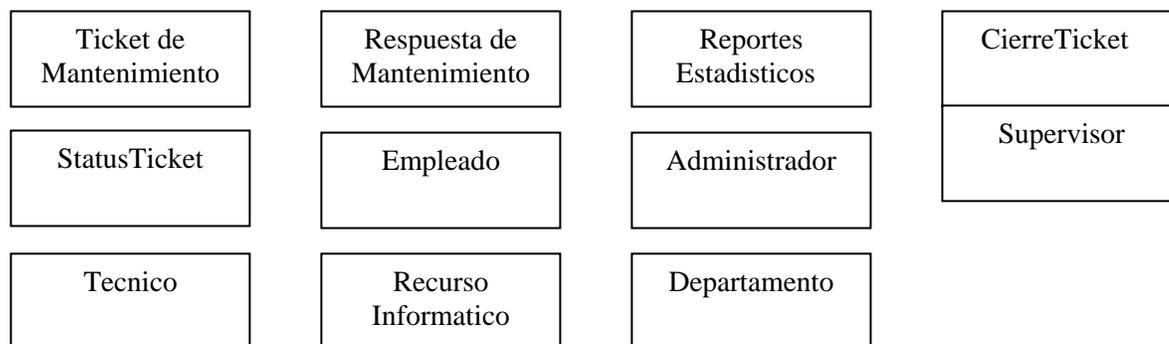


Figura 3.2. Clases Conceptuales.

[Fuente: Elaboración Propia]

3.2.2. Asociaciones de las Clases Conceptuales:

Una asociación se representa como una línea entre conceptos, identificado con el nombre de la asociación. Este nexo es totalmente abstracto, por lo que no es una afirmación sobre las conexiones entre las entidades del software.

Para identificar las asociaciones entre las clases conceptuales de la sección 3.2.1 se hará uso de la siguiente lista que contiene características comunes. Los ejemplos sugeridos son extraídos del dominio de la tecnología helpdesk:

Tabla 3.2. Asociaciones de las Clases Conceptuales. (1/2)

CATEGORÍA DE LA ASOCIACIÓN	EJEMPLOS
A es una parte física de B	RecursoInformatico-Inventario
A es una parte lógica de B	CierreTicket-TicketMantenimiento

[Fuente: Elaboración Propia]

Tabla 3.2. Asociaciones de las Clases Conceptuales. (2/2)

CATEGORÍA DE LA ASOCIACIÓN	EJEMPLOS
A está físicamente contenido en B	RecursoInformatico-Departamento
A está contenido lógicamente en B	TicketMantenimiento-ReporteEstadistico
A es una descripción de B	RespuestaMantenimiento- TicketMantenimiento
A es un elemento de línea (o renglón) en una transacción o reporte B	RecursoInformatico-TicketMantenimiento
A es miembro de B	Empleado-Departamento
A es una unidad organizacional de B	Departamento-Empresa
A usa o dirige a B	Técnico-TicketMantenimiento Supervisor-TicketMantenimiento
A se comunica con B	Supervisor-Tecnico Empleado-Tecnico
A se relaciona con una transacción B	RespuestaMantenimiento- TicketMantenimiento
A es una transacción relacionada con otra transacción B	StatusTicket-TicketMantenimiento
A es propiedad de B	RecursoInformatico-Departamento
A se conoce/introduce/registra/presenta/captura en B	Administrador-RecursoInformatico Administrador-Departamento

[Fuente: Elaboración Propia]

3.2.3. Diagrama del Modelo de Dominio:

El diagrama del Modelo de Dominio de la Figura 3.3 muestran el conjunto de conceptos y asociaciones idóneos para el representar el sistema de negocios para el cual se desarrollara el SAIME. En él se observa cómo se realiza el proceso de administración de recurso y solicitud de mantenimiento siguiendo la filosofía

helpdesk, donde se puede apreciar que el Empleado solicita un Ticket de Mantenimiento que describe la falla asociada a un determinado Recurso Informático perteneciente al Departamento al cual laboran, el Ticket de Mantenimiento es atendido por un Técnico, quien se encarga de emitir una Respuesta de Mantenimiento y de asignar el Status del Ticket que puede ser revisado por el Empleado, pero previamente certificado por un Supervisor quien se encarga de asignar y validar el trabajo del Técnico, así como también de determinar el Cierre del Ticket que indica que ya la solicitud del Empleado fue procesada.

La gestión y control del inventario de los Recursos Informáticos, estará a cargo del Administrador que a su vez se encarga de la gestión de los Departamentos. Tanto el Administrador como el Supervisor tienen autorización para solicitar Reportes Estadísticos asociados a los Tickets de Mantenimiento.

3.2.4. Glosario de Términos:

Es necesario contar con un lenguaje común para facilitar el intercambio de ideas. El glosario o diccionario (ver Tabla 3.3) incluye y define todos los términos que son de vital importancia para la comprensión del análisis que se realizará en las fases de desarrollo del presente estudio, los cuales se mencionan a continuación:

Tabla 3.3. Glosario de Términos. (1/3)

TÉRMINO	DESCRIPCIÓN
Administrador	Representa a la persona encargada de mantener actualizado el sistema.
Calificación de Mantenimiento	Se refiere a la valoración propinada por el empleado al mantenimiento que ya fue atendido por el técnico.
Cierre de Ticket	Se refiere al proceso que determina que ya el mantenimiento ha sido atendido correctamente.

[Fuente: Elaboración Propia]

Diagrama de Modelo de Dominio del SAIME

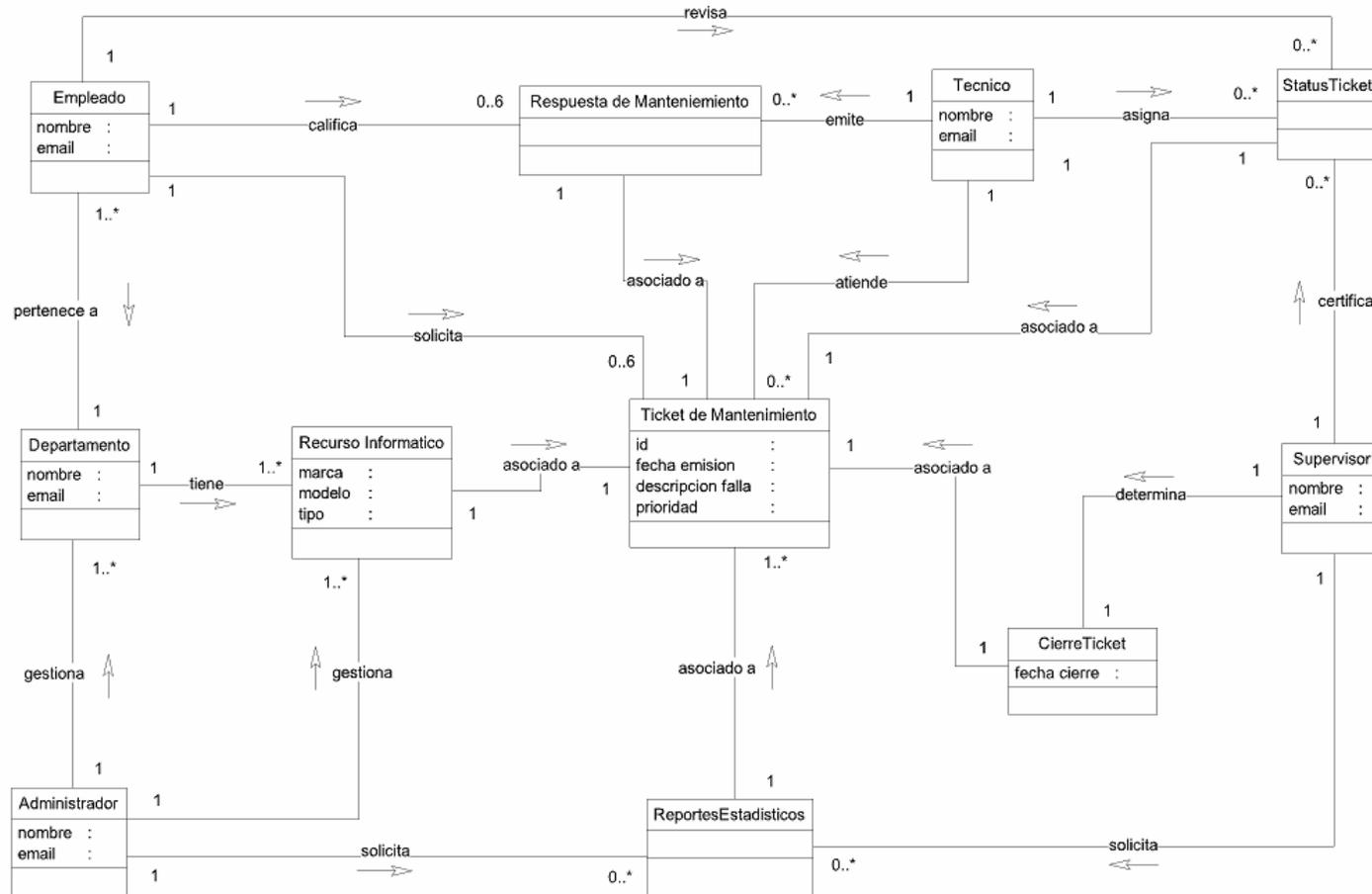


Figura 3.3. Diagrama del Modelo de Dominio [Fuente: Elaboración Propia]

Tabla 3.3. Glosario de Términos. (2/3)

TÉRMINO	DESCRIPCIÓN
Departamento	Unidad física donde pertenece el recurso informático que se le presta el servicio de mantenimiento y a la cual pertenece el empleado que emite la solicitud.
Empleado	Representa a la persona que reporta la falla de un recurso mediante la solicitud de un ticket de mantenimiento.
Falla	Es la terminación de la capacidad de un recurso informático para realizar una función requerida.
Prioridad de Ticket de Mantenimiento	Establece la precedencia de atención de un determinado ticket sobre otros.
Recurso Informático	Componente de Hardware o Software.
Reportes Estadísticos	Se refiere a la visualización de gráficas que generan las estadísticas de los mantenimientos.
Respuesta de Mantenimiento	Es el mantenimiento que se aplica cuando se ha producido el fallo y el paro súbito de un recurso informático.
Status del Ticket	Identifica el estado actual de una determinada solicitud de mantenimiento.
Supervisor	Representa a la persona encargada del control de las actividades relacionadas al mantenimiento de equipos.
Técnico	Representa a la persona encargada de dar soporte a las solicitudes que han sido emitidas en el sistema.

[Fuente: Elaboración Propia]

Tabla 3.3. Glosario de Términos. (3/3)

TÉRMINO	DESCRIPCIÓN
Ticket de Mantenimiento	Es un registro con toda la información necesaria para la realización de una labor de mantenimiento. Éste incluye la descripción del recurso, la falla que presenta, la prioridad de respuesta, entre otros.
Tipo de Recurso Informático	Representa los diferentes tipos de recursos informáticos a los que se realizan mantenimiento.

[Fuente: Elaboración Propia]

3.3. Riesgos del Sistema

En esta sección se presenta la descripción de los riesgos identificados durante la etapa inicial del proyecto. Para cada riesgo observado se valorarán sus efectos y contexto de aparición para el caso en que se convierta en un hecho. Además, se definirán estrategias para reducir la probabilidad del riesgo o para controlar sus posibles efectos, por lo que será necesario durante el desarrollo del proyecto revisar y actualizar los contenidos del análisis de riesgos en caso de que se detecten nuevos riesgos no visibles en este momento.

Una vez identificados los posibles riesgos serán evaluados considerando las siguientes perspectivas:

- Magnitud. Estimación de la importancia de sus efectos en caso de que se convierta en un hecho. Se evalúa como muy baja, baja, media, alta, muy alta o catastrófica.
- Descripción. Breve descripción textual.
- Impacto. Descripción textual de los efectos sobre el proyecto de la transformación del riesgo en un hecho.
- Indicadores. Magnitudes a observar para intuir la aparición del riesgo.

- Plan de acción. Medidas a tomar en el proyecto para evitar la aparición del riesgo o minimizar su futuro impacto, aplicadas antes de que el riesgo se convierta en un hecho.

- Plan de contingencia. Medidas a tomar en el proyecto una vez que el riesgo se haya transformado en un hecho.

En el proyecto en desarrollo, los riesgos que fueron identificados para el sistema de inventario y mantenimiento SAIME se muestran a continuación:

3.3.1. Falta de información inicial

- Magnitud. Media.

- Descripción. El equipo de desarrollo tiene dificultades a la hora de realizar sus objetivos por la falta de información suministrada por la empresa.

- Impacto. Puede suponer retrasos.

- Indicadores. No procede.

- Plan de acción. Realización de varias reuniones con el representante de la empresa para el solicitar información y aclarar las dificultades relativas a los objetivos del proyecto.

- Plan de contingencia. Si el riesgo se convierte en hecho, se revisará y modificará la documentación de diseño afectada. En caso necesario, dejarán de realizarse tareas menos importantes para centrarse en las principales. Se tratará de reajustar la planificación del proyecto.

3.3.2. Falta de experiencia con las herramientas utilizadas

- Magnitud. Baja.

- Descripción. El equipo de desarrollo tiene dificultades a la hora de realizar sus objetivos (tanto de documentación como de implementación) por su inexperiencia con los recursos software del proyecto.

- Impacto. Puede suponer retrasos.

- Indicadores. No procede.

- Plan de acción. Una parte del tiempo de desarrollo del proyecto se destinará al aprendizaje de las herramientas de documentación e implementación.

- Plan de contingencia. Si se produce un retraso por parte de un miembro del equipo de desarrollo, los demás miembros tratarán de ayudar a superarlo. Si no resultara, consultar a fuentes externas

3.3.3. Cambios en los requisitos

- Magnitud. Baja.

- Descripción. El cliente puede solicitar que se incorporen nuevos requisitos o que se modifiquen requisitos ya conocidos en cualquier momento del desarrollo del sistema.

Se considera más probable que aparezcan modificaciones durante las fases de Inicio y Elaboración del proyecto, por dos causas:

Al tratarse de las primeras fases del desarrollo, el cliente está aún descubriendo sus propias necesidades respecto a la aplicación deseada.

El propio proceso de descubrimiento y análisis de requisitos realizado por el equipo de desarrollo puede hacer aflorar nuevas ideas y necesidades a los ojos del cliente.

Aunque en estas fases se considere más probable, no se descarta en absoluto que el cliente pueda descubrir nuevas necesidades durante las fases posteriores del proyecto.

- Impacto. La incorporación o modificación de requisitos durante el desarrollo requerirá realizar cambios sobre gran parte de la documentación del producto elaborada con anterioridad al momento del cambio. Estas modificaciones serán fácilmente asumibles durante las dos primeras fases del proyecto, pero pueden suponer trastornos importantes durante las fases de Construcción y Transición.

- Indicadores.

El cliente anuncia al equipo de desarrollo el cambio de requisitos.

También puede extraerse información de conversaciones directas con el cliente.

- Plan de acción. Realización de varias reuniones con el cliente para la aclaración de requisitos. Relativamente frecuentes en las primeras iteraciones, y en descenso a medida que avanza el proyecto.

- Plan de contingencia. En las primeras fases se realizarán los cambios necesarios para incorporar los nuevos requisitos o los cambios necesarios. En las fases de Construcción y Transición se valorará la importancia de las modificaciones / requisitos nuevos frente a la cantidad de tiempo disponible para abordarlos.

En caso de que se decida aceptarlos, se revisarán los requisitos afectados, así como toda la documentación y código derivado de los mismos hasta el punto de aparición del cambio.

3.3.4. Diseño erróneo

- Magnitud. Alta

- Descripción. El diseño del sistema resulta inadecuado. Al realizar actividades de implementación puede encontrarse que el diseño carece del suficiente nivel de detalle o está mal enfocado, bien por la naturaleza del problema, o bien por restricciones de uso impuestas por tecnologías de terceros.

- Impacto. Puede introducir retrasos en el proyecto ante la necesidad de volver a considerar el diseño trazado. Requiere la actualización o modificación de los artefactos de diseño.

- Indicadores. La arquitectura no cumple las expectativas. Se complica la implementación.

- Plan de acción. Se desarrollará en paralelo un prototipo conteniendo la arquitectura del sistema para comprobar la validez de la misma. En caso de encontrarse errores o inconsistencias, podrá modificarse el diseño al mismo tiempo que la implementación del prototipo.

- Plan de contingencia. Se estudiará una solución acorde a los tiempos de plazo de que se dispone, se revisará y modificará la documentación de diseño afectada. La planificación se reajustará si fuera necesario.

3.3.5. Pérdida de datos

- Magnitud. Alta.
- Descripción. De alguna manera, se produce la pérdida de datos.
- Impacto. Variable, puede suponer una catástrofe, o un simple retraso.
- Indicadores. Ninguno.
- Plan de acción. Se usará una forja (repositorio) para el control de versiones. Se realizarán copias de seguridad en los ordenadores personales de cada uno de los miembros del equipo de desarrollo.
- Plan de contingencia. Recuperar la versión anterior a la versión perdida y tratar de reconstruirla.

3.3.6. Falla de operatividad

- Magnitud. Alta.
- Descripción. No hay operatividad del sistema en la intranet.
- Impacto. Interrupción del sistema, posible retraso.
- Indicadores. Falla en la energía eléctrica y/o problemas con el servidor
- Plan de acción. En caso de falla de la energía eléctrica sugiere esperar que repongan el servicio. Si el problema proviene de la intranet se debe reportar fallo de al departamento de comunicación y soporte.
- Plan de contingencia. Si el riesgo persiste durante un tiempo prolongado, se realizará el servicio a través de llamadas telefónica llevando nota de estos hasta que se restablezca la operatividad del sistema y se carguen los nuevos casos.

3.3.7. Hardware y/o Software inadecuados

- Magnitud. Alta.

- Descripción. El hardware y/o software donde se ejecutará el sistema debe estar adaptado a los requerimientos de la aplicación.
- Impacto. Inoperatividad del sistema.
- Indicadores. El hardware y/o software no cumple con los requerimientos de la aplicación.
- Plan de acción. Revisar si el software está bien configurado para un buen desempeño, si persiste el problema se reemplazará el componente incompatible con el sistema.
- Plan de contingencia. Ejecutar el sistema en otro equipo.

3.4. Requerimientos del Sistema

Los requisitos de información del sistema, es una de las actividades claves para el desarrollo del mismo, y por lo tanto, reflejan las necesidades del sistema para procesar la información. Estos requisitos son esenciales e importantes ya que determinan específicamente las necesidades.

El sistema propuesto, que permitirá la automatización de las actividades relacionadas a la administración de inventario y mantenimiento de equipos utilizados en los departamentos de la C.A. Hidrológica del Centro, HIDROCENTRO, debe cumplir con los siguientes requerimientos funcionales y no funcionales:

3.4.1. Requisitos funcionales

Los requisitos funcionales determinan las necesidades de información y automatización de los procesos de negocios, que tienen los usuarios y que el sistema Web debe cumplir luego de su desarrollo. A lo largo de una serie de reuniones con el asesor industrial, los usuarios y su representante, se logró identificar los siguientes requisitos funcionales:

- Permitir la gestión de inventario de todos los recursos informáticos de HIDROCENTRO, es decir, se debe poder insertar datos específicos a cada tipo de recurso, además de su posterior modificación o eliminación en caso de ser necesario.

- En el caso de que un recurso falle, el sistema debe proporcionar a los empleados de la empresa un medio donde podrán realizar solicitud de mantenimiento, así como también verificar el estado de su(s) solicitud(es).

- Las solicitudes deben ser detallada, por lo que el sistema debe registrar: un id único, la fecha de emisión, la descripción del recurso y la falla que presenta, así como también la prioridad de respuesta.

- La aplicación debe permitir a los supervisores del helpdesk poder hacer control y seguimiento a las solicitudes abiertas y cerradas a través de un modulo de gestión de mantenimiento que permita visualizar las solicitudes, asignar requerimientos a los técnicos de forma automática y emitir respuestas de mantenimiento.

- Permitir que los supervisores puedan certificar las respuestas de mantenimiento emitidas por los técnicos y determinar el cierre de casos abiertos.

- Validar usuarios, para evitar que usuarios no autorizados entren al sistema y los que si estén autorizado restringir el campo de acción según el rol que posea.

3.4.2. Requisitos no funcionales

Los requisitos no funcionales representan aquellos aspectos del sistema que no cumplen una función específica, pero que ayudan a la interacción entre los actores y el sistema. Son las restricciones de la aplicación, los atributos de calidad, los límites de memoria, requerimientos de seguridad, restricciones de software, restricciones de hardware, etc. En este caso se encontraron los siguientes:

- La arquitectura de software a usar debe ser la siguiente: Sistema operativo Windows o Linux, donde debe estar el Java Virtual Machine, PostgreSQL 8.2 como servidor de base de datos, NetBeans IDE o Eclipse como herramientas de desarrollo, el lenguaje de programación debe ser Java el cual debe poseer la librería JFreeChart que será la herramienta para el desarrollo de gráficos.

- Los estilos del diseño utilizado en el desarrollo de las páginas Web deben cumplir con los estándares de la Gerencia de Informática, así como también, los

nombres de las páginas, funciones, variables, procedimientos almacenados, vistas y tablas de datos del sistema.

- El acceso al sistema Web sólo puede realizarse a través de la intranet de la empresa.

- La información del password o clave del usuario debe almacenarse encriptado en la base de datos para brindar seguridad e integridad de los datos, ya que la organización maneja volúmenes de información de considerable importancia por lo que se hace necesario software de seguridad para ello.

- La interfaz Web de usuario debe ser agradable e intuitiva, que sea fácil de operar, ya que existen usuarios de nivel intermedio en computación.

3.5. Modelo de Casos de Uso

El modelo de casos de uso tiene como finalidad la captura de todos o parte de los requisitos funcionales del sistema, ya que describe la funcionalidad propuesta del nuevo sistema. El recurso principal del cual se vale este modelo es el diagrama de casos de uso, que muestra las distintas operaciones que se esperan de una aplicación o sistema, y cómo se relaciona con su entorno (usuarios u otras aplicaciones).

3.5.1. Identificación de Actores

Un actor representa un conjunto coherente de roles que los usuarios de los casos de uso representan al interactuar con éstos. Por lo tanto, una vez que se han identificado todos los actores del sistema, se tiene identificado el entorno externo al sistema.

Para el Sistema de Administración de Inventario y Mantenimiento de Equipos (SAIME), se identificaron cuatro actores fundamentales como lo son: Administrador del Sistema, Empleado, Técnico y Supervisor o Jefe de Departamento (Tabla 3.4).

Tabla 3.4. Descripción de Actores. (1/2)

ACTOR	DESCRIPCIÓN
Administrador del Sistema	Es el encargado de registrar y manipular la información contenida referente a los usuarios, departamentos, recursos informáticos y fallas relacionadas con el mantenimiento.

[Fuente: Elaboración Propia]

Tabla 3.4. Descripción de Actores. (2/2)

ACTOR	DESCRIPCIÓN
Empleado	Está facultado para emitir solicitudes de mantenimiento, a las que también puede hacerle seguimiento.
Supervisor / Jefe de Departamento	Es el actor principal del sistema, ya que representa a la persona encargada de llevar a cabo el control de las actividades relacionadas al mantenimiento de equipos.
Técnico	Es el encargado de recibir y dar soporte a las solicitudes que han sido emitidas en el sistema.

[Fuente: Elaboración Propia]

3.5.2. Identificación de los Casos de Uso

Los casos de uso son casos de utilización del sistema, descripciones narrativas de su comportamiento, gracias a los cuales se puede mejorar la comprensión de los requisitos. En la Tabla 3.5 se definen los casos de uso existentes en el sistema Web en desarrollo.

Tabla 3.5. Definición de Casos de Uso. (1/2)

CASO DE USO	DESCRIPCIÓN
Gestionar Inventario	Permite insertar, modificar y eliminar ajustes de inventario referente a recursos informáticos
Gestionar Departamentos	Permite procesar la información de los departamentos a los que presta servicio el sistema.

[Fuente: Elaboración Propia]

Tabla 3.5. Definición de Casos de Uso. (2/2)

CASO DE USO	DESCRIPCIÓN
Procesar Usuarios	Permite registrar y manipular la información referente a los usuarios que tienen acceso al sistema.
Consultar Asignaciones de Ticket	Permite visualizar al usuario actual del sistema los mantenimientos asignados a su cargo. Solo técnicos y supervisores están relacionados con este caso de uso.
Asignar Encargado de Ticket	Se refiere al registro procesado por el supervisor para fijar que una solicitud de mantenimiento sea atendida por un determinado técnico
Emitir Comentarios de Respuesta de Solicitud	Permite la expedición de comentarios para mantenimientos procesados, permitiendo así la comunicación directa empleado técnico y viceversa.
Consultar Tickets	Representa las consultas que el sistema puede realizar para hacer seguimiento a los mantenimientos procesados, no procesados, así como los que están pendiente.
Procesar Nuevo Ticket	Se refiere al registro que realiza el empleado para hacer la notificación de la falla que presenta un equipo en un determinado momento.

[Fuente: Elaboración Propia]

3.5.3. Descripción de los Casos de Uso

La descripción de los casos de uso mostrados en la Tabla 3.5 se presentan desde la Tabla 3.6 hasta la Tabla 3.13. Se describen actores involucrados, las precondiciones y poscondiciones. También se denota el flujo de sucesos para cada caso de uso que especifica la secuencia de acciones del mismo, es decir, el flujo de sucesos especifica lo que hace el sistema cuando un actor invoca un caso de uso, y como; dicho actor interactúa con el caso de uso invocado.

Tabla 3.6. Gestionar Inventario. (1/2)

GESTIONAR INVENTARIO	
Actor Principal:	Administrador del Sistema
Personal Involucrado e intereses:	
Administrador del Sistema: Insertar, Modificar y Eliminar nuevos recursos informáticos al inventario	
Precondiciones:	Ninguna
Pos condiciones:	Gestión de Inventario
Escenario principal de éxito (flujo básico)	
<p>A. El administrador del sistema invoca al caso de uso insertar recurso informático.</p> <ol style="list-style-type: none"> 1. Se registran los datos asociados al nuevo recurso. 2. El actor solicita guardar los datos. 3. Se almacenan permanentemente los datos. 4. Finaliza el caso de uso. <p>B. El administrador del sistema invoca al caso de uso modificar recurso.</p> <ol style="list-style-type: none"> 1. Se selecciona el recurso a modificar. 2. El sistema despliega los datos del recurso informático que el usuario desea 3. modificar. 4. Se modifican los datos deseados. 5. El actor salva los cambios. 6. El sistema guarda las modificaciones. 	

[Fuente: Elaboración Propia]

Tabla 3.6. Gestionar Inventario. (2/2)

7. El sistema informa al usuario que las modificaciones hechas a los datos del recurso informático se han guardado satisfactoriamente.
8. Finaliza el caso de uso.
C. El administrador del sistema invoca al caso de uso eliminar recurso.
1. Se selecciona el recurso a eliminar.
2. El sistema elimina de la base de datos la información correspondiente al recurso.
3. El sistema pregunta si se desea eliminar otro recurso.
4. El administrador selecciona cancelar y vuelve a la pantalla anterior.
5. Finaliza el caso de uso.

Extensiones (flujos alternativos)

Alternativa al flujo A.2. El actor Cancela y se vuelve a la pantalla anterior.

Alternativa al flujo B.4. El actor Cancela y se vuelve a la pantalla anterior.

Alternativa al flujo C.4. El administrador selecciona aceptar y se devuelve el proceso al flujo C.1.

[Fuente: Elaboración Propia]

Tabla 3.7. Gestionar Departamento. (1/2)

GESTIONAR DEPARTAMENTOS	
Actor Principal:	Administrador del Sistema
Personal Involucrado e intereses:	
Administrador del Sistema: Insertar y Modificar departamentos al sistema.	
Precondiciones:	Ninguna
Pos condiciones:	Gestión de Departamento

[Fuente: Elaboración Propia]

Tabla 3.7. Gestionar Departamento. (2/2)

Escenario principal de éxito (flujo básico)	
A.	El administrador del sistema invoca al caso de uso insertar departamento. <ol style="list-style-type: none"> 1. Se registran los datos asociados al nuevo departamento 2. El actor solicita guardar los datos. 3. Se almacenan permanentemente los datos. 4. Finaliza el caso de uso.
B.	El administrador del sistema invoca al caso de uso modificar departamento. <ol style="list-style-type: none"> 1. Se selecciona el departamento a modificar. 2. El sistema despliega los datos del departamento que el usuario desea modificar. 3. Se modifican los datos deseados. 4. El actor salva los cambios. 5. El sistema guarda las modificaciones. 6. El sistema informa al usuario que las modificaciones hechas a los datos del recurso informático se han guardado satisfactoriamente 7. Finaliza el caso de uso.
Extensiones (flujos alternativos)	
	Alternativa al flujo A.2. El actor Cancela y se vuelve a la pantalla anterior.
	Alternativa al flujo B.4. El actor Cancela y se vuelve a la pantalla anterior.

[Fuente: Elaboración Propia]

Tabla 3.8. Procesar Usuarios. (1/3)

PROCESAR USUARIO	
Actor Principal:	Administrador del Sistema

[Fuente: Elaboración Propia]

Tabla 3.8. Procesar Usuarios. (2/3)

Personal Involucrado e intereses:	
Administrador del Sistema: Administrar todo tipo de usuario del sistema	
Supervisor: Solo puede administrar usuario del sistema con rol de técnico o empleado	
Técnico: Solo puede administrar usuario del sistema con rol de empleado	
Precondiciones:	Ninguna
Pos condiciones:	Administración de usuario
Escenario principal de éxito (flujo básico)	
A. El administrador del sistema invoca al caso de uso insertar usuario.	
<ol style="list-style-type: none"> 1. Se registran los datos asociados al nuevo usuario. 2. El actor solicita guardar los datos. 3. Se almacenan permanentemente los datos. 4. Finaliza el caso de uso. 	
B. El administrador del sistema invoca al caso de uso modificar usuario.	
<ol style="list-style-type: none"> 1. Se selecciona el usuario a modificar. 2. El sistema despliega los datos del usuario que el actor desea modificar. 3. Se modifican los datos deseados. 4. El actor salva los cambios. 5. El sistema guarda las modificaciones. 6. El sistema informa al actor que las modificaciones hechas a los datos del usuario se han guardado satisfactoriamente. 7. Finaliza el caso de uso. 	
C. El administrador del sistema invoca al caso de uso eliminar usuario.	
<ol style="list-style-type: none"> 1. Se selecciona el usuario a eliminar. 	

[Fuente: Elaboración Propia]

Tabla 3.8. Procesar Usuarios. (3/3)

<ol style="list-style-type: none"> 2. El sistema elimina de la base de datos la información correspondiente al usuario. 3. El sistema pregunta si se desea eliminar otro recurso 4. El administrador selecciona cancelar y vuelve a la pantalla anterior. 5. Finaliza el caso de uso.
Extensiones (flujos alternativos)
Alternativa al flujo A.2. El actor Cancela y se vuelve a la pantalla anterior.
Alternativa al flujo B.4. El actor Cancela y se vuelve a la pantalla anterior.
Alternativa al flujo C.4. El administrador selecciona aceptar C.4 y se devuelve el proceso al flujo C.1.

[Fuente: Elaboración Propia]

Tabla 3.9. Consultar Asignaciones de Tickets. (1/2)

CONSULTAR ASIGNACIONES DE TICKETS	
Actor Principal:	Técnico
Personal Involucrado e intereses:	
Técnico: Visualizar los tickets de mantenimientos que le corresponde atender.	
Supervisor: Monitorear el estado de los mantenimientos asignados a los técnicos.	
Precondiciones:	Que haya sido ejecutado y cargado datos en el caso de uso Asignar Encargado de Tickets.
Pos condiciones:	El sistema muestra los tickets asignados por el supervisor.

[Fuente: Elaboración Propia]

Tabla 3.9. Consultar Asignaciones de Tickets. (2/2)

Escenario principal de éxito (flujo básico)
<p>A. El actor invoca al caso de uso asignar encargado de tickets:</p> <ol style="list-style-type: none"> 1. El sistema muestra en tablas datos generales de los tickets de mantenimiento asignados y especificaciones del técnico encargado de atenderlo. 2. Se selecciona la columna en la que desea ver los datos en forma ordenada. 3. Finaliza el caso de uso.
Extensiones (flujos alternativos)
No hay flujos alternativos

[Fuente: Elaboración Propia]

Tabla 3.10. Asignar Encargado de Tickets. (1/2)

ASIGNAR ENCARGADO DE TICKETS	
Actor Principal:	Supervisor
Personal Involucrado e intereses:	
Supervisor: Establecer asignaciones o reasignaciones de tickets de mantenimiento a miembros de soporte técnico.	
Precondiciones:	Que haya usuarios de rol de Técnico en el sistema.
Pos condiciones:	El sistema registra y visualiza asignaciones de tickets.
Escenario principal de éxito (flujo básico)	
<p>A. El supervisor invoca el caso de uso asignar encargado de tickets:</p> <ol style="list-style-type: none"> 1. El sistema muestra los tickets que no han sido atendidos. 2. Se selecciona un ticket a asignar o reasignar según sea el caso. 3. Se asigna un técnico encargado de atender el ticket. 4. Se confirma la asignación. 5. Finaliza el caso de uso. 	

[Fuente: Elaboración Propia]

Tabla 3.10. Asignar Encargado de Tickets. (2/2)

Extensiones (flujos alternativos)
Alternativa al flujo A.1. El sistema muestra un mensaje indicando que no hay tickets pendientes y vuelve al menú principal.
Alternativa al flujo A.3. El sistema muestra un mensaje de error debido a que no existe personal (técnico) a quien asignar el ticket y vuelve al menú principal.
[Fuente: Elaboración Propia]

Tabla 3.11. Emitir Comentarios de Respuesta de Solicitud. (1/2)

EMITIR COMENTARIOS DE RESPUESTA DE SOLICITUD	
Actor Principal:	Técnico
Personal Involucrado e intereses:	
Supervisor / Técnico / Empleado: Añadir observaciones/comentarios a los tickets	
Precondiciones:	Que haya sido ejecutado el Caso de Uso Procesar Nuevo Ticket y el Caso de Uso Asignar Encargado de Ticket
Pos condiciones:	Inserción de comentarios en relación a un ticket de mantenimiento.
Escenario de éxito (flujo básico)	
<p>A. El actor invoca el caso de uso emitir comentario en respuesta de solicitud:</p> <ol style="list-style-type: none"> 1. El sistema muestra los tickets de mantenimientos correspondiente a un determinado usuario. 2. Se selecciona un ticket 3. Se registra el comentario asociado al ticket. 4. El sistema envía un correo electrónico tanto al empleado como al técnico correspondiente informando que ha sido emitido un comentario en relación a un mantenimiento que se encuentra en proceso. 5. Finaliza el caso de uso. 	
[Fuente: Elaboración Propia]	

Tabla 3.11. Emitir Comentarios de Respuesta de Solicitud. (2/2)

Extensiones (flujos alternativos)
No hay flujos alternativos

[Fuente: Elaboración Propia]

Tabla 3.12. Consultar Tickets.

CONSULTAR TICKETS	
Actor Principal:	Empleado
Personal Involucrado e intereses:	
Supervisor / Empleado: Visualizar las solicitudes de tickets que ha realizado.	
Precondiciones:	Que haya sido ejecutado el Caso de Uso Procesar Nuevo Ticket.
Pos condiciones:	El sistema muestra una tabla de datos correspondientes a todos los tickets de mantenimiento pertenecientes al usuario.
Escenario principal de éxito (flujo básico)	

A. El actor invoca al caso de uso consultar ticket.

1. El sistema despliega unos campos para introducir los datos de el/los tickets a buscar.
2. El actor puede realiza la búsqueda de el/los tickets de su interés.
3. Dependiendo del tipo de búsqueda seleccionada se muestran los resultados.
4. Finaliza el caso de uso.

Extensiones (flujos alternativos)
Alternativa al flujo A.3. El sistema muestra un mensaje donde se le informa al actor que no se encontraron resultados.

[Fuente: Elaboración Propia]

Tabla 3.13. Procesar Nuevo Ticket.

PROCESAR NUEVO TICKET	
Actor Principal:	Empleado
Personal Involucrado e intereses:	
Supervisor / Empleado: Solicitar un nuevo ticket de mantenimiento.	
Precondiciones:	Ninguna
Pos condiciones:	Registro en sistema de un nuevo ticket de mantenimiento.
Escenario principal de éxito (flujo básico)	
<p>A. El usuario invoca al caso de uso procesar nuevo ticket.</p> <ol style="list-style-type: none"> 1. El sistema despliega unos campos para introducir los datos del ticket a registrar. 2. El actor solicita guardar los datos. 3. Se almacenan permanentemente los datos. 4. El sistema limpia todos los campos. 5. El sistema pregunta si se desea registrar otro ticket. 6. El actor selecciona cancelar y vuelve a la pantalla anterior. 7. Finaliza el caso de uso. 	
Extensiones (flujos alternativos)	
<p>Alternativa al flujo A.5. El actor selecciona aceptar y se devuelve el proceso al flujo A.1.</p>	

[Fuente: Elaboración Propia]

Con este proceso de ilustración de los casos de uso se obtiene una visión mas refinada del diseño, necesaria para llevar a cabo el diagrama de caso de uso general del sistema que se muestra a continuación.

3.5.4. Diagrama de Casos de Uso

En el diagrama de Casos de Uso representado en la Figura 3.4 se puede observar que el administrador del sistema es el encargado de ejecutar el proceso de recolección de la información tanto de los equipos a los cuales se le realiza el proceso de mantenimiento, como a los departamentos donde pertenecen dichos equipos y también del personal involucrado en el mantenimiento de los mismos.

La actividad principal del sistema es gestionar la programación de mantenimiento, donde el supervisor de mantenimiento genera la programación de actividades a realizar asignando los tickets de mantenimiento a los miembros de soporte técnico a su cargo de acuerdo a sus capacidades individuales.

El supervisor puede hacer consultas de tickets según sus asignaciones en un determinado momento a través del Caso de Uso Consultar Asignaciones de Tickets, lo que le permite hacer seguimiento al desempeño laboral del personal técnico.

El sistema SAIME permite a los miembros de soporte técnico que tengan tickets de mantenimiento encargados emitir comentarios de respuestas a estos, para que el empleado que realizó la solicitud pueda hacer un seguimiento su estado a través del caso de uso consultar tickets.

El Caso de Uso Procesar Nuevo Ticket permite al empleado realizar solicitudes de mantenimiento, la cual incluye un número correlativo, la descripción del recurso, la falla que presenta y la prioridad de atención. La falla puede ser descrita textualmente por el empleado o puede ser escogida de una lista de fallas predeterminadas que están en el sistema.

Diagrama de Casos de Uso del SAIME

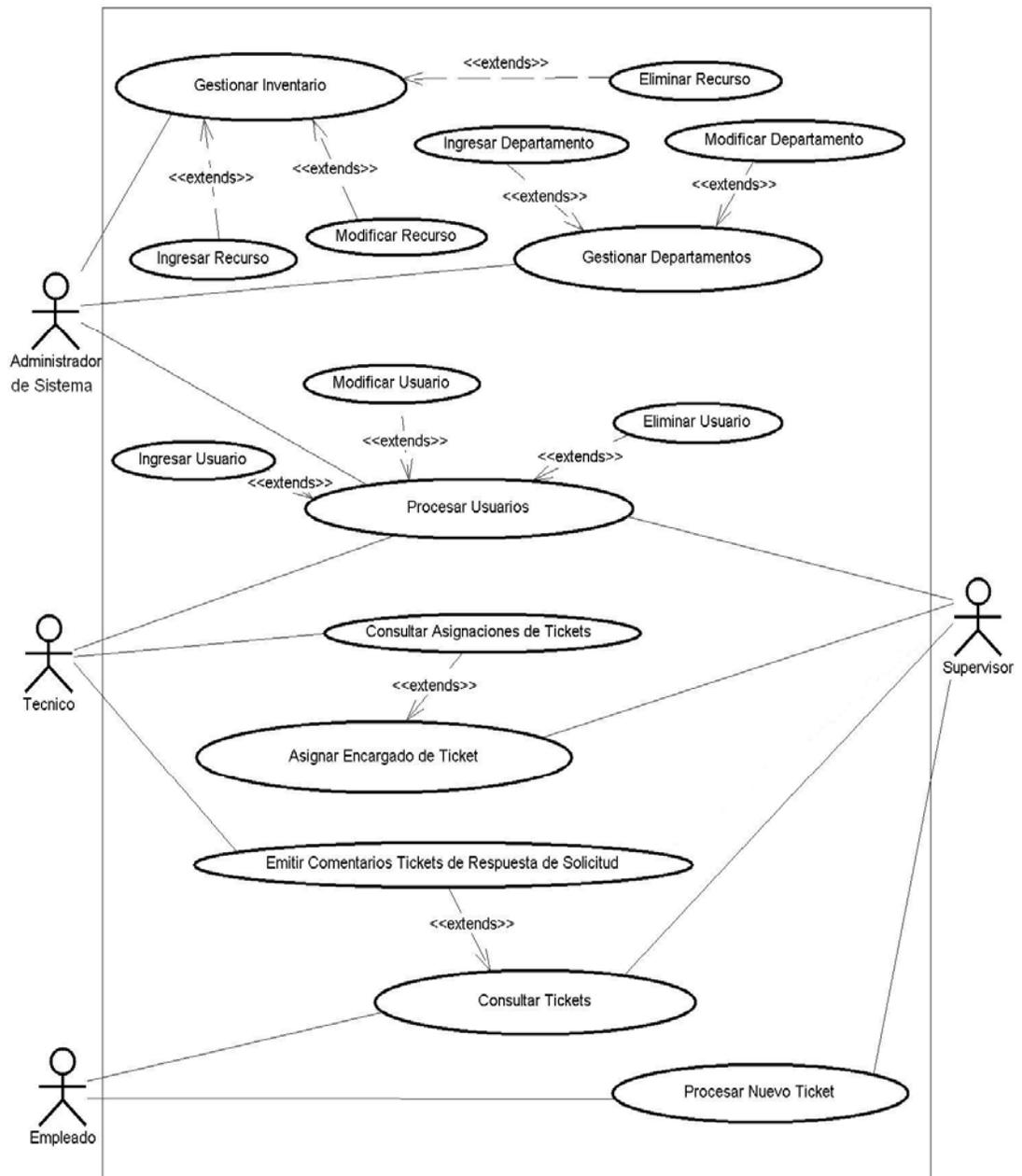


Figura 3.4. Diagrama de Casos de Uso General de SAIME

3.5. Análisis

El objetivo de este flujo de trabajo es traducir los requisitos a detalle de manera que especifique cómo implementar el sistema. En el análisis se obtiene una visión del sistema que se preocupa de “ver qué hace”, de modo que sólo se interesa por los requisitos funcionales.

A diferencia del lenguaje intuitivo utilizado en la captura de requisitos, en el flujo de trabajo análisis se utiliza un lenguaje basado en modelo de casos de uso y clases de análisis, que permite refinar el flujo anterior y comenzar a indagar en aspectos internos del sistema. En este flujo también se describe el diagrama de paquetes de análisis, que permite precisar y fundamentar la línea base de la arquitectura del sistema.

3.5.1. Análisis de los Casos de Uso

El análisis de los requisitos capturados en forma de casos de uso es un flujo de trabajo que permite de manera iterativa e incremental la construcción del modelo de análisis del sistema. Este modelo de análisis crece a medida que se analizan más y más casos de uso, ofreciendo una comprensión más precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que ayude a estructurar el sistema entero.

3.5.1.1. Identificación de las Clases de Análisis

Las clases de análisis, también llamados objetos de análisis, son clases estereotipadas que representan un modelo conceptual para los elementos del sistema que tienen responsabilidad y comportamiento. Hay tres tipos de clases de análisis usadas en todo el modelo de análisis: Clases de interfaz, Clases de entidad y Clases de control

Las clases de interfaz modelan las partes del sistema que dependen de sus actores, lo cual implica que clarifican y reúnen los requisitos en los límites del sistema. El prototipo de clase de interfaz se muestra en la Figura 3.5



Figura 3.5. Prototipo de Clase de Interfaz

A continuación se detallan las clases de interfaz que se identificaron en el modelo de análisis (ver Tabla 3.14):

Tabla 3.14. Clases de Interfaz (1/3)

CLASE DE INTERFAZ	DESCRIPCIÓN
IU Gestionar Inventario	Permite al administrador del sistema insertar, modificar y eliminar recursos informáticos del inventario
IU Insertar Recurso	Permite al administrador del sistema insertar un nuevo recurso informático al inventario.
IU Modificar Recurso	Permite al administrador del sistema modificar información de un recurso informático del inventario.
IU Eliminar Recurso	Permite al administrador del sistema eliminar un recurso informático del inventario.

Tabla 3.14. Clases de Interfaz (2/3)

CLASE DE INTERFAZ	DESCRIPCIÓN
IU Gestionar Departamentos	Permite al administrador del sistema insertar y modificar información de los departamentos que se prestan servicio.
IU Insertar Departamento	Permite al administrador del sistema insertar un nuevo departamento.
IU Modificar Departamento	Permite al administrador del sistema modificar información de un departamento.
IU Consultar Estadísticas	Permite al administrador del sistema generar graficas estadísticas sobre la información almacenada relacionada con los mantenimiento
IU Procesar Usuarios	Permite al supervisor insertar, modificar y eliminar información de los usuarios que tienen acceso al sistema.
IU Insertar Usuario	Permite al supervisor insertar un nuevo usuario al sistema.
IU Modificar Usuario	Permite al supervisor modificar información de un usuario del sistema.
IU Eliminar Usuario	Permite al supervisor eliminar un usuario del sistema.
IU Asignar Encargado de Ticket	Permite al supervisor seleccionar al miembro de soporte técnico encargado de atender un determinado ticket
IU Consultar Asignaciones	Permite al miembro de soporte técnico consultar las asignaciones de tickets que tiene pendiente en sistema.

[Fuente: Elaboración Propia]

Tabla 3.14. Clases de Interfaz (3/3)

CLASE DE INTERFAZ	DESCRIPCIÓN
IU Emitir Comentarios	Permite al miembro de soporte técnico emitir comentarios en respuesta al ticket asignado a su cargo.
IU Consultar Tickets	Permite al empleado realizar seguimiento de los tickets que ha emitido en el sistema.
IU Procesar Nuevo Ticket	Permite al empleado cargar los datos de la falla presentada en un recurso informático.

[Fuente: Elaboración Propia]

Las clases de control representan coordinación, secuencia, transacciones y control de otros objetos, y se usan con frecuencia para encapsular el control de un caso de uso en concreto. El prototipo de la clase de control representa un servicio que una instancia de la clase puede realizar.



Figura 3.6. Prototipo de Clase de Control

[Fuente: Binder, 2000]

En la Figura 3.6 se puede observar el estereotipo de la clase de control, a continuación se especificarán las clases que se encontraron en el diseño de análisis (ver Tabla 3.15):

Tabla 3.15. Clases de Control (1/2)

CLASE DE CONTROL	DESCRIPCIÓN
Gestor Gestionar Inventario	Actualiza los registros del inventario en el sistema.
Gestor Insertar Recurso	Inserta nuevos recursos informáticos al inventario.
Gestor Modificar Recurso	Modifica datos de recursos informáticos del inventario.
Gestor Eliminar Recurso	Elimina registros de recursos informáticos del inventario.
Gestor Gestionar Departamentos	Actualiza el registro de departamentos en el sistema.
Gestor Insertar Departamento	Inserta nuevos departamentos en el sistema.
Gestor Modificar Departamento	Modifica datos de departamentos en el sistema.
Gestor Eliminar Departamento	Elimina datos de departamentos en el sistema. Por motivos de integridad de los datos este gestor no se implementara en el sistema
Gestor Procesar Usuarios	Actualiza los registros de usuarios en el sistema.
Gestor Insertar Usuario	Inserta nuevos usuarios al sistema.
Gestor Modificar Usuario	Modifica datos de los usuarios del sistema.

[Fuente: Elaboración Propia]

Tabla 3.15. Clases de Control (2/2)

CLASE DE CONTROL	DESCRIPCIÓN
Gestor Eliminar Usuario	Elimina registros de usuarios del sistema.
Gestor Asignar Encargado de Ticket	Procesa la asignación de los tickets de mantenimiento que debe atender un determinado miembro de soporte técnico
Gestor Consultar Asignaciones de Ticket	Procesa la visualización de estado los tickets de mantenimiento encargado a un determinado miembro de soporte técnico.
Gestor Tickets	Procesa las actividades relacionadas al registro, consultas y comentarios de tickets de mantenimiento.

[Fuente: Elaboración Propia]

Las clases de entidad, por su parte, se utilizan para modelar información que posee una vida larga y que es a menudo persistente. Modela la información y el comportamiento asociado de algún fenómeno o concepto, como una persona, un objeto del mundo real o un suceso del mundo real. La clase de entidad permite modelar la información. La Figura 3.7 muestra el prototipo de la clase entidad.



Figura 3.7. Prototipo de Clase de Entidad

[Fuente: Binder, 2000]

A continuación se especifican las clases entidad encontradas en el modelo de análisis (ver Tabla 3.16).

Tabla 3.16. Clase de Entidad.

CLASE DE ENTIDAD	DESCRIPCIÓN
Asignaciones Tickets	Representa el registro donde se asocia un determinado ticket de mantenimiento a un miembro de soporte técnico que se ocupe de atenderlo
Comentarios Tickets	Representa la información textual de las actividades que se realizan al recurso una vez emitido un ticket de mantenimiento.
Detalles Recurso Informático	Representa todas las especificaciones de un recurso y descripción técnica del mismo.
Departamento	Representa la información del área de trabajo donde opera un determinado recurso.
Estadísticas	Representa el registro que genera las funciones estadísticas aplicada a los tickets de mantenimiento.
Falla	Representa el registro que describe una falla determinada.
Inventario	Representa las descripciones generales de un determinado recurso informático
Rol	Representa el rol que determina los permisos en el sistema de un determinado usuario
Ticket	Representa el registro que se genera cuando ocurre una falla en un determinado recurso informático
Usuarios	Representa a cada uno de los empleados que desempeñan un rol específico en el sistema

[Fuente: Elaboración Propia]

3.5.1.2. Diagramas de las Clases de Análisis

Un diagrama de clase de análisis representa una abstracción de una o varias clases y/o subsistemas del diseño del sistema, y constituye una colección de clases que describe los requisitos funcionales del sistema.

En el diagrama de colaboración del caso de uso gestionar inventario, como se muestra en la Figura 3.8, el administrador del sistema usa las interfaces de gestionar inventario para acceder solicitar a través del gestor gestionar inventario el ingreso, modificación o eliminación del registro de los recursos informáticos.

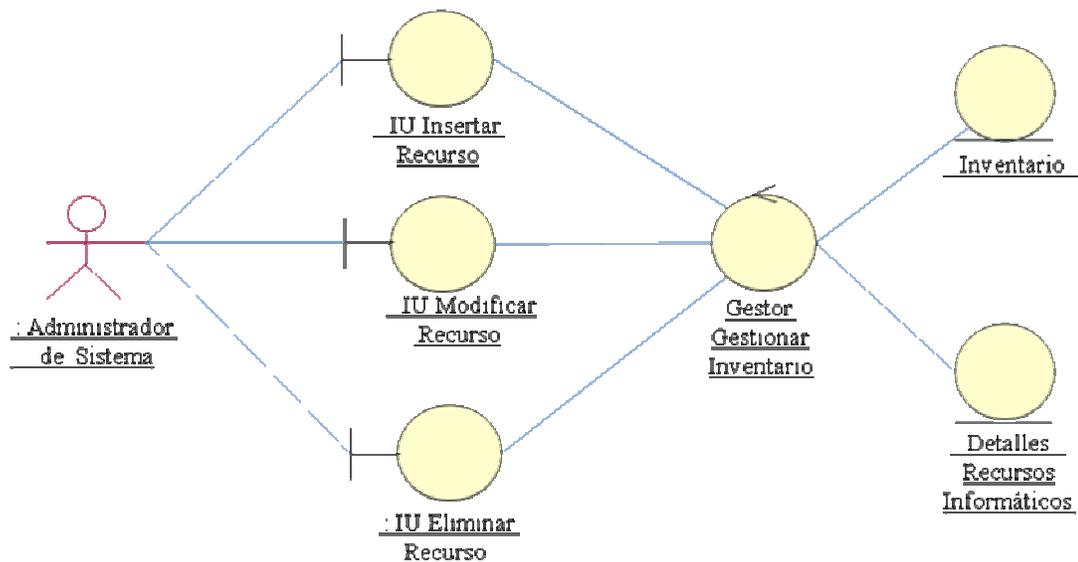


Figura 3.8. Diagrama de Clase de Análisis para el Caso de Uso Gestionar Inventario

[Fuente: Elaboración Propia]

Las principales actividades relacionadas con la programación de los mantenimientos realizada por el supervisor, se observan en la Figura 3.9 con el diagrama de clases de análisis de los casos de uso Asignar Encargado de Ticket y Consultar Asignaciones, donde se especificó una clase de interfaz denominada Interfaz Asignar Encargado de Ticket, encargada de la interacción con el supervisor,

que por medio del proceso Gestor Asignar Encargado de Ticket permite establecer asignaciones de tickets al personal de soporte técnico. La otra clase de clase de interfaz definida es Consultar Asignaciones, que a través de la clase de control Gestor Consultar Asignaciones procesa las consultas de las asignaciones ya programadas.

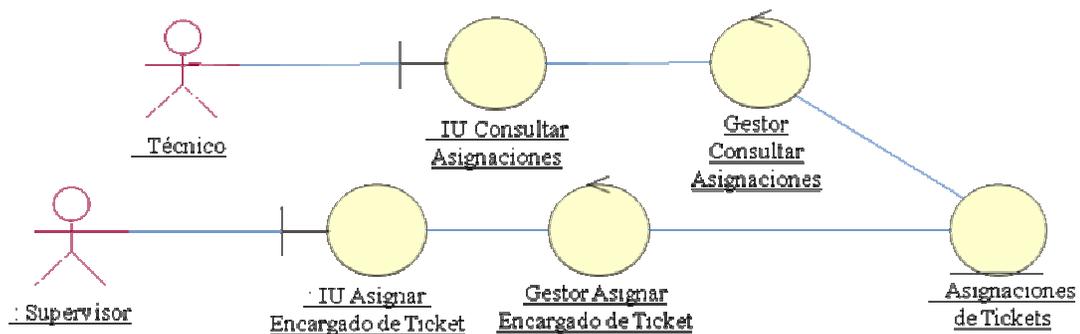


Figura 3.9. Diagrama de Clase de Análisis para los Casos de Uso Asignar Encargado de Ticket y Consultar Asignaciones de Tickets
[Fuente: Elaboración Propia]

El diagrama de clases de análisis de la Figura 3.10 agrupa las principales actividades relacionadas con la ejecución de mantenimientos, en él se representan los casos de uso Procesar Nuevo Ticket, Consultar Tickets y Emitir Comentarios Tickets de Respuesta de Solicitud, donde el Empleado selecciona la clase de interfaz Nuevo Ticket, en la cual se introducen los datos para la solicitud del mantenimiento, la siguiente clase de interfaz definida como Consultar Ticket es activada por el Empleado para hacer seguimiento a los mantenimiento solicitados, y por último la clase de interfaz Comentar responde al caso de uso Emitir Comentario en Respuesta de Solicitud, en donde tanto por el Empleado como el Técnico pueden añadir acotaciones, notas y/o observaciones a determinados tickets de mantenimiento. Cada una de estas interfaces es manejada por la clase de control Gestor Tickets, que se encarga de procesar la ejecución del mantenimiento.

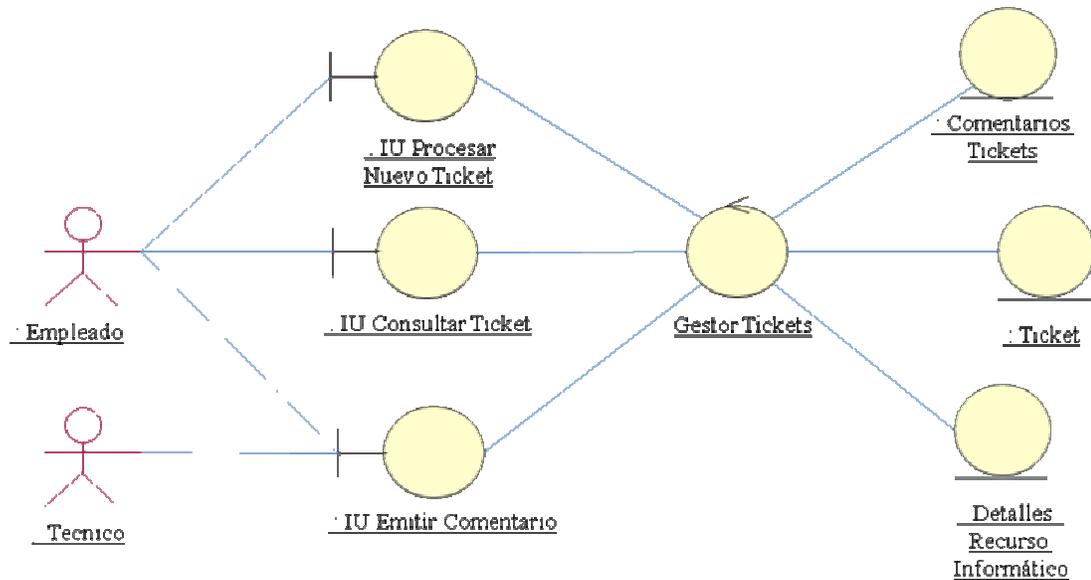


Figura 3.10. Diagrama de Clase de Análisis para los Casos de Uso Procesar Ticket, Consultar Ticket y Emitir Comentario de Respuesta de Solicitud
[Fuente: Elaboración Propia]

3.5.1.3. Diagramas de las Colaboración

En los diagramas de colaboración, se muestran las interacciones entre objetos creando enlaces entre ellos y añadiendo mensajes a esos enlaces. El nombre de un mensaje debe denotar el propósito del objeto invocante en la interacción con el objeto invocado.

En el diagrama de colaboración del caso de uso Gestionar Inventario, como se muestra en la Figura 3.11, muestra las interacciones de la Figura 3.8 esta vez con mensajes que describen el propósito de los objetos que interactúan en el diagrama, en donde el administrador tiene 3 flujos básicos que representan las opciones que puede realizar, los cuales son: Solicitar Ingreso de Recurso, Solicitar Modificación Recurso, Solicitar Eliminación de Recurso.

Para registrar un nuevo recurso al inventario, el administrador Solicita Ingreso de Recurso y Selecciona el Tipo de Recurso, seguidamente el sistema activa la

interfaz Ingresar Recurso donde se ingresan los datos que serán procesados por el Gestor Gestionar Inventario para verificar que este todo correcto antes de Cargar el Recurso al Inventario y a Detalles Recurso Informático.

Si el administrador Solicita Modificación de Recurso, el sistema activa la interfaz Modificar Recurso donde se Selecciona el Recurso a Modificar, una vez realizada la modificación los datos que son procesados por el Gestor Gestionar Inventario para verificar que este todo correcto antes de Actualizar el Inventario y los Detalles del Recurso Informático.

En caso de Solicitar la Eliminación de Recurso, se activara la interfaz Eliminar Recurso donde se Selecciona el Recurso a Eliminar, seguidamente se confirma para dar paso al Gestor Gestionar Inventario que verifique que este todo correcto antes de Eliminar el Recurso.

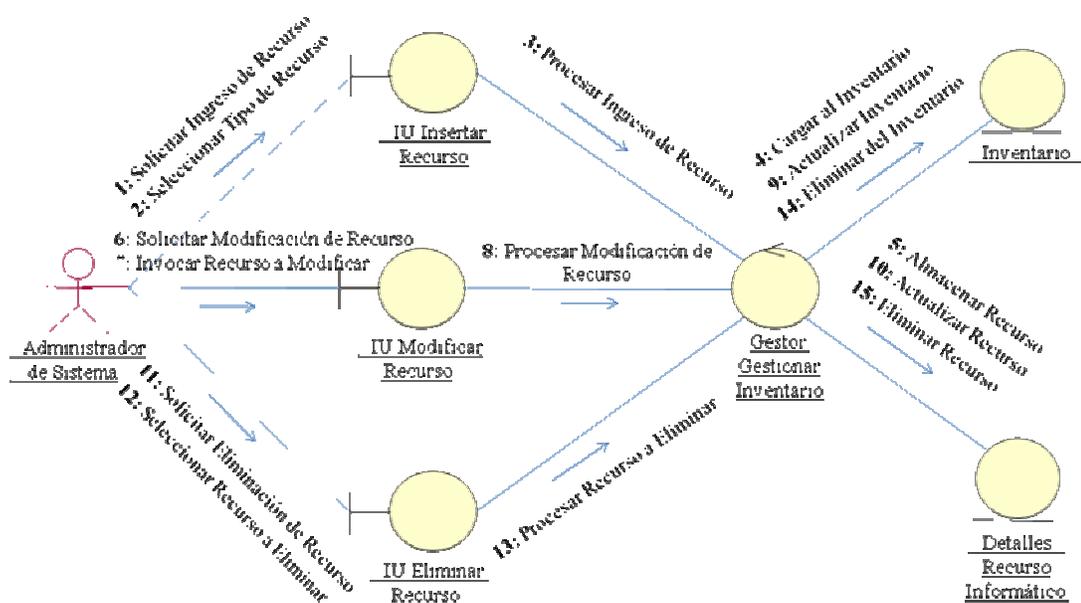


Figura 3.11. Diagrama de Colaboración para el Caso de Uso Gestionar Inventario

[Fuente: Elaboración Propia]

La Figura 3.12, representa el diagrama de colaboración para los casos de uso Asignar Encargado Ticket y Consultar Asignación Ticket. El supervisor al Solicitar Asignación de Encargado de Ticket, activa la interfaz Asignar Encargado de Ticket para realizar la asignación de mantenimientos a técnicos verificando según sus cualidades al ideal según la falla descrita en los tickets, esta asignación es procesada a través del Gestor Asignar Encargado de Tickets y almacenada en la entidad Asignaciones.

En el caso del técnico quiera realizar seguimiento al trabajo puede invocar la la interfaz Consultar Asignaciones, la cual es procesada por medio del Gestor Consultar Asignaciones que se encarga de Obtener las Asignaciones.

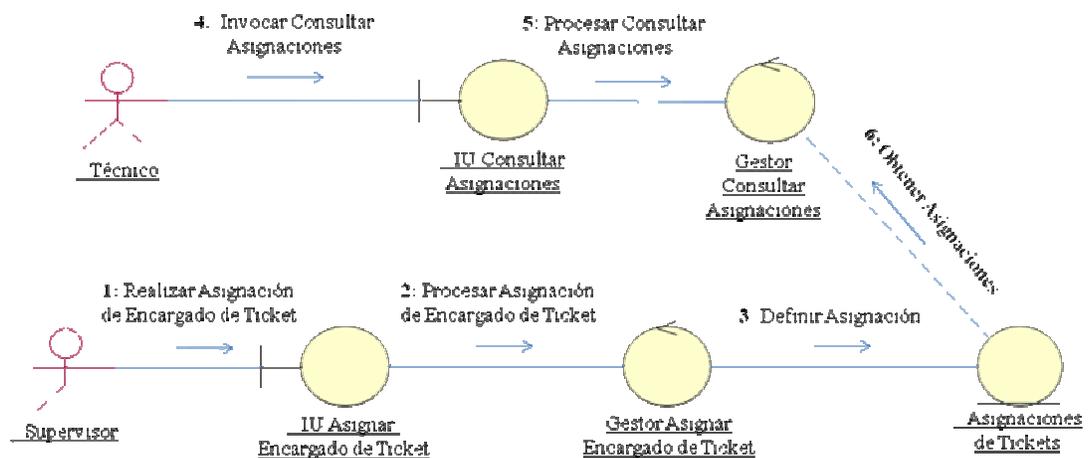


Figura 3.12. Diagrama de Colaboración para los Casos de Uso Asignar Encargado de Ticket y Consultar Asignaciones de Tickets.

[Fuente: Elaboración Propia]

El diagrama de colaboración de la Figura 3.13 representa los casos de Procesar Nuevo Ticket, Consultar Ticket y Emitir Comentario Ticket de Respuesta de Solicitud, tal como fue explicado en la Figura 3.10 estas interacciones agrupan las principales actividades relacionadas con la ejecución de los mantenimientos.

Cuando ocurre una falla en un recurso informático el actor Empleado es el encargado de solicitar mantenimiento en el sistema, describiendo los detalles de este a través de la interfaz Nuevo Ticket, el Gestor Tickets es el encargado de procesar de que todos los datos estén correctos para posteriormente Almacenar el Ticket e Inactivar el Recurso declarando su estado como inactivo.

Para hacer seguimiento de la solicitud, el Empleado puede solicitar al sistema la interfaz Consultar Tickets, que por medio del Gestor Tickets obtiene y despliega en pantalla los detalles de los Tickets que ha solicitado.

Tanto el Empleado como el Técnico pueden solicitar al sistema la interfaz Emitir Comentarios, en donde seleccionan un Ticket y le añade un comentario, que por medio del Gestor Ticket es procesado para ser almacenado en la entidad Comentarios de Mantenimiento.

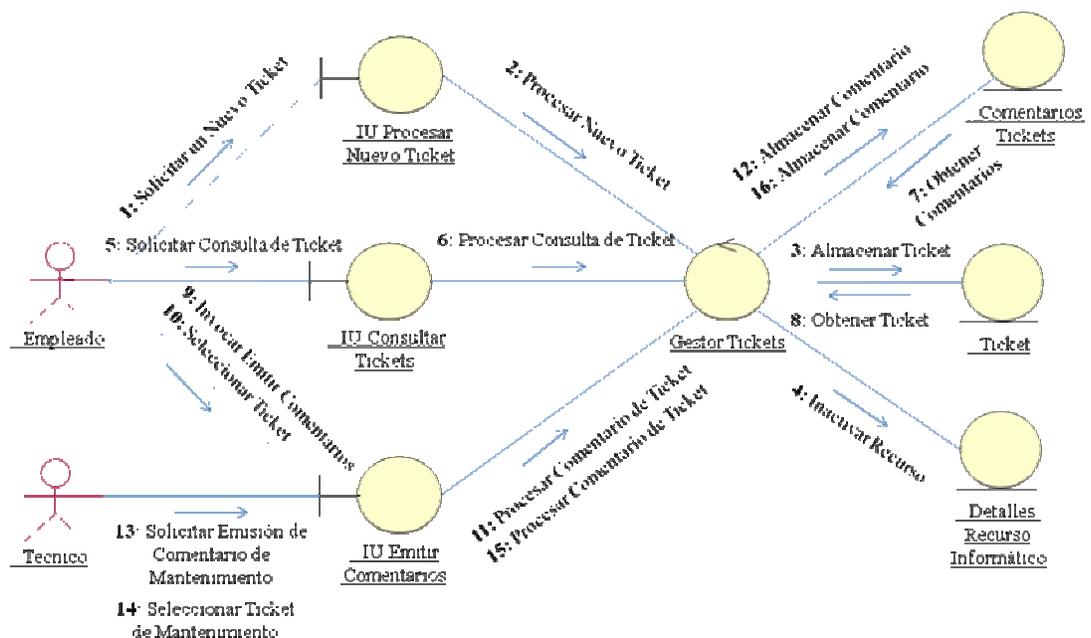


Figura 3.13. Diagrama de Colaboración para los Casos de Uso Procesar Nuevo Ticket, Consultar Ticket y Emitir Comentario Ticket de Respuesta de Solicitud

[Fuente: Elaboración Propia]

3.5.2. Análisis de la Arquitectura

El propósito del análisis de la arquitectura es esbozar el modelo de análisis y la arquitectura mediante la identificación de paquetes de análisis. La descripción de la arquitectura contiene una vista de la arquitectura del modelo de análisis, que muestra sus artefactos significativos para la arquitectura como la descomposición del modelo de análisis en paquetes de análisis y sus dependencias.

3.5.2.1. Identificación de los Paquetes de Análisis

Un paquete es una forma de agrupar clases en modelos grandes, pueden tener asociaciones de dependencias o de generalizaciones entre ellos. Los paquetes de análisis proporcionan un medio para organizar los artefactos del modelo de análisis en piezas manejables. En general, los paquetes de análisis deben ser cohesivos (fuertemente relacionados), y deben ser débilmente acoplados (con mínima dependencia).

Los paquetes de análisis pueden constar de clases de análisis, de realizaciones de casos de uso, y de otros paquetes de análisis. A continuación se presentan desde la Figura 3.14 hasta la Figura 3.20, cada uno de los paquetes del sistema con sus respectivas clases de análisis.



Figura 3.14. Paquetes de Análisis: Ticket

[Fuente: Elaboración Propia]



Figura 3.15. Paquetes de Análisis: Departamentos.

[Fuente: Elaboración Propia]



Figura 3.16. Paquetes de Análisis: Usuarios.

[Fuente: Elaboración Propia]



Figura 3.18. Paquetes de Análisis: Consultas DB.

[Fuente: Elaboración Propia]



Figura 3.19. Paquetes de Análisis: Inventario.

[Fuente: Elaboración Propia]



Figura 3.20. Paquetes de Análisis: Reportes Estadísticos.

[Fuente: Elaboración Propia]

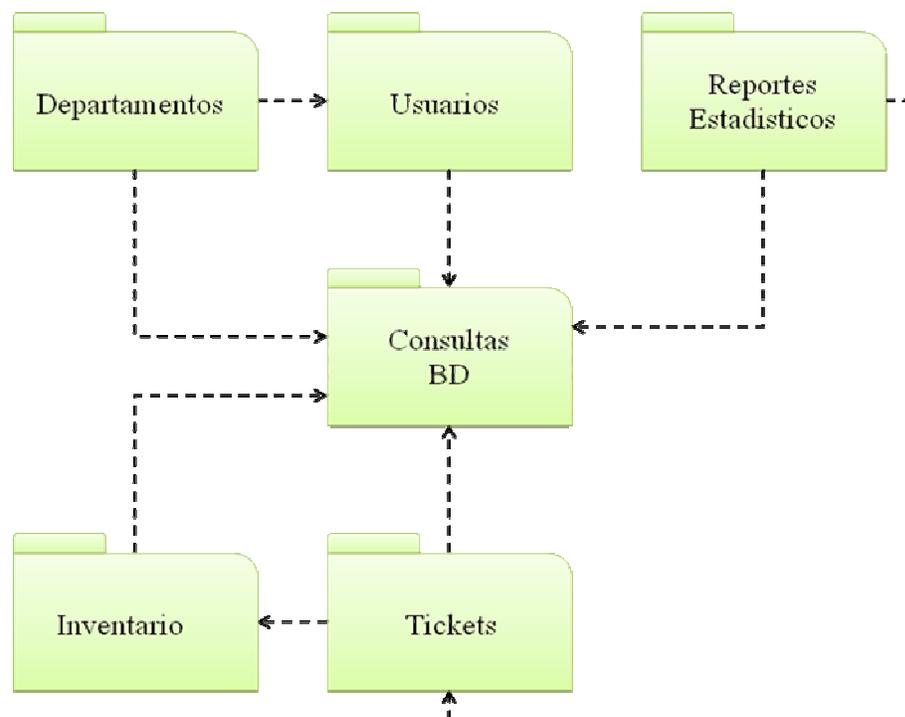


Figura 3.21. Diagrama de Paquetes de Análisis del Sistema SAIME.

[Fuente: Elaboración Propia]

La identificación de paquetes de análisis permite organizar el modelo de análisis en unidades que, básicamente, engloban los requerimientos funcionales del sistema definidos con anterioridad como casos de uso en el modelo de casos de uso.

Siguiendo las directrices del Proceso Unificado de Desarrollo de Software, se realizan una serie de actividades cuyos resultados constituyen la versión inicial del modelo de análisis (ver Figura 3.21), el cual contiene los paquetes Departamentos, Usuarios, Consultas DB, Reportes Estadísticos, Tickets, Inventario y Fallas Predeterminadas.

3.6. Evaluación de la Fase de Inicio

En este capítulo se llevó a cabo la fase de inicio, la cual es primera fase del Proceso Unificado de Desarrollo de Software, en donde se estableció el concepto inicial del sistema, creando un esquema general del mismo, se identificaron y describieron los requisitos de sistema bajo estudio mediante los llamados casos de uso, con el fin de elaborar un conjunto de modelos iniciales que permitieron capturar la semántica y comportamiento del sistema.

La fase de inicio es la que reviste más importancia en todo el proceso, dado que es en ella donde se efectuó el análisis del sistema propuesto, se describió el contexto del sistema, representado a través del modelo de dominio, lo que permitió identificar los riesgos críticos que pondrían en peligro el desarrollo del proyecto y proponer una arquitectura candidata factible. Los casos de uso identificados con el flujo de trabajo de los requisitos fueron estudiados en detalle con diagramas de colaboración en el flujo de análisis, para representar las interacciones entre los objetos. Además, se identificaron y describieron las clases de análisis, haciendo la refinación de los casos de uso, mediante el uso de diagramas de clases de análisis, y se construyó el diagrama de paquetes de análisis para encapsular las clases de análisis.

CAPÍTULO IV

FASE DE ELABORACIÓN

La fase de elaboración refleja una perspectiva del sistema completo, describe el diseño arquitectónico del sistema, así como también, el diseño y aprovisionamiento de los componentes, a partir del conjunto de requerimientos funcionales y no funcionales obtenidos en el capítulo anterior. Se describen puntos como las metas del diseño, relación de los requisitos obtenidos en fases anteriores con la arquitectura del sistema, identificación de subsistemas, descripción de las vistas arquitectónicas, diseño de la base de datos y diseño de la interfaz de usuario/sistema.

En esta fase se van a transformar y refinar los modelos de la fase de inicio en otra serie de modelos que irán perfilando una solución más cercana al mundo real.

En la iteración de requerimientos se deducirá una visión más refinada de los requisitos funcionales del sistema para completar el diagrama de casos de uso de la fase inicio y obtener así el diagrama de casos de uso final del sistema.

En la iteración de análisis se refinarán las funcionalidades a través de los diagramas de clase de análisis. Luego, en la iteración de diseño, se especificarán los diagramas de clases y paquetes, se desarrollará el diagrama de base de datos y el diseño de prototipos de las interfaces de usuario; asimismo, se incluirán diagramas de secuencias para las principales actividades.

En la Figura 4.1. se puede notar que en esta fase, se realiza más énfasis en el análisis y diseño del sistema.

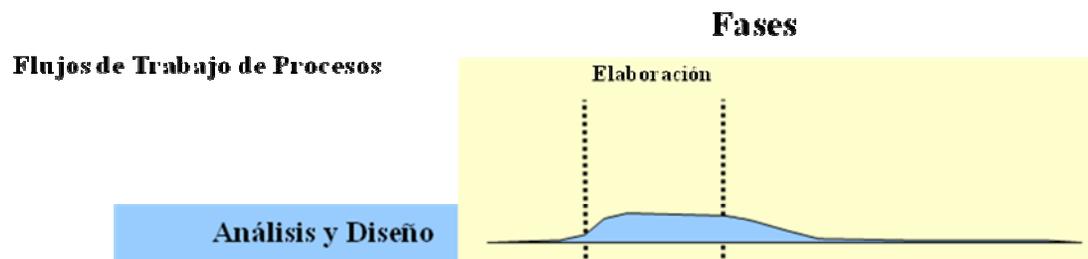


Figura 4.1. Esfuerzo en flujos de trabajo de la fase de elaboración.

[Fuente: Elaboración Propia]

Por último, se realizara en la iteración de implementación el diagrama de despliegue de la aplicación, para combinar la arquitectura de hardware y la arquitectura de software necesaria, así como también se desarrollara un diagrama parcial de componentes.

4.1. Requerimientos del Sistema

En esta nueva iteración de la fase de elaboración se han capturado nuevos requisitos para terminar de validar todas las funcionalidades del sistema.

4.1.1. Requisitos adicionales.

Generar reportes estadísticos, por ejemplo: solicitudes asignadas a miembros del staff técnico, solicitudes atendidas por mes, tiempos de respuestas, valoración de respuestas, entre otros.

Integrar la librería JFreeChart al lenguaje de programación JAVA como herramienta para la generación de gráficas estadísticas.

4.1.2 Interfaz de Inicio de SAIME

Esta interfaz es la que observaran los usuarios al momento de validar el sistema, es decir cuando el usuario ingrese la dirección de la aplicación se desplegara una ventana emergente que permite al usuario autenticarse utilizando su nombre de

usuario y contraseña como datos para verificar su existencia en la base de datos. En la Figura 4.2 se muestra un bosquejo de la interfaz tentativa.

Figura 4.2. Interfaz de Inicio (autenticación de usuario)

[Fuente: Elaboración Propia]

4.1.3. Interfaz de Usuario.

Las interfaces graficas son todos aquellos canales por los cuales se permite la comunicación entre el usuario y el computador. Son los que facilitan la comunicación y la interacción entre estos dos sistemas, usuario y máquina. Esto implica que la interfaz deba ser:

- Usable: para que los usuarios puedan conseguir los objetivos específicos con efectividad, eficiencia y satisfacción; con mayor facilidad posible
- Accesible: que los usuarios puedan ser capaz de usar la interfaz

Tal interfaz se compone de un grupo de paneles con elementos de formularios de Java Swing: caja de texto, botones de acción, tabla de datos, casilla de verificación y listas/menú.

La interfaz de usuario a diseñar está compuesta por los siguientes elementos:

- Ventana principal del sistema: muestra el nombre de la pantalla en uso
- Menú desplegables: muestra las opciones disponibles del sistema para un usuario determinado
- Botones de Acceso Rápido: muestra las opciones más comunes del sistema para un usuario determinado

- Identificación del usuario: identifica la sección y la unidad a la cual pertenece el usuario que está haciendo uso del sistema.
- Panel de Datos: muestra el formulario correspondiente a la pantalla en uso. Este panel puede contener distintos elementos: caja de texto, etiquetas, lista de opciones, casillas de verificación, etc.
- Botones de Acción: identifica todos los botones de acción que permiten manipular los datos mostrado en el formulario del panel.

En la Figura 4.3 se muestra un prototipo de interfaz de usuario utilizando todos los elementos antes descritos.



Figura 4.3. Prototipo de Interfaz de Usuario

[Fuente: Elaboración Propia]

4.2. Modelo de Casos de Uso Actualizado

En esta sección se presentan los casos de uso que conforman al sistema, incluyendo aquellos desarrollados durante la fase de inicio, sus modificaciones y el resto de los casos de uso específicos que completan los requisitos.

En este flujo de trabajo se identificaron nuevos casos de uso así como también un nuevo usuario que se adicionaran a los antes establecidos, con la finalidad de mejorar el entendimiento del contexto planteado en la fase de inicio. Estas adiciones permiten fijar directamente los lineamientos que se van a utilizar durante el desarrollo del proyecto.

4.2.1. Identificación de actor adicional.

Con respecto a la fase de inicio se incluyo un nuevo actor al sistema que vendrá a interactuar en el diagrama de caso de uso descrito anteriormente. En la Tabla 4.1 se presenta la descripción conceptual del actor

Tabla 4.1. Descripción de actor adicional.

ACTOR	DESCRIPCIÓN
Librería JFreeChart	Es un componente del lenguaje de programación encargado de proporcionar los métodos y clases necesarias para la generación de graficas estadísticas que manipulan la información contenida referente a los usuarios, departamentos, recursos informáticos y fallas relacionadas con el mantenimiento.

[Fuente: Elaboración Propia]

4.2.2. Identificación del Caso de Uso adicional

En la Tabla 4.2 se definen los nuevos casos de uso identificados en la fase de elaboración.

Tabla 4.2. Descripción de los Casos de Uso adicionales

CASO DE USO	DESCRIPCIÓN
Procesar Consultas Estadísticas	Permite recolectar y procesar información de la base de datos para generar gráficas estadísticas.
Consultar Estadísticas	Representa la visualización a través de gráficas estadísticas los mantenimientos procesados por departamento, por técnico, por fecha y también su tiempo de ejecución.

[Fuente: Elaboración Propia]

4.2.3. Descripción del Caso de Uso adicional

La descripción del caso de uso expuesto en la Tabla 4.3 muestra actor involucrado, las precondiciones y poscondiciones, los flujos básicos y los alternativos.

Tabla 4.3. Procesar Consultas Estadísticas. (1/2)

PROCESAR CONSULTAS ESTADISTICAS	
Actor Principal:	Librería JFreeChart
Personal Involucrado e intereses:	
Librería JFreeChart: recolecta información referente a las consultas realizadas, esto con la finalidad de poder presentar información específica al momento de generar un reporte gráfico	
Precondiciones:	El usuario debe realizar una consulta estadística.
Poscondiciones:	Generación de gráficos de barras y/o de torta.

[Fuente: Elaboración Propia]

Tabla 4.3. Procesar Consultas Estadísticas. (2/2)

Escenario principal de éxito (flujo básico)
A. El Sistema recolecta de la base de datos la información de las consultas realizadas por todos los usuarios.
B. Finaliza el caso de uso.
Extensiones (flujos alternativos)
No hay flujos alternativos

[Fuente: Elaboración Propia]

Tabla 4.4. Consultar Estadísticas. (1/2)

CONSULTAR ESTADISTICAS	
Actor Principal:	Administrador del Sistema
Personal Involucrado e intereses:	
Administrador del Sistema / Supervisor: Visualizar gráficamente los niveles de atención de tickets para cada uno de los días de un determinado mes, para cada uno de los departamentos así como también para cada uno de los miembros de soporte técnico.	
Precondiciones:	Ninguna
Poscondiciones:	Gráfico de barras y/o de torta donde se muestra el comportamiento de los niveles de atención de tickets para cada uno de los días del mes seleccionado, también para cada uno de los departamentos, así mismo, para cada unos de los miembros de soporte técnico.
Escenario principal de éxito (flujo básico)	
A. El usuario selecciona ver niveles de atención de tickets para cada uno de los días de un mes determinado.	
B. El usuario selecciona ver niveles de atención de tickets para cada uno de los	

[Fuente: Elaboración Propia]

Tabla 4.4. Consultar Estadísticas. (2/2)

departamentos.
C. El usuario selecciona ver niveles de atención de tickets para cada uno de los miembros de soporte técnico.
Extensiones (flujos alternativos)
No hay flujos alternativos

[Fuente: Elaboración Propia]

4.2.4. Diagrama de Casos de Uso Actualizado

En el diagrama de Casos de Uso representado en la Figura 4.4 se puede observar que los actores administrador del sistema y supervisor de mantenimiento a través del Caso de Uso Consultar Estadísticas pueden visualizar gráficamente los mantenimientos procesados, no procesados y los que están pendiente.

La librería JFreeChart representa al actor encargado de la ejecución del Caso de Uso Procesar Consultas Estadísticas, que a su vez se ejecuta al ser llamado el Caso de Uso Consultar Estadística.

4.3. Análisis

En la fase de inicio se realizó un análisis de la arquitectura del sistema pero sólo para determinar la factibilidad de la misma, a partir de este flujo de trabajo se analizarán más detalladamente los casos de uso del sistema, esto mediante la utilización de diversos diagramas que servirán para precisar y fundamentar la línea base de la arquitectura que se pondrá en ejecución.

Diagrama de Casos de Uso del SAIME

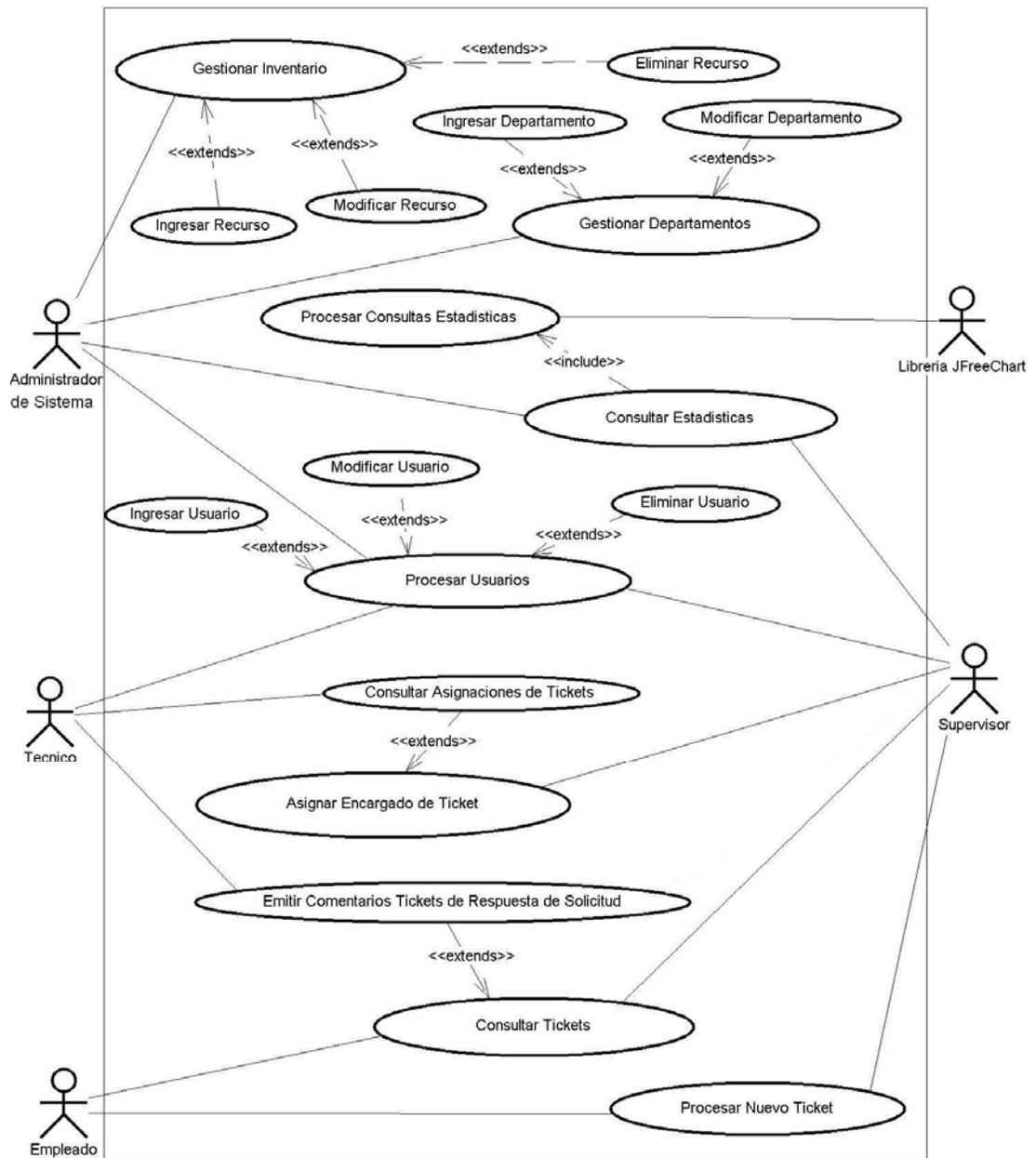


Figura 4.4. Diagrama de Casos de Uso General de SAIME
[Fuente: Elaboración Propia]

4.3.1. Identificación de las Clases de Análisis.

El análisis de los requisitos capturados en la fase de elaboración conlleva a la construcción de un nuevo modelo de análisis del sistema, a partir del caso de uso Consultar Estadísticas se identifican las nuevas clases de análisis que serán descritas en la Tabla 4.5.

Tabla 4.5. Identificación de las Clases de Análisis para el Caso de Uso Consultar Estadísticas.

CLASE DE ENTIDAD	DESCRIPCIÓN
IU Consultar Estadísticas	Permite al administrador del sistema generar graficas estadísticas sobre la información almacenada relacionada con los mantenimiento
Gestor Consultar Tickets	Procesa la visualización de la información correspondiente a los tickets de mantenimiento.
Gestor Consultar Usuarios	Procesa la visualización de estado los usuarios en general del sistema.
Gestor Consultar Departamentos	Procesa la visualización del área de trabajo donde opera un determinado recurso.

[Fuente: Elaboración Propia]

4.3.2. Diagrama de las Clases de Análisis.

En el diagrama de clase de análisis del caso de uso Consultar Estadísticas (ver Figura 4.5) podemos observar que tan solo el administrador y el supervisor tienen el permiso necesario para visualizar graficas estadísticas en el sistema, solicitando acceso a través de la interfaz de usuario Consultar Estadísticas.

Las consultas que podrán ser solicitadas están asociadas a las entidades tickets, usuario y departamentos las cuales fueron definidas en la fase de inicio (ver Tabla 3.16).

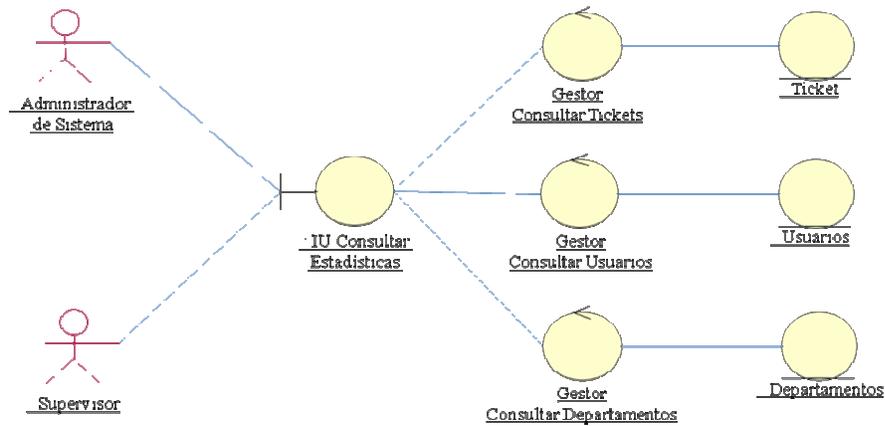


Figura 4.5. Diagrama de Clases de Análisis para el Caso de Uso Consultar Estadísticas
[Fuente: Elaboración Propia]

4.3.3. Diagrama de Colaboración.

En las Figura 4.6 y Figura 4.7, se muestra las interacciones del diagrama de clase de análisis anterior. El diagrama de colaboración exhibe mensajes que describen el propósito de los objetos contenido en el, en donde el administrador de sistema y el supervisor poseen el mismo flujo para el objetivo final que es el de Consultar Estadísticas.

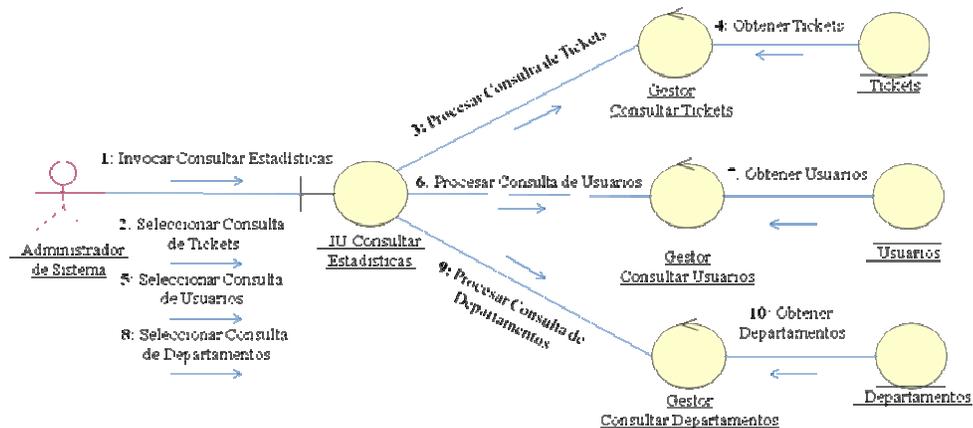


Figura 4.6. Diagrama de Colaboración para el Caso de Uso Consultar Estadísticas
(actor Administrador de Sistema)
[Fuente: Elaboración Propia]

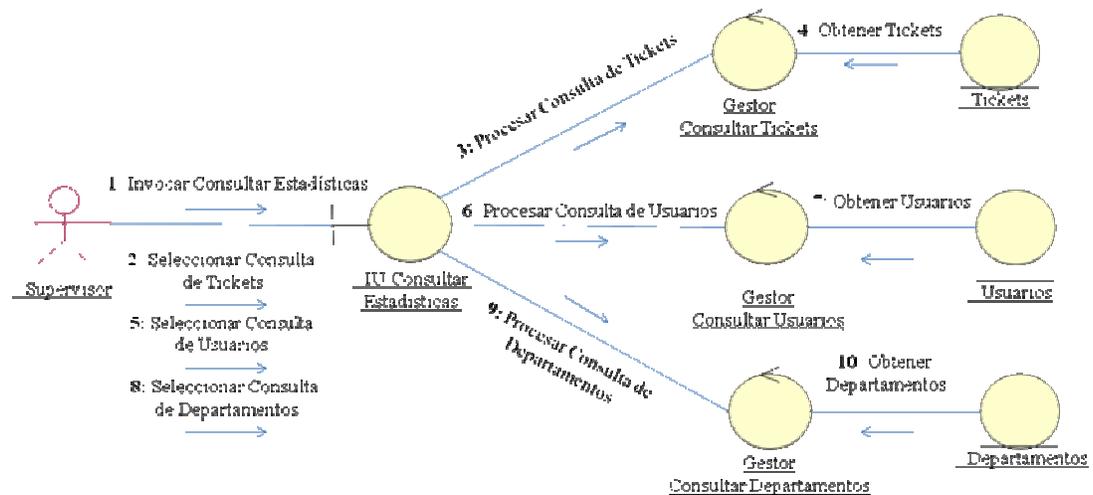


Figura 4.7. Diagrama de Colaboración para el Caso de Uso Consultar Estadísticas
(actor Supervisor)

[Fuente: Elaboración Propia]

4.4. Diseño

Una entrada esencial en el diseño es el resultado del análisis, esto es, el modelo de análisis. El modelo de análisis proporciona una comprensión detallada de los requisitos que impone una estructura del sistema, por lo tanto esta se debe conservar lo más fielmente posible cuando se da forma al sistema.

El flujo de diseño en la fase de elaboración tiene como objetivo la obtención de la vista de la arquitectura del Modelo de Diseño, es decir, la realización física de los casos de uso.

4.4.1. Diagrama de Clase de Diseño.

Para modelar los aspectos estáticos de un sistema se hace uso del diagrama de clases, el cual es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases

son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro.

El diagrama de clases es el principal diagrama de diseño para un sistema. Los diagramas de clases de diseño muestran la funcionalidad del sistema mediante clases relacionadas entre sí por medio de líneas, haciendo la especificación de las instancias que participan en la relación.

Las clases de diseño definen los atributos y operaciones con que debe contar cada clase de análisis para llevar a cabo sus responsabilidades, así como las relaciones existentes entre ellas. Los atributos y operaciones se encuentran en su mayoría inmersos dentro de los diagramas y artefactos obtenidos en los diferentes flujos de trabajo, por lo que se necesita realizar una revisión exhaustiva de estos y agrupar las distintas responsabilidades de las clases para ofrecer soporte a todos sus roles. Los diagramas de clases muestran la estructura estática de los casos de uso.

Una vez obtenido el resultado del flujo de análisis en la fase de inicio, en la Figura 4.8 se ilustra el nivel de abstracción del sistema representado a través del Diagrama de clases de Diseño del Sistema SAIME, definiendo las clases presentes en el sistema. La clase IU SAIME Principal representa toda la aplicación como un conjunto, la clase IU Login es agregada a la clase IU SAIME Principal, y constituye una ventana de interfaz para la validación de usuario. La clase IU Departamentos es agregada a la clase IU SAIME Principal, y constituye una ventana de interfaz de usuario que se usa para la gestión de Departamentos de la Empresa. La clase Departamentos es agregada de la clase IU Departamentos, que implican el procesamiento de los parámetros para realizar las acciones de insertar, modificar y eliminar Departamentos en la Base de Datos.

La clase IU Usuarios es agregada a la clase IU SAIME Principal, y constituye una ventana de interfaz de usuario que se usa para realizar el ingreso de la data de los usuarios que hacen uso del sistema, mediante la clase Usuarios, que es agregado a la clase IU Usuarios. La clase IU Asignaciones es agregada a la clase IU SAIME

Principal, y constituye una ventana de interfaz de usuario que se usa para hacer asignaciones de ticket a los técnicos encargados de los mantenimientos de equipos. La clase Ticket y la clase Usuario son agregadas a IU Asignaciones. La clase Ticket también es agregada a las clases IU Nuevo Ticket, IU Consultar Ticket y IU Observaciones, para representar el ingreso, visualización y comentarios de tickets respectivamente. La clase RecursoInformatico es agregada a la clase IU Inventario, y representa el ingreso, modificación y eliminación de recursos informáticos del Inventario de la empresa.

La clase IU Estadísticas es agregada a la clase IU SAIME Principal, y constituye una ventana de interfaz de usuario que se usa obtener las graficas estadísticas del desempeño de la programación de mantenimiento.

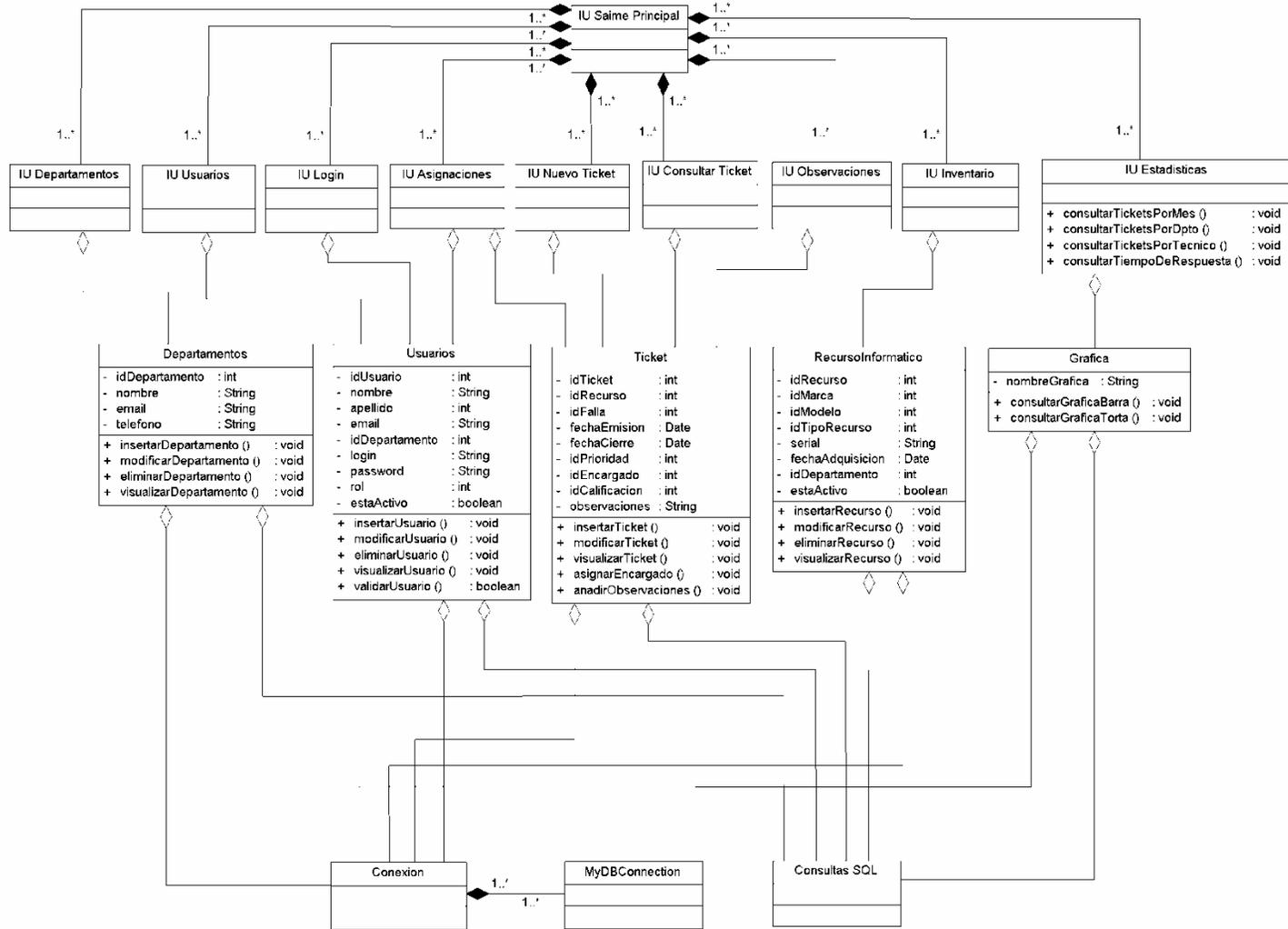


Figura 4.8. Diagrama de Clases de Diseño del SAIME [Fuente: Elaboración Propia]

4.4.2. Diagramas de Secuencia.

Los Diagramas de Secuencia son conocidos como Diagramas de Interacción dado que muestran la interacción que se establece en un conjunto de objetos y sus relaciones, además de los mensajes que estos se envían. Es decir, tratan la vista dinámica de un sistema. Estos diagramas enfatizan como se realiza un caso de uso en términos de las clases del diseño, identifica las instancias de los actores, las interacciones entre los objetos del diseño y los mensajes que se envían y reciben estos objetos. Aquí lo importante es encontrar secuencias de interacciones ordenadas en el tiempo. A continuación se muestran los Diagramas de Secuencia de los Casos de Uso más importantes del sistema SAIME.

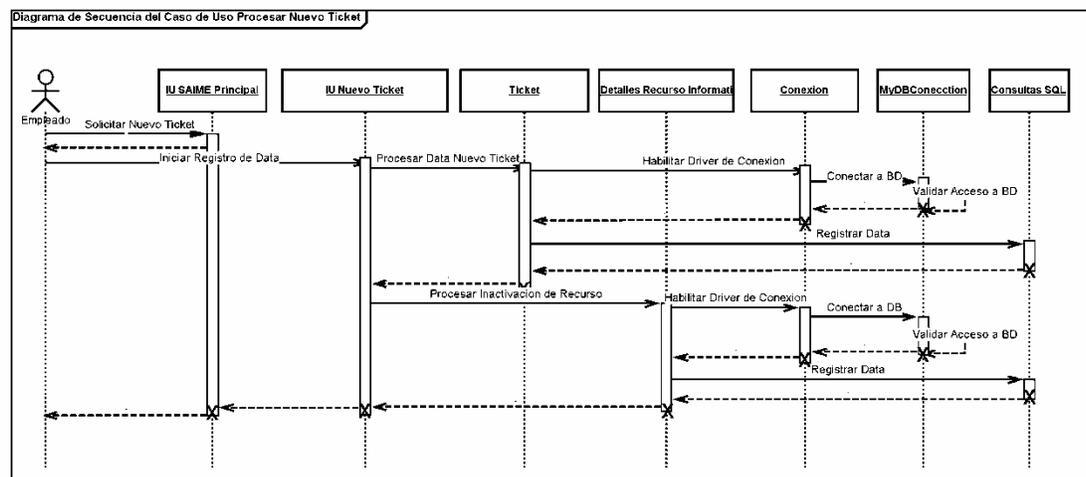


Figura 4.9. Diagrama de Clases de Secuencia del Caso de Uso Procesar Nuevo Ticket

[Fuente: Elaboración Propia]

En la Figura 4.9 se despliega el diagrama de secuencia para el caso de uso procesar nuevo ticket, en el cual el actor solicita un nuevo ticket, para esto el usuario envía una solicitud a una interfaz principal la cual envía los datos a la interfaz adecuada para el actor, encargada de transmitir los datos a la clase correspondiente a la gestión de ticket y finalmente transmitir a la clases de conexión y consultas para realizar la operación seleccionada en la base de datos. Una vez registrada

exitosamente la data correspondiente al nuevo ticket automáticamente el sistema procede a cambiar el estatus del recurso informático a inactivo mediante el mismo procedimiento a través de la clase conexión y consultas para poder operar con la base de datos.

La Figura 4.10 muestra el diagrama de secuencia para el caso de uso gestionar inventario, en el cual el actor solicita alguna de las operaciones disponibles en relación a los recursos informáticos como son, ingresar, modificar o eliminar información, para esto el usuario envía una solicitud a una interfaz principal la cual envía los datos a la interfaz adecuada para el actor, encargada de transmitir los datos a la clase correspondiente a la gestión de señales según el usuario y finalmente transmitir a la clases de conexión y consultas para realizar la operación seleccionada en la base de datos.

4.4.3. Diagrama de Capas.

El uso de la arquitectura de capas es aplicable a muchos tipos de sistemas. Este patrón define como organizar el modelo de diseño en capas, lo cual quiere decir que los componentes de una capa solo pueden hacer referencia a componentes en capas inmediatamente inferiores. Este patrón es importante porque simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores. Además, nos ayuda a identificar que puede reutilizarse, y proporciona una estructura que nos ayuda a tomar decisiones sobre que partes comprar y que partes construir. Un sistema con una arquitectura en capas pone a los subsistemas de aplicación individuales en lo más alto. Estos se construyen a partir de subsistemas en las capas más bajas, como son los marcos de trabajo y las bibliotecas de clases.

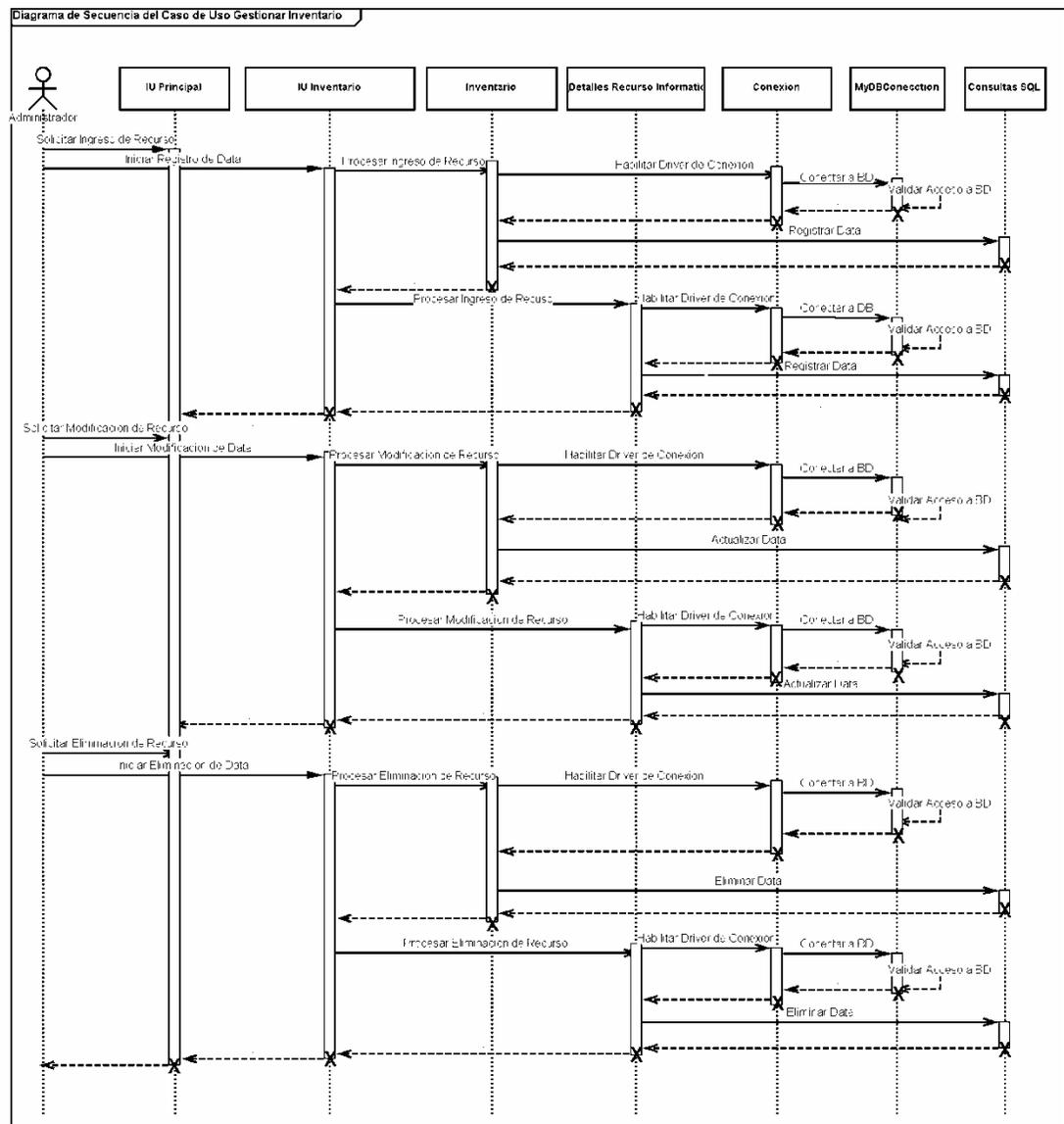


Figura 4.10. Diagrama de Clases de Secuencia del Caso de Uso Gestionar Inventario

[Fuente: Elaboración Propia]

La capa general de aplicación contiene los subsistemas que no son específicos de una sola aplicación, sino que pueden ser reutilizados por muchas aplicaciones diferentes dentro del mismo dominio o negocio. La arquitectura de las dos capas inferiores puede establecerse sin considerar los casos de uso de que no son dependientes del negocio. La arquitectura de las dos capas superiores se crea a partir

de los casos se uso significativos para la arquitectura (estas capas son dependientes del negocio). Una capa es un conjunto de subsistemas que comparten el mismo grado de generalidad y de volatilidad en las interfaces: las capas inferiores son de aplicación general a varias aplicaciones y deben poseer interfaces más estables, mientras que las capas más altas son más dependientes de la aplicación y pueden tener interfaces menos estables. La Figura 4.11 nos muestra la arquitectura del sistema SAIME utilizando un diagrama de capas.

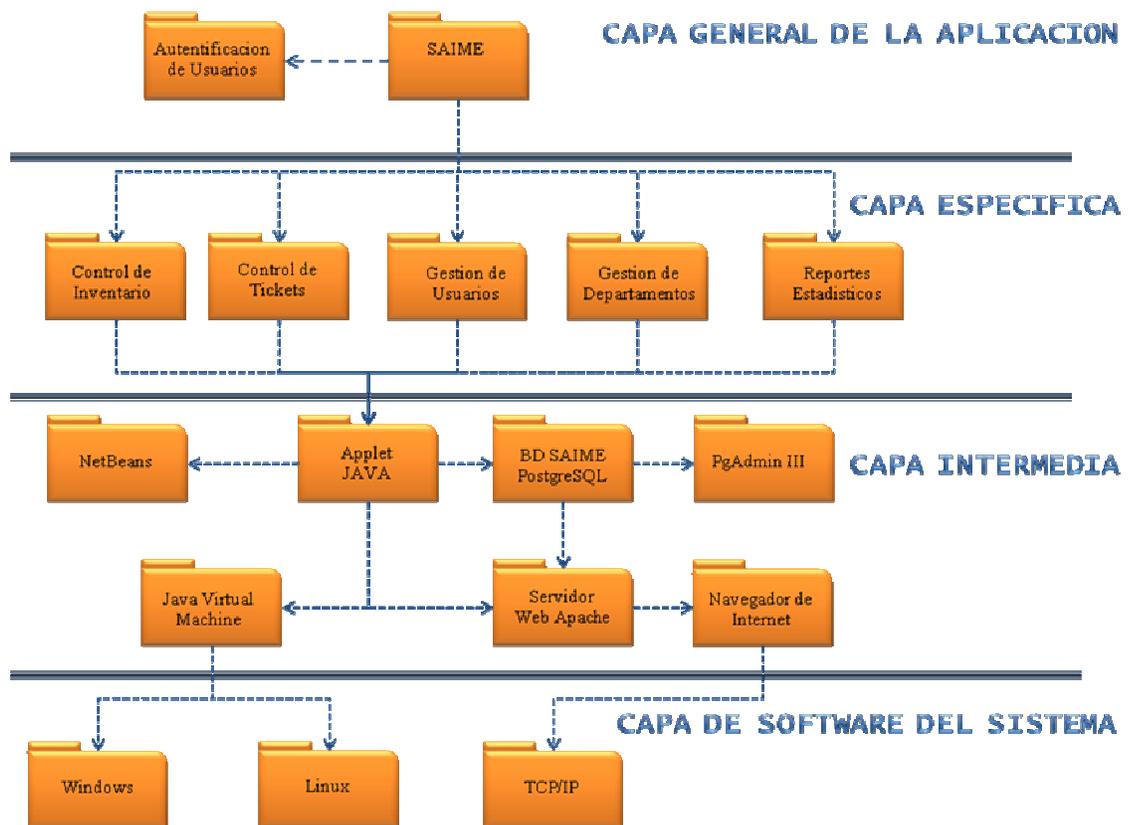


Figura 4.11. Diagrama de Capas

[Fuente: Elaboración Propia]

4.4.4. Diagrama de Base de Datos.

Para el diseño de la base de datos se han tomado como referencia los diagramas de análisis y los diagramas de clases, como punto de partida para el diseño de las

tablas y el manejo de la información requerida por el sistema, así como la relación existente entre dichas tablas.

Cabe destacar que el buen diseño de una base de datos permitirá entender las entidades de manera sencilla, así como realizar consultas y reportes desde diferentes puntos de vista, de una manera rápida y eficiente.

Al tener todas la tablas que representan todas las entidades que utilizará el sistema, se procede a realizar las relaciones exigiendo la integridad referencial y la actualización en cascada de los campos relacionados, en la Figura 4.12 se puede observar la relación que existe entre las tablas que conforman la base de datos del sistema SAIME, mediante alguno de sus atributos.

La tabla Recursos Informáticos representa al inventario del cual se alimenta el sistema, los recursos informáticos se relacionan con los tickets en caso de que sea necesario realizarle un mantenimiento, además se relacionan con los departamentos para especificar el lugar dentro de la empresa donde se ubica. Las tablas Computador, Monitor, Ratón, Regulador, Teclado, Impresora y Software se encargan de darle características distintiva a una tupla particular de la tabla de Recursos Informáticos.

Las tablas Marca Recurso, modelo Recurso, Tipo Conector, Tipo Recurso y Calificación son tablas empleadas para la normalización de datos.

4.5. Implementación

En la implementación se comenzará con el uso de los resultados del diseño y se implementará el sistema en términos de componentes.

Modelo Entidad-Relacion del SAIME

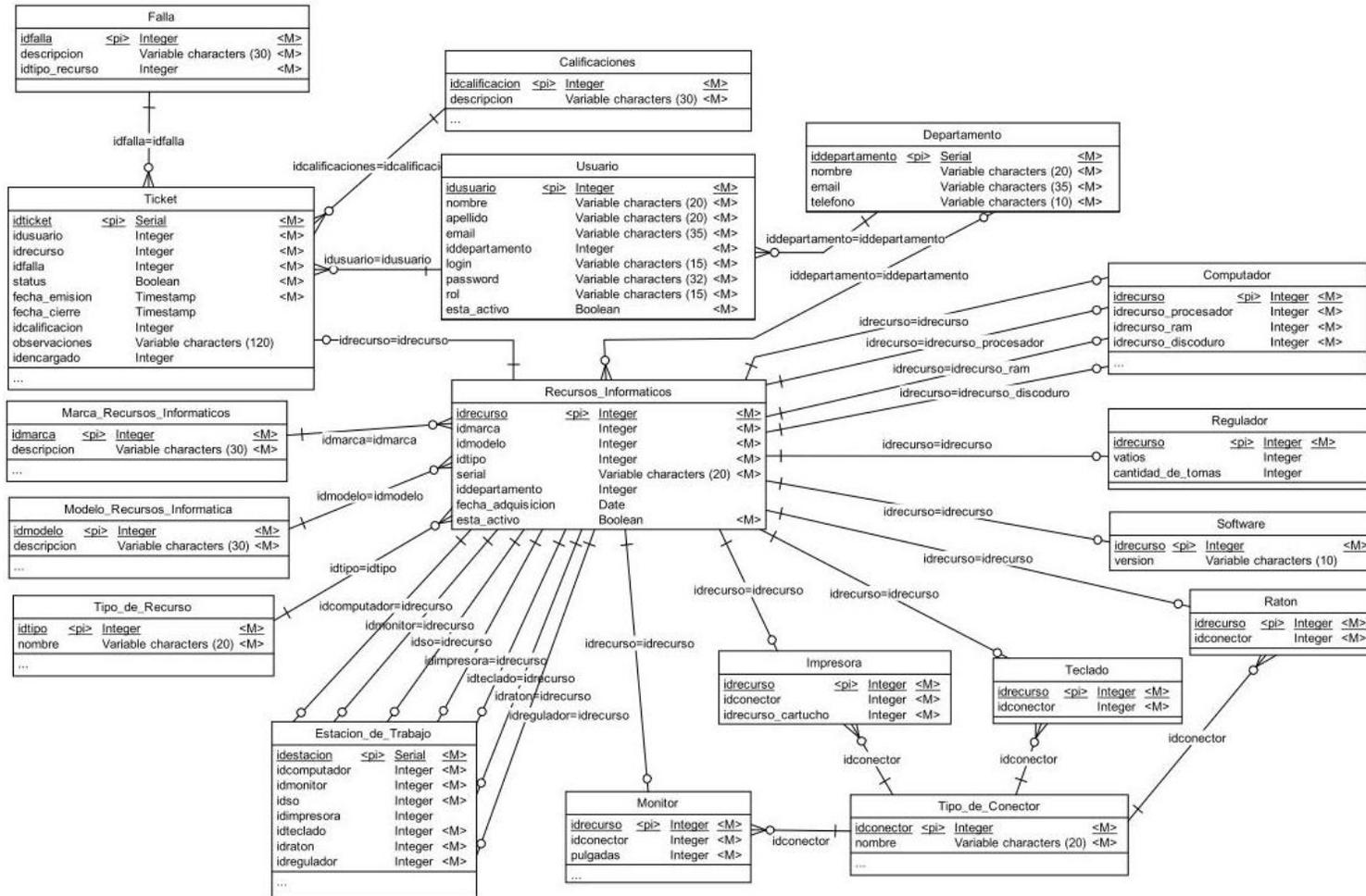


Figura 4.12. Modelo Entidad – Relacion del SAIME [Fuente: Elaboración Propia]

El propósito principal de este flujo de trabajo es desarrollar la arquitectura y el sistema como un todo. Aquí se realizará el diagrama de componentes parcial de la aplicación y se implementará la interfaz de bienvenida a la aplicación y validación de usuarios, debido a que el resto se verán completamente implementadas en la fase de Construcción.

4.5.1 Diagrama de Componentes:

Para la fase de elaboración se ha construido un adelanto de los componentes que conforman el sistema con sus relaciones. En la Figura 4.13 observamos los componentes relacionados a la solicitud de un nuevo ticket por parte del usuario general.

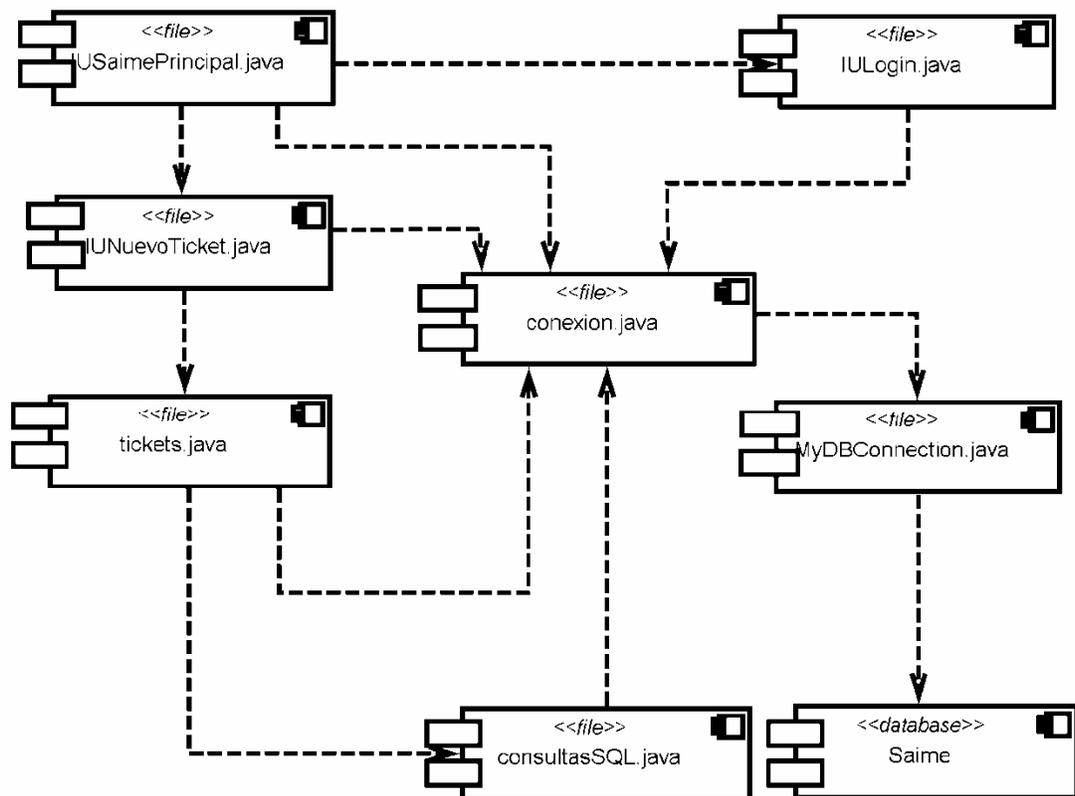


Figura 4.13. Diagrama parcial de Componentes (CU Procesar Nuevo Ticket)

[Fuente: Elaboración Propia]

4.5.2. Implementación de los componentes asociados a la solicitud de un nuevo ticket:

En la Figura 4.14 observamos la interfaz para el usuario principal, con esta estamos observando la ejecución de los componentes IU Saime Principal, IU Login y IU Nuevo Ticket, del diagrama de la Figura 4.13

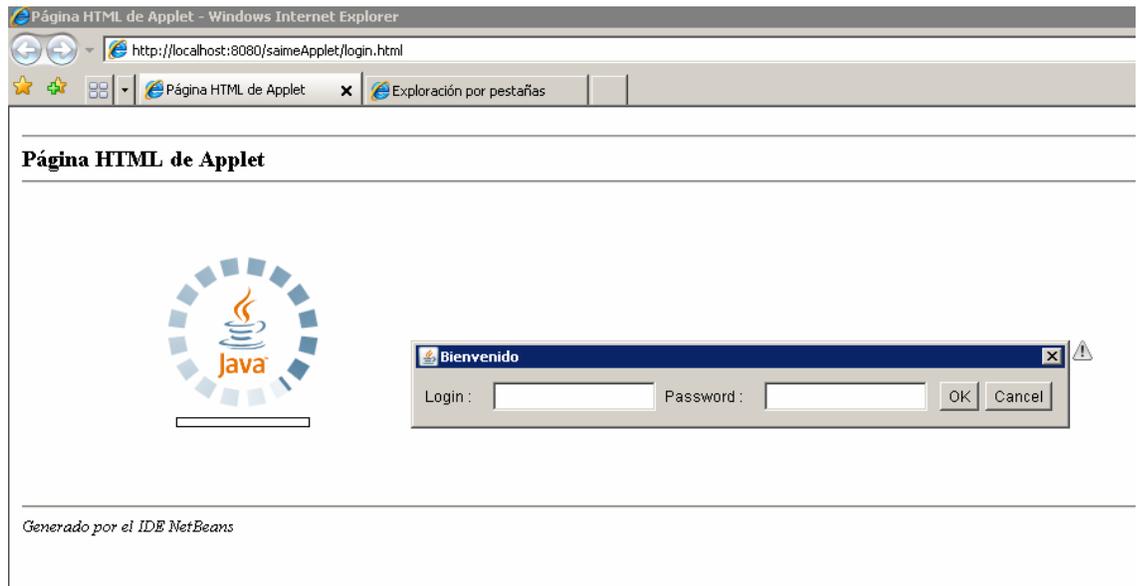


Figura 4.14. Interfaz de inicio del Sistema (autenticación de usuario)

[Fuente: Elaboración Propia]

Código (función `init()` del Applet)

```
public void init() {
    lista_dptos = new ArrayList<departamentos>();
    lista_users = new ArrayList<usuarios>();
    username_temp = null;
    try {
        Objcon= new conexion();
        Objcon.conectarBD("SIGMA");
    }catch (Exception e) {e.printStackTrace(); }
```

```

// Validación del nombre de usuario y la clave
if (!login()) {
    try {
        getAppletContext().showDocument
            (new java.net.URL(getCodeBase()+"accessdenied.html"), "_top");
    } catch (Exception e) {e.printStackTrace(); }
}
else{
    try {
        java.awt.EventQueue.invokeLaterAndWait(new Runnable() {
            public void run() {
// Declaracion de todos los componentes
                initComponents();
// Habilitacion de las opciones segun el tipo de usuario
                if(getTipoUsuario().equals("Cliente"))
                    habilitarCliente();
                else
                    if(getTipoUsuario().equals("Supervisor"))
                        habilitarSupervisor();
                    else
                        if(getTipoUsuario().equals("Tecnico"))
                            habilitarTecnico();
                        else
                            habilitarAdministrador();
                }
            });
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}

```

```

}
}

```

Para la Figura 4.15 observamos la interfaz Nuevo Ticket, que es la interfaz por defecto para usuarios de tipo empleado.

Figura 4.15. Interfaz de Usuario Nuevo Ticket

[Fuente: Elaboración Propia]

Código (crearTicket)

```

public boolean NuevoTicket(){
    boolean listo = true;
    String user = username_temp==null? getCedula(): getCedula_Temp();
    String serial = SerialRecurso2.getText();
    int codfalla = getCodFalla(TipoFalla2.getSelectedItem().toString());
    String prioridad = PrioridadFalla2.getSelectedItem().toString();
    long time = new java.util.Date().getTime();
    String fecha = new java.sql.Timestamp(time).toString();

```

```

String nota = NotaNuevoTicket.getText().length()<5?
    null : NotaNuevoTicket.getText();
String status ="Abierto";
String sql;
try{
    sql = "insert into tickets values (NEXTVAL('tickets_idticket_seq'),"
        +user+"", ""+serial+"", ""+codfalla+"", ""+prioridad+"", ""+
        status+"", ""+fecha+"")";
//    Objcon.stmt.execute(sql);
} catch(Exception e){ listo = false; nota = null;
    javax.swing.JOptionPane.showMessageDialog(null,e.toString(),
        "ERROR insertar tickets",
        javax.swing.JOptionPane.ERROR_MESSAGE);
    }

if(nota!=null)
{ try{
    sql = "insert INTO notas_tickets VALUES (" +
        "(SELECT last_value from tickets_idticket_seq), ""+nota+"")";
//    Objcon.stmt.execute(sql);
} catch(Exception e){ listo = false;
    javax.swing.JOptionPane.showMessageDialog(null,e.toString(),
        "ERROR insertar nota in nota_tickets",
        javax.swing.JOptionPane.ERROR_MESSAGE);
    }
}
return listo;
}

```

Por ultimo, observamos la Figura 4.16 que corresponde a la ventana de confirmación de entrada de datos asociado al evento del botón Enviar de la IU Nuevo Ticket. En esta ventana se verifica que los datos estén correctos para poder almacenar el ticket a la base de datos.

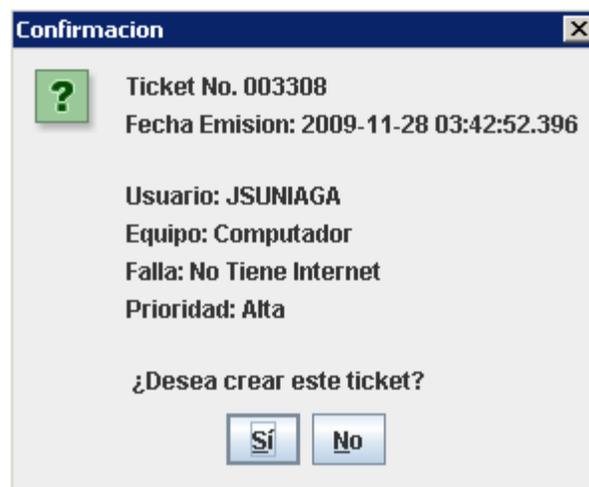


Figura 4.16. Ventana de confirmación para el almacenamiento del Ticket
[Fuente: Elaboración Propia]

Código(evento botón enviar):

```

Private void
EnviarNuevoTicketActionPerformed(java.awt.event.ActionEvent evt) {
    if(RecursosCargo2.getSelectedIndex()==0 ||
        TipoFalla2.getSelectedIndex()==0 ||
        (OtraFalla2.isEnabled() && OtraFalla2.getText().isEmpty())){
        javax.swing.JOptionPane.showMessageDialog
            (null, "Hay campos vacios", "intente de nuevo",
            javax.swing.JOptionPane.ERROR_MESSAGE);
    }

    else{ long time = new java.util.Date().getTime();

```

```
String fecha = new java.sql.Timestamp(time).toString();
String usuario = username_temp==null? getCedula() :
getCedula_Temp();
```

```
String salida =
    "Ticket No. " + "<numero_ticket>" + "\n" +
    "Fecha Emision: " + fecha + "\n\n"+
    "Usuario: " + usuario + "\n"+
    "Equipo: " + RecursosCargo2.getSelectedItem().toString()
        + "\n"+
    "Falla: " + TipoFalla2.getSelectedItem().toString()
        + "\n" +
    "Prioridad: "+PrioridadFalla2.getSelectedItem().toString()
        + "\n\n " +
    "¿Desea crear este ticket?";
int opcion = javax.swing.JOptionPane.showConfirmDialog
    (null,salida,"Confirmacion",
        javax.swing.JOptionPane.YES_NO_OPTION);
if(opcion == javax.swing.JOptionPane.YES_OPTION){
    if(insertarTicket()){
        javax.swing.JOptionPane.showMessageDialog
            (null, "Solicitud Exitosa", "Ticket No. ####",
                javax.swing.JOptionPane.INFORMATION_MESSAGE);}
    else{ javax.swing.JOptionPane.showMessageDialog
        (null, "No se puede almacenar los datos",
            "Solicitud Fallida",
                javax.swing.JOptionPane.ERROR_MESSAGE);}
    }
}
```

}

4.6. Evaluación de la Fase de Elaboración

Al comienzo de la fase de elaboración, se recibió de la fase de inicio la base para su desarrollo, un modelo de casos de uso completo y una descripción de la arquitectura candidata. Durante el desarrollo de los flujos de trabajo se obtuvieron las diferentes vistas de la arquitectura del sistema, estableciendo la prioridad de los casos de uso, la línea base de la arquitectura soporta los casos de uso críticos del sistema, ya que los riesgos críticos fueron tratados a través de los casos de uso, siendo mitigados al implementar los casos de uso correspondientes. El objetivo principal de esta fase, fue llevar a cabo el diseño y estructura del sistema para dar soporte a todos los requisitos que se obtuvieron durante la fase de inicio.

Durante esta fase se diseñaron las interfaces de usuario, lo que permitió obtener el medio de comunicación entre el usuario y el sistema, de igual manera se presentó el diseño de los reportes emitidos por el sistema, una vez hecha la solicitud del usuario.

CAPÍTULO V

FASE DE CONSTRUCCIÓN

La finalidad principal de la fase de construcción es alcanzar la capacidad operacional del producto. Durante esta fase todos los componentes, características y requisitos identificados y detallados en las fases anteriores deben ser implementados, integrados y probados en su totalidad, obteniendo una versión aceptable del software.

En esta fase se hace hincapié en las iteraciones de implementación y prueba del flujo de trabajo normal (ver Figura 5.1), debido a que su meta es lograr el desarrollo del sistema con calidad de producción, mediante la implementación de toda la funcionalidad y realización de las pruebas.

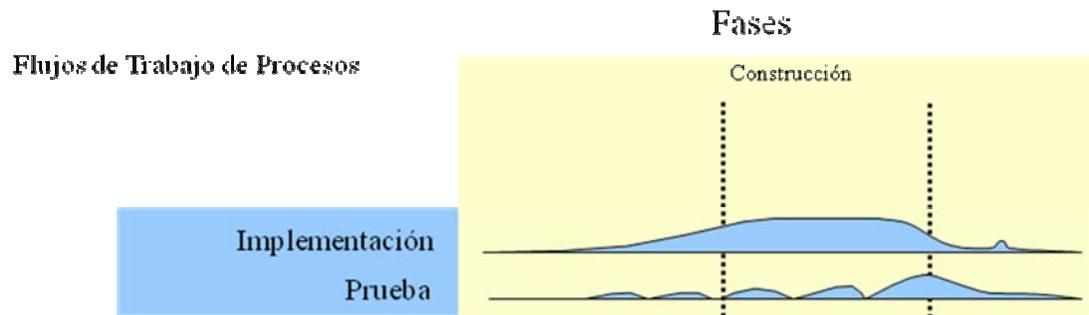


Figura 5.1. Esfuerzo en flujos de trabajo de la fase de construcción.

[Fuente: Elaboración Propia]

5.1 Implementación

En este flujo de trabajo se implementan las clases y objetos en ficheros fuente, binarios, ejecutables y demás. La implementación trata al sistema en términos de desarrollo de componentes y codificación del software, su relación con la base de datos, integración de módulos y la explicación acerca de las funcionalidades del sistema y su correcto uso, delatando así toda la estructura en forma de código abierto e identificación de las actividades que este puede realizar.

5.1.1. Escogencia del Lenguaje de Programación:

Para el desarrollo del SAIME, se debe contar con un lenguaje que soporte la programación orientada a objetos, ya que esta promueve una mejor comprensión de los requisitos, diseños más limpios y sistemas más fáciles de mantener.

Es por esto que se escogió como lenguaje de programación JAVA, debido a que es un lenguaje simple, seguro y multiplataforma que puede ser usado indistintamente bajo cualquier sistema operativo sin necesidad de hacer cambios en la programación.

El desarrollo Web estará fundamentado en la programación de un applet con la finalidad de crear un “subprograma” que pueda ser incrustado dentro de código HTML. La ventaja principal de la programación en applet es liberar al servidor de la ejecución del programa, la cual se lleva a cabo por parte del cliente.

Como entorno de desarrollo se escogió el NetBeans. Algunas de las características que hacen interesante la elección de NetBeans IDE como plataforma para nuestra aplicación son las siguientes:

- Los proyectos desarrollados no dejan de ser multiplataforma, y poseen largadores (launchers) para cada plataforma.
- Sistema de ventanas práctico para desarrollar las interfaces de usuario
- Sistema de ficheros virtual en el cual se van montan los diferentes módulos con el cual se van adaptando automáticamente los menús, barra de herramientas, menús contextuales, etc. de la aplicación
- Su licencia nos permite construir tanto aplicaciones open source como comerciales
- Compatibilidad con Java Web Start
- No es obligatorio que una aplicación deba tener interfaz de usuario grafica (GUI), ya que la plataforma permite dejar de lado la misma y seguir disfrutando del resto de los beneficio, por ejemplo la actualización de módulos desde un repositorio remoto.

- Soporte completo para desarrollar desde NetBeans IDE, por lo que no necesitaremos otra herramienta adicional para el desarrollo

En la Figura 5.2 observamos la interfaz del software NetBeans, a través del cual se realizó la programación del applet en el lenguaje JAVA.

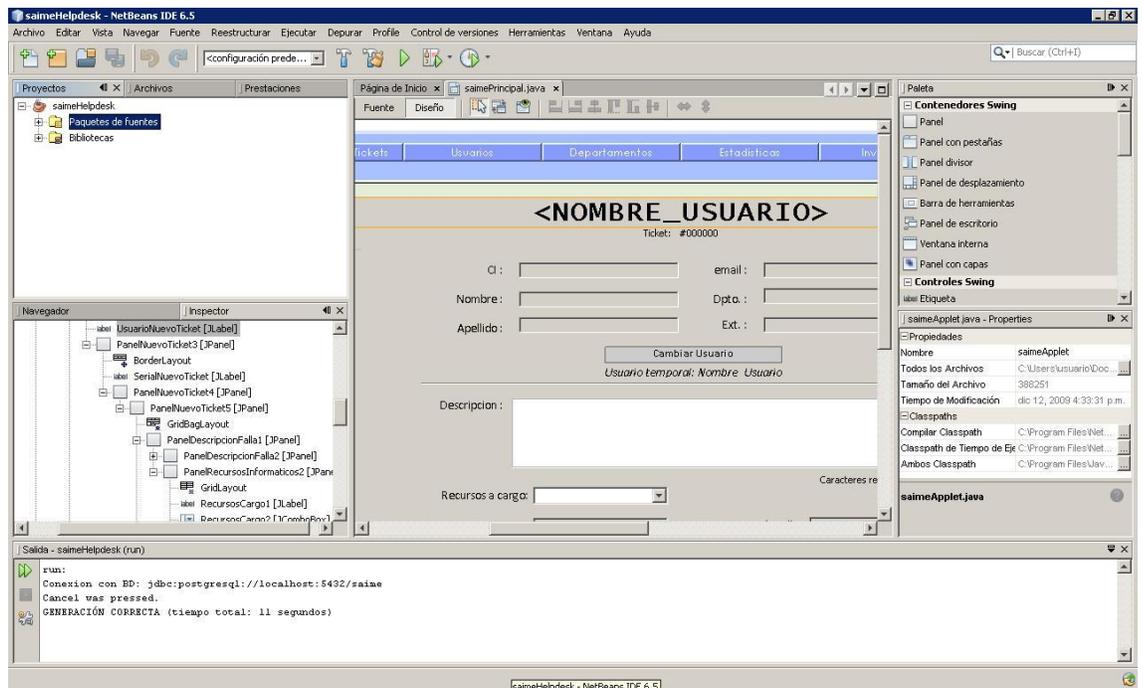


Figura 5.2. Interfaz del entorno de desarrollo JAVA

[Fuente: Elaboración Propia]

5.1.2. Escogencia del Gestor de Base de Datos:

Para la construcción de la base de datos de SAIME, se escogió PostgreSQL, es un sistema de gestión de base de datos relacional, multihilo y multiusuario, además, es altamente confiable en cuanto a estabilidad se refiere. El cliente gráfico de PostgreSQL es el pgAdmin III en el cual podemos ver y trabajar con casi todos los objetos de la base de datos, examinar sus propiedades y realizar tareas administrativas. En la Figura 5.3 se observa el entorno por medio del cual se construyeron las tablas que constituyen la base de datos SAIME.

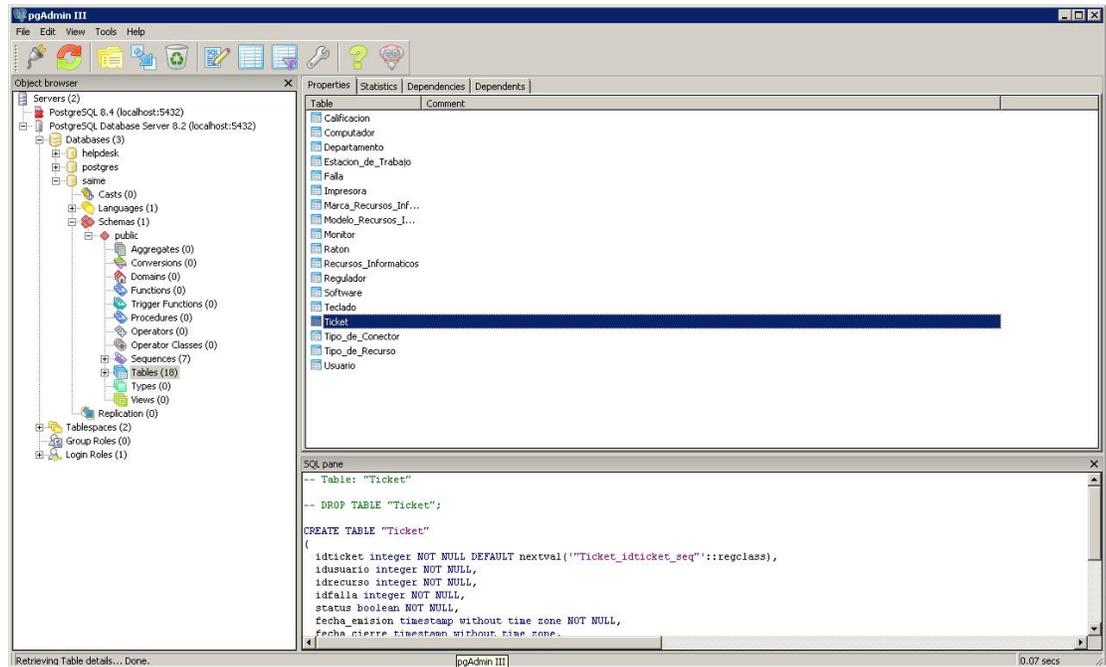


Figura 5.3. Interfaz del entorno grafico de PostgreSQL

[Fuente: Elaboración Propia]

5.1.3. Diagrama de Componentes Total:

En esta fase se desarrolla el diagrama de componentes con la totalidad de los componentes del sistema, indicando las clases que necesitan cada uno. En la Figura 5.4 podemos observar este diagrama.

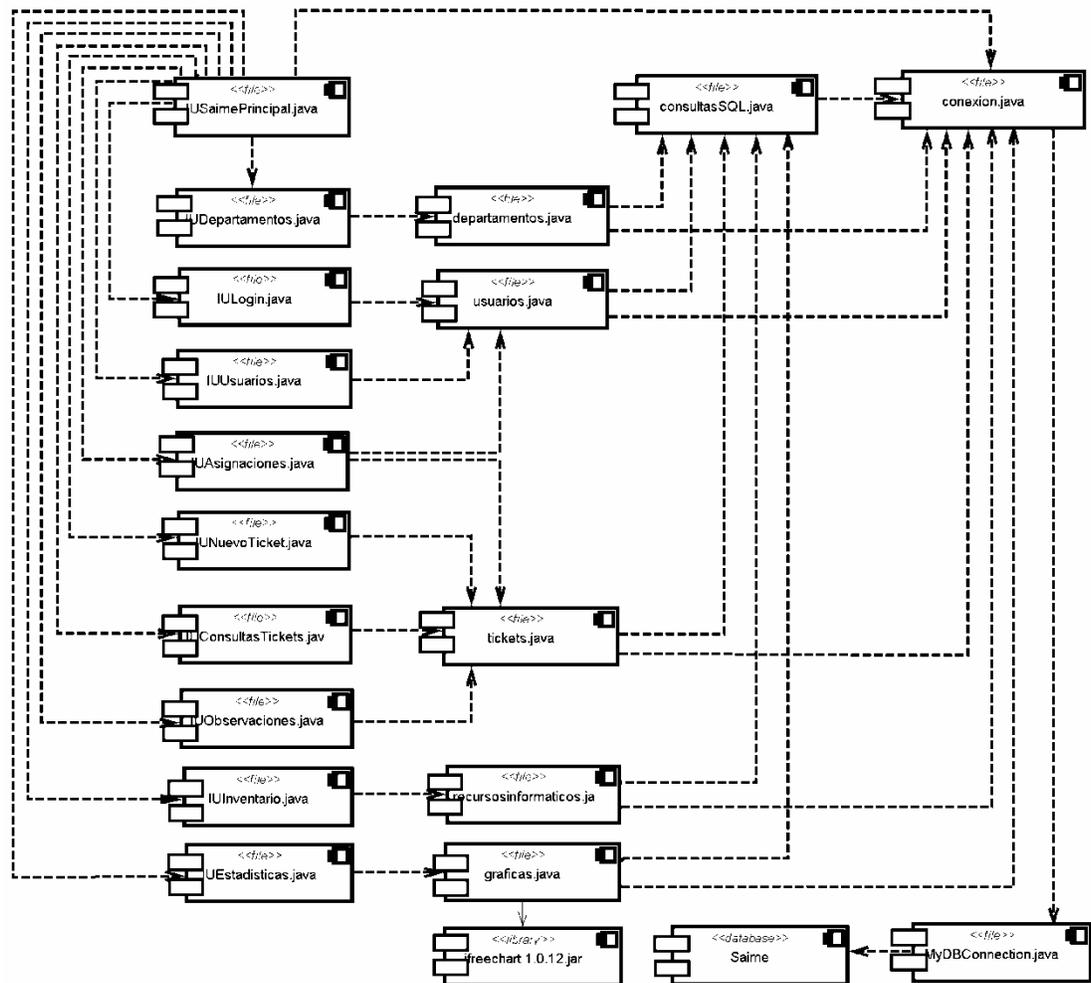


Figura 5.4. Diagrama de Componentes Total

[Fuente: Elaboración Propia]

5.2. Pruebas

En este flujo de trabajo se implementan las clases y objetos en ficheros fuente, binarios, Las pruebas son el instrumento que permite validar y verificar el software, es decir, son los procesos que determinan si el software satisface los requisitos y funciona de la manera establecida.

5.2.1. Pruebas por Unidad

Las pruebas por unidad se aplicaron mediante la aplicación de la prueba de la caja negra sobre los diversos componentes del sistema. Para realizar este tipo de pruebas, se identifican un conjunto de valores que pueden ser introducidos por un actor, y se expresan como clases de equivalencia para poder abarcar la totalidad de las ocurrencias de un evento de inserción de datos.

A continuación en la Tabla 5.1, se representan las clases de equivalencia del componente departamentos.java, el mismo se encarga de las actividades relacionadas a los departamentos, como lo son ingresar y modificar. (Ver Figura 5.4).

Tabla 5.1. Clase de equivalencia para el componente departamentos.java

No	DATO	Clase de Equivalencia	Valido	Invalido
1	Nombre	Carácter numérico		X
2	Nombre	Carácter alfanumérico	X	
3	Nombre	Longitud carácter <= 30	X	
4	Nombre	Dato nulo		X
5	Teléfono	Carácter numérico	X	
6	Teléfono	Carácter alfanumérico		X
7	Teléfono	Longitud carácter <= 30	X	
8	Teléfono	Dato nulo		X
9	Email	Carácter numérico		X
10	Email	Carácter alfanumérico	X	
11	Email	Longitud carácter <= 30	X	
12	Email	Dato nulo		X

[Fuente: Elaboración Propia]

En la Tabla 5.2 se presenta algunos casos de prueba de caja negra, donde se incluirán datos que serán cotejados por las clases de equivalencia referenciadas anteriormente. Si se cumplen todas las clases involucra con dicho dato, entonces la salida será validada. (Ver Figura 5.4).

Tabla 5.2. Casos de prueba de caja negra para el componente departamentos.java

DATO	Casos de Prueba	Salida	Clase Cubierta
Nombre	123	Invalido	1
Nombre	Logística 1	Valido	2,3
Nombre	Dato nulo	Invalido	4
Teléfono	4577	Valido	5
Teléfono	HC-3023	Invalido	6,7
Teléfono	Dato nulo	Invalido	8
Email	34466	Invalido	9
Email	Logistica1@hidrocentro.gov.ve	Valido	10,11
Email	Dato nulo	Invalido	12

[Fuente: Elaboración Propia]

5.2.2. Pruebas de Integración

El objetivo general de las pruebas de integración es, detectar las fallas de interacción entre las distintas clases que componen al sistema. Debido a que cada clase probada por separado se inserta de manera progresiva dentro de la estructura, las pruebas de integración son realmente un mecanismo para comprobar el correcto ensamblaje del sistema completo. Al efectuar la integración de los módulos, se concentra el esfuerzo en la búsqueda de fallas que puedan provocar excepciones arrojadas por los métodos; el empleo de operaciones equivocada, e invocación inadecuada de los métodos.

Luego de verificar la calidad de los componentes, se procede a comprobar la eficiencia de un conjunto de componentes integrados por fase, estas se pueden observar resaltadas en la Figura 5.5 en donde se despliega el diagrama de componentes por fase de integración del sistema.

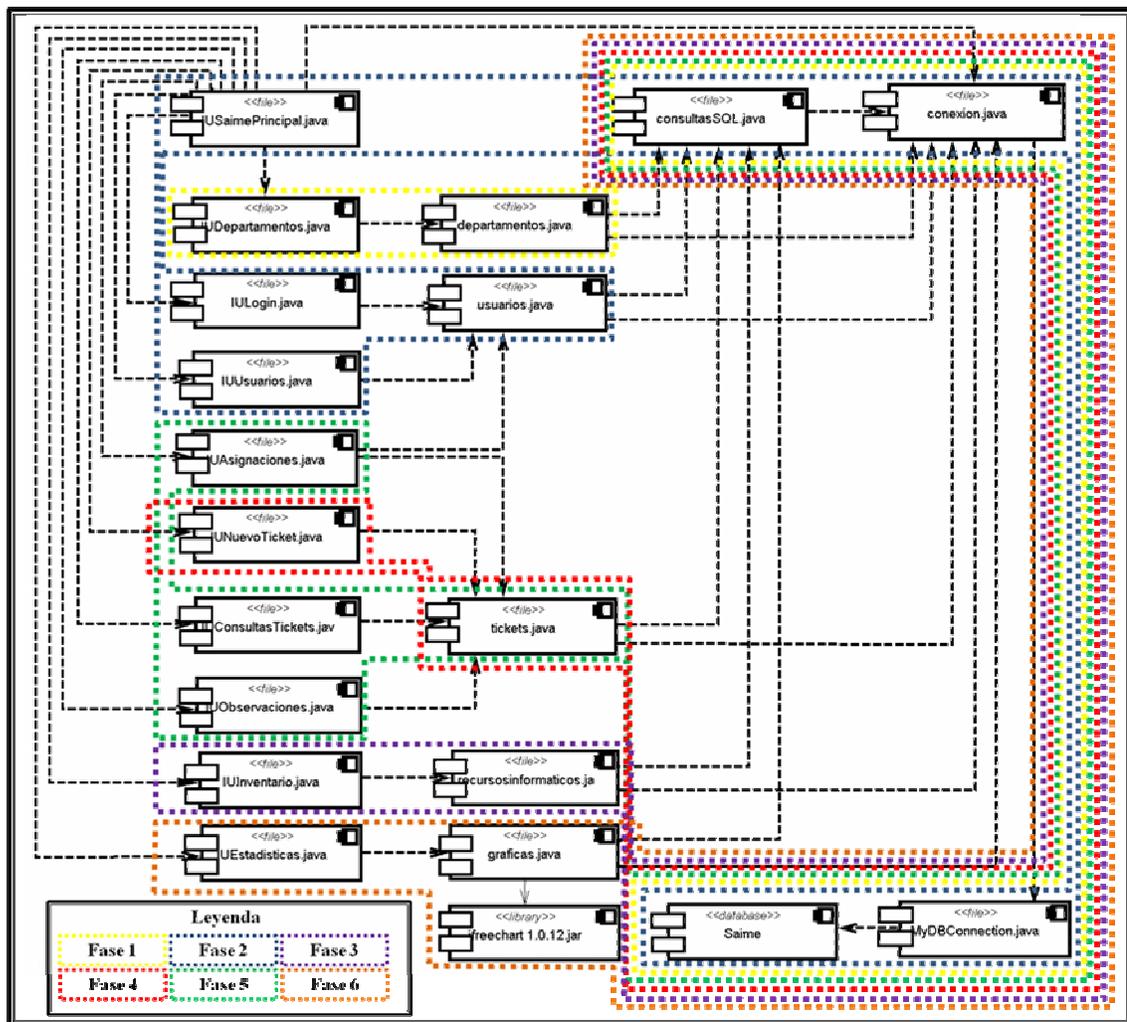


Figura 5.5. Diagrama de Componentes Total por fase de integración de SAIME

[Fuente: Elaboración Propia]

5.2.2.1. Pruebas de Integración de gestionar Departamento

Este caso de prueba verifica la funcionalidad de la implementación de la fase resaltada con el color amarillo de la Figura 5.5.

Entrada:

Tabla 5.3. Datos de entrada para probar la integración de Configuración de Señales

DATOS			
Nombre	Soporte y Comunicaciones	Teléfono	4577
Email	scomunicaciones@hidrocentro.gob.ve		

[Fuente: Elaboración Propia]

Procedimiento de Prueba:

- Activar la interfaz de usuario administrador.
- Acceder a la interfaz de administrar Departamentos.
- Introducir los datos de la Tabla 5.3 y hacer click en el botón Crear. (Ver Figura 5.6).
- Si los datos son correcto hacer Click en Si. (Ver Figura 5.7),
- Se consulta en la ventana el dato que se acaba de almacenar, esto para modificar en caso de ser necesario (Ver Figura 5.8).
- Se realiza la comunicación con la base de datos a través del componente conexión y se observa el departamento creado correctamente. (Ver Figura 5.9)

A continuación se presenta una secuencia de imágenes que muestran el procedimiento de integración:

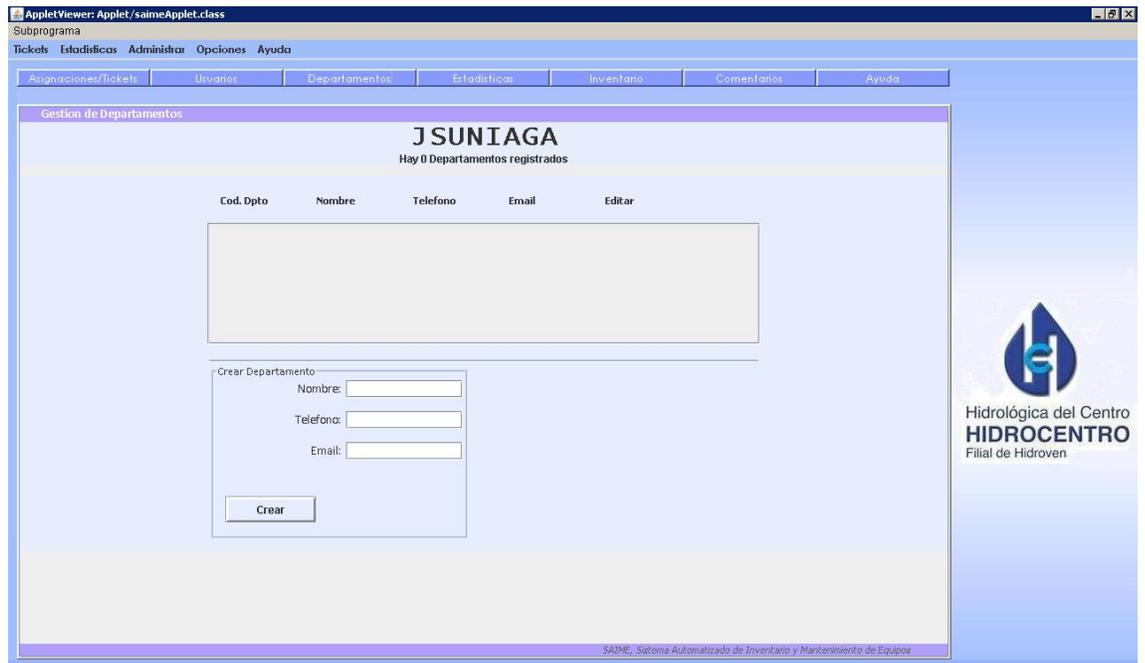


Figura 5.6. Interfaz de Gestión de Departamentos [Fuente: Elaboración Propia]

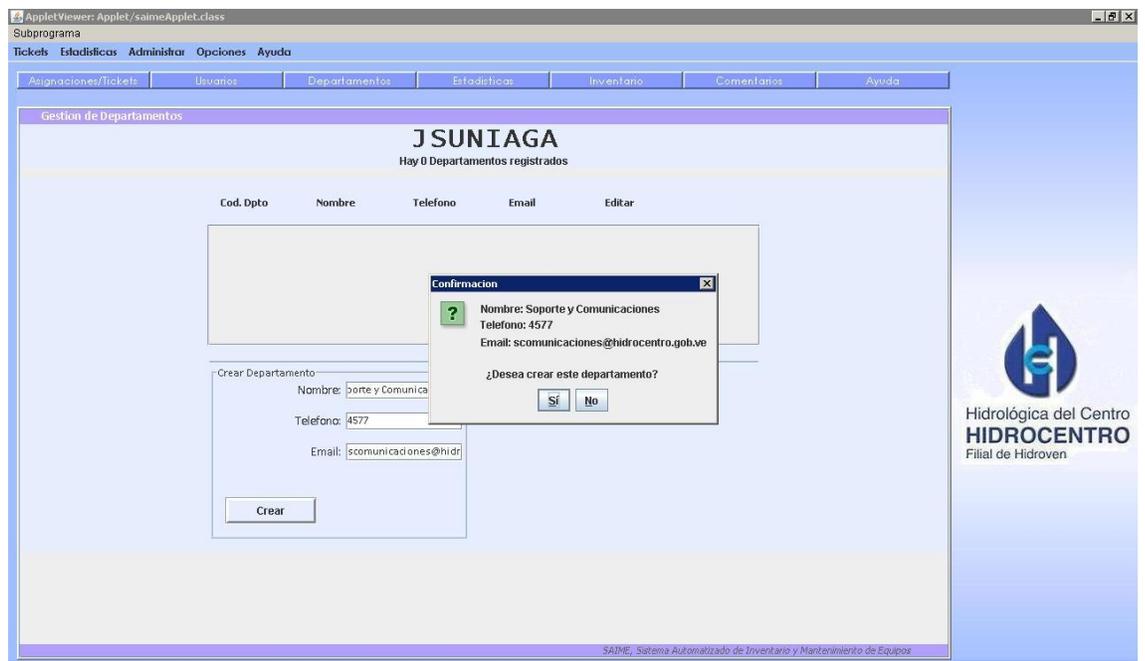


Figura 5.7. Introducción de datos en la ventana de Departamentos [Fuente: Elaboración Propia]

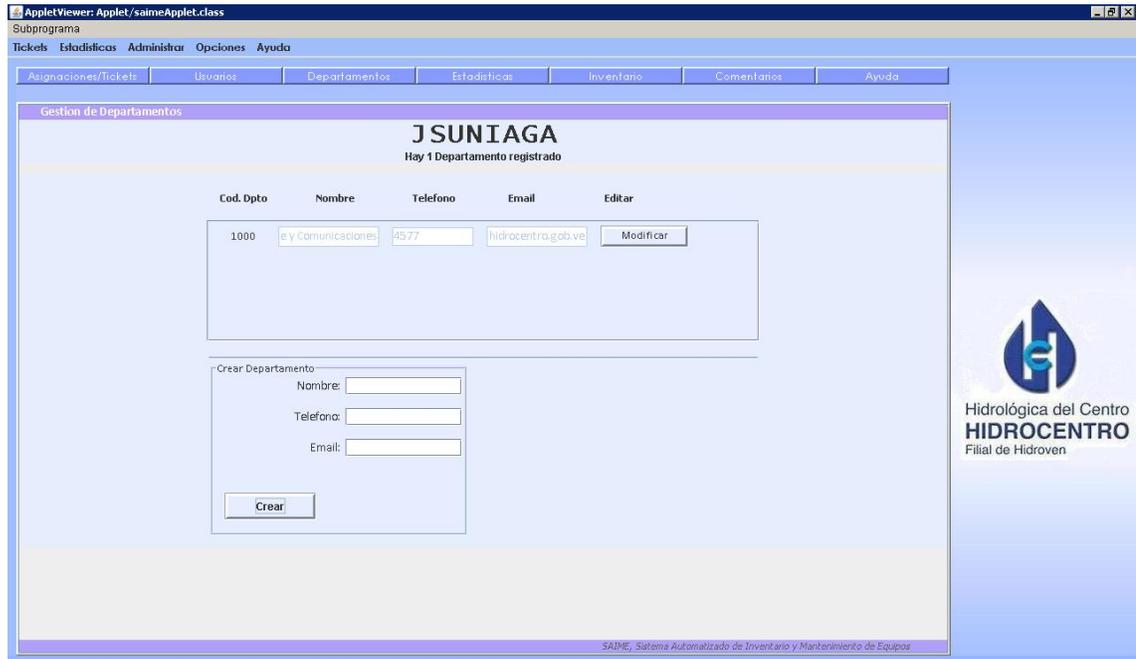


Figura 5.8. Consulta de datos en la ventana de Departamentos [Fuente: Elaboración Propia]

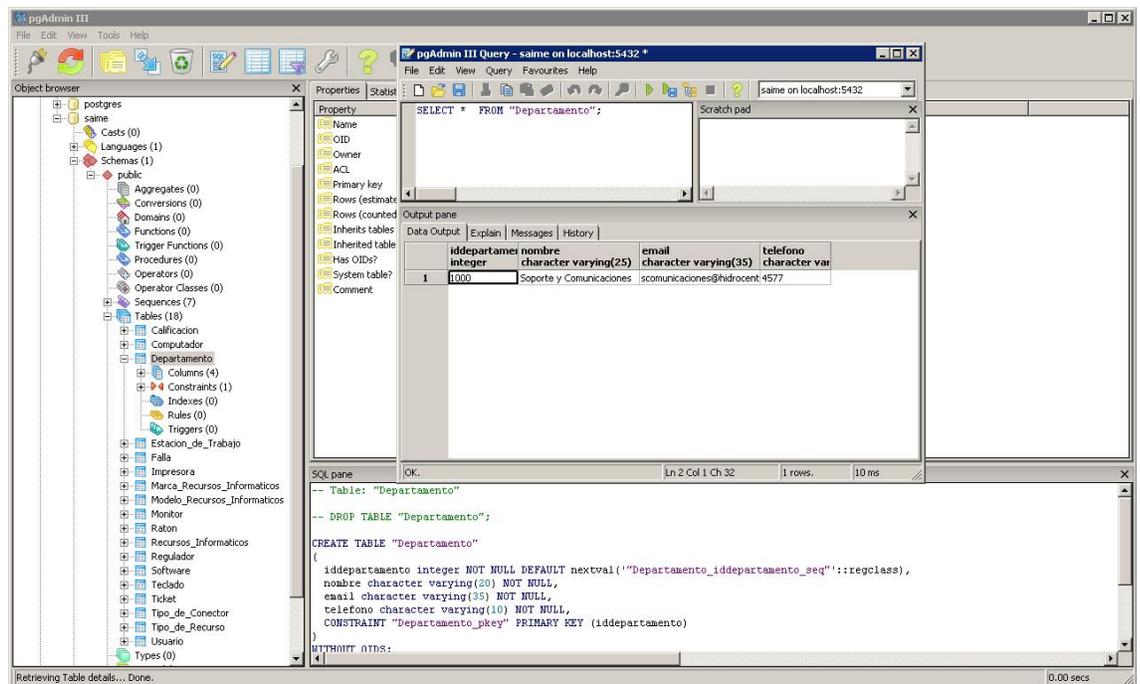


Figura 5.9. Detalle del departamento creado correctamente [Fuente: Elaboración Propia]

Codigo Crear Departamento: (Figuras 5.6, 5.7 y 5.8)

```

private void
BotonCrearDepartamentoActionPerformed(java.awt.event.ActionEvent evt) {
    if(CrearDepartamentoNombre2.getText().isEmpty() ||
        CrearDepartamentoTelefono2.getText().isEmpty() ||
        CrearDepartamentoEmail2.getText().isEmpty()){
        javax.swing.JOptionPane.showMessageDialog
            (null, "Debe completar todos los campos");
    }
    else{
        String salida =
            "Nombre: " + CrearDepartamentoNombre2.getText() + "\n"+
            "Telefono: " + CrearDepartamentoTelefono2.getText() + "\n" +
            "Email: " + CrearDepartamentoEmail2.getText()+"\n\n " +

            "¿Desea crear este departamento?";

        int opcion = javax.swing.JOptionPane.showConfirmDialog
            (null,salida,"Confirmacion",javax.swing.JOptionPane.YES_NO_OPTION);

        if(opcion == javax.swing.JOptionPane.YES_OPTION){
            int proximo;
            if(lista_dptos.isEmpty())
                proximo = 1000;
            else
                proximo = Integer.valueOf(lista_dptos.get(
                    lista_dptos.size()-1).getDpto_ID()+10;

            if(dpto_existe && insertar_dpto()){

```

```

departamentos nuevo =
    new departamentos(String.valueOf(proximo));
    CrearDepartamentoNombre2.getText();
    CrearDepartamentoTelefono2.getText();
    CrearDepartamentoEmail2.getText();

    lista_dptos.add(nuevo);
    int index = lista_dptos.indexOf(nuevo);
    PanelModificarDepartamentos2.add(lista_dptos.get(index));
    PanelModificarDepartamentos2.setVisible(false);
    PanelModificarDepartamentos2.setVisible(true);
    CrearDepartamentoNombre2.setText("");
    CrearDepartamentoTelefono2.setText("");
    CrearDepartamentoEmail2.setText("");
    }
    }
}
}

```

Codigo del Método insertar_dpto:

```

public boolean inserter_dpto(){
    boolean listo = true;
    "Nombre: " ++ "\n"+
    "Telefono: " ++ "\n" +
    "Email: " ++ "\n\n " +

    String Nombre = CrearDepartamentoNombre2.getText();
    String Email = CrearDepartamentoEmail2.getText();
    String Tlf = CrearDepartamentoTelefono2.getText();
    String sql;

```

```

        try{
            sql = "INSERT INTO departamentos VALUES ('"+nombre+"','"+email+"",
            "+telefono+"");
        }catch(Exception e){ listo = false;
            javax.swing.JOptionPane.showMessageDialog(null,e.toString(),
            "ERROR Insertar Departamento",
            javax.swing.JOptionPane.ERROR_MESSAGE);
        }
        return listo;
    }

```

5.3. Evaluación de la Fase de Construcción

Durante la fase de construcción se complementó el diseño de la arquitectura base de SAIME y se codificaron e integraron todos los componentes del sistema. Las pruebas de unidad se aplicaron mediante el método de caja negra, adicionalmente se aplicaron pruebas de integración lo cual permitió detectar y corregir fallas en el funcionamiento de los componentes y en la forma de acoplarse unos a otros.

Durante la construcción, se amortizaron los riesgos correspondientes a esta fase, logrando el propósito de tener lista la primera versión operativa del sistema.

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

Mediante este proyecto de investigación se creó una herramienta de helpdesk para ofrecer servicio técnico e inventario, a través de una interfaz gráfica sencilla que facilitará la búsqueda, actualización de información y optimizará el tiempo de respuesta por parte del personal de la gerencia de informática en caso de una falla asociada a los recursos informáticos de la empresa C.A. HIDROCENTRO.

El RUP, (Rational Unified Process), como metodología elegida, aunado al uso de UML como herramienta principal para la documentación y guía, permitieron la organización y desarrollo del sistema, cumpliendo con todas las etapas de este proceso y ayudó a la prevención de fallas que se pudieron presentar, evitando así un replanteamiento del proyecto, canalizando los diferentes flujos de trabajo necesarios por cada una de las fases de desarrollo.

La fase de inicio del proyecto jugó un papel fundamental, puesto que permitió obtener una noción clara del contexto del sistema, a través del Modelo de Dominio y el diagrama de Casos de Uso, logrando con ello la obtención de los requisitos necesarios para llevar a cabo el diseño. Los casos de uso también fueron de gran importancia para representar las condiciones y requerimientos del sistema, logrando esto a través de una buena documentación y análisis.

La fase de elaboración permitió eliminar la mayor cantidad de riesgos posibles empezando por los más altos o de más relevancia. Los requisitos funcionales y no funcionales además de toda información obtenida en las etapas anteriores, fueron refinadas en esta fase, y adecuadas a las posibilidades de la herramienta de desarrollo que se usó. También se definieron los flujos de trabajo en diseño y organización del sistema, representados por los diagramas correspondientes.

En la fase de construcción se procedió con la codificación de todos los componentes hasta obtener una versión beta del sistema.

La escogencia de Java como lenguaje de programación permitió hacer una limpia codificación del sistema ya que en las etapas anteriores se planteo una solida estructura de clases en conformidad con el paradigma orientado a objetos y Java es un lenguaje programación orientado a objetos de última generación y figura como uno de los lenguajes con un mayor crecimiento y amplitud. Por otra parte Java es netamente orientado a aplicaciones gráficas y posee potentes herramientas para el desarrollo de GUI's, lo que facilitó la construcción de Interfaces Graficas organizadas, comunicativas y muy amigables.

Se realizó de manera exitosa la integración, pruebas y documentación de funcionamiento del sistema, se evaluó y depuró el sistema en varias iteraciones, obteniendo un software altamente funcional, al que puede realizarse mantenimiento y actualizaciones.

6.2. Recomendaciones

Efectuar un mantenimiento periódico a la base de datos, realizando respaldos, eliminando de las tablas los usuarios inactivos, también agregando recursos informáticos nuevos, para así garantizar el correcto funcionamiento del sistema, optimizar los tiempos de respuesta de los tickets emitidos y mantener la información actualizada.

Mantener actualizados los documentos de diseño y manuales de usuario, con respecto a cualquier modificación o actualización que se realice al sistema y llevar un registro de estos.

Desarrollar a futuro módulos para la automatización de la información referentes a las fallas presentadas en los recursos para hacer la gestión del registro de nuevas fallas, esto con la finalidad de almacenar un historial de fallas ocurridas para determinar las más comunes y los equipos que reinciden en las mismas.

Implementar a futuro un gestor de reportes impresos y electrónicos, en donde sea posible representar los datos obtenidos mediante tablas para llevar un control más preciso de los mantenimientos realizados a los equipos a través del tiempo.

BIBLIOGRAFÍA

[1] CABRERA, A. “Desarrollo de un software educativo e interactivo como apoyo en el proceso enseñanza – aprendizaje de la asignatura matemáticas de séptimo grado de Educación Básica, utilizando la tecnología World Wide Web”. Trabajo Especial de Grado, no publicado. Universidad de Oriente. Puerto La Cruz (2002).

[2] FERMÍN, R. “Diseño de un Sistema de Información para el Monitoreo de la Red LAN y de Apoyo al Departamento de Redes y Comunicaciones de una Empresa Siderúrgica”. Trabajo de Grado, Ingeniería de Sistemas, Universidad de Oriente. Anzoátegui, Venezuela (2005).

[3] VELASCO, M. y Gotilla, N. “Diseño de un sistema de información bajo licencia de software libre para el manejo de las actividades administrativas de los módulos compra y venta en una pequeña y mediana empresa”. Trabajo Especial de Grado, no publicado. Universidad de Oriente. Puerto La Cruz (2006).

[4] OBANDO, M y Sales, A. “Desarrollo de un software para automatizar el control de donantes en el banco de sangre del Hospital Universitario Dr. Luis Razetti de Barcelona, Estado Anzoategui”. Trabajo Especial de Grado, no publicado. Universidad de Oriente. Puerto La Cruz (2008).

[5] MIDDELTON, I y Marcella, R. “Key Factors in Help Desk Succes”. Ensayo, Biblioteca Británica. Londres, Inglaterra (1996).

[6] FERNANDEZ, C. “Aplicación informática de factura nacional”. Disponible en: http://www.elperuano.com.pe/edc/2009/06/30/cien_tec1.asp [Consulta: 15 de Julio de 2009]

[7] ORÉ, F. “Conceptos: Servidor, Hosting y HTML”. Disponible en: <http://franchusca2008.blogspot.com/2008/08/mundo-digital.html> [Consulta: 17 de Julio de 2009]

[8] ELMASRI y NAVATHE. “Sistemas de Base de Datos”. Addison Wesley Iberoamericana. USA (1997).

[9] GARCÍA-CARO, I. Diseño de una Aplicación Web para el conocimiento y conversión de unidades. Tesis de Grado, Universidad Nacional de Educación a Distancia. Madrid, España (2006).

[10] SÁNCHEZ, C. “ONess: un proyecto open source para el negocio textil mayorista desarrollado con tecnologías open source innovadoras”. Tesis de Grado, Universidad de La Coruña. La Coruña, España (2004).

[11] LEMAY, L. “Aprendiendo Java 2 en 21 dias”. Prentice Hall. Mexico (2003).

[12] DEITEL y DEITEL. “Como programar en Java”. Prentice Hall. Mexico (2004).

[13] BOUDREU, T y Wielenga, G. “Rich Client Programming – Plugging into NetBeans Platform”. Prentice Hall. Delaware, USA (2004).

[14] MORA, O. “Bases de Datos”. Fundació per a la Universitat Oberta de Catalunya. Barcelona, España (2005).

[15] FARIAS, J. “UML”. Pearson México (2003).

[16] KRUTCHEN, P. “The Rational Unified Process An Introduction”. Addison -Wesley. USA (2004).

[17] RUEDA, J. “Aplicación de la metodología RUP para el desarrollo rápido de aplicaciones basado en el estándar J2EE”. Tesis de Grado, Universidad San Carlos de Guatemala. Guatemala (2006).

BINDER, R. “Testing Object-Oriented Systems: Models, Patterns and Tools”. EditorialAddison-Wesley

CEBALLOS, F. Java 2: Interfaces gráficas y aplicaciones para Internet. RAMA. España, 2006.

JACABOSON, I. Booch, G. y Rumbaugh J. “El Proceso Unificado de Desarrollo de Software”. Addison - Wesley. USA, 2000.

HOLZNER, S. La Biblia del Java 2. Anaya Multimedia. España, 2000.

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

TÍTULO	DESARROLLO DE UNA APLICACIÓN WEB BASADA EN TECNOLOGÍA HELPDESK PARA OFRECER SERVICIOS DE SOPORTE TÉCNICO E INVENTARIO EN LA GERENCIA DE INFORMÁTICA DE LA EMPRESA C.A. HIDROLÓGICA DEL CENTRO, EN VALENCIA ESTADO CARABOBO
SUBTÍTULO	

AUTOR (ES):

APELLIDOS Y NOMBRES	CÓDIGO CULAC / E MAIL
Suniaga S., Jose M.	CVLAC: V-16.827.771 E MAIL: suniagajose@gmail.com
	CVLAC: E MAIL:
	CVLAC: E MAIL:
	CVLAC: E MAIL:

PALÁBRAS O FRASES CLAVES:

HELPDESK

INVENTARIO

SOPORTE TECNICO

APLICACION WEB

JAVA

POSTGRESQL

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

ÀREA	SUBÀREA
Ingeniería y Ciencias aplicada	
	Ingeniería en Computación

RESUMEN (ABSTRACT):

La Gerencia de Informática de la empresa C.A. Hidrológica del Centro, HIDROCENTRO tiene como objetivo ofrecer a los empleados de la empresa servicios de calidad en el área de información; desarrollando sistemas óptimos y dando soporte a los recursos informáticos con eficiencia, en el menor tiempo posible. En tal sentido, la Gerencia de Informática debe: mantener un control del inventario de los recursos informáticos de la empresa así como de los servicios que se prestan a estos, de manera que se pueda gestionar los equipos que estén activos y vigilar el desempeño de los servicio de soporte técnico a cargo de los empleados de la gerencia. Actualmente, la gerencia lleva este control de manera manual, es por esa razón que se requiere de un proyecto que permita la automatización de dichos procesos. En este informe de trabajo de grado se presenta el diseño de un Sistema de Administración de Inventario y Mantenimiento de Equipos (SAIME) que apoyará los procesos descritos anteriormente, el cual está modelado y documentado bajo el Lenguaje Unificado de Modelado (UML), siguiendo la metodología RUP, implantado bajo la plataforma Windows, programado con el lenguaje de programación Java y cuyos datos son almacenados en una base de datos PostgreSQL.

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

CONTRIBUIDORES:

APELLIDOS Y NOMBRES	ROL / CÓDIGO CVLAC / E_MAIL				
Primera G., Eduardo M.	ROL	CA	AS X	TU	JU
	CVLAC:	V-12.322.319			
	E_MAIL	<u>eprimera@hidrocentro.gov.ve</u>			
	E_MAIL				
García R., Zulirais A.	ROL	CA	AS	TU X	JU
	CVLAC:	V-10.299.576			
	E_MAIL	<u>zzzuliii@hotmail.com</u>			
	E_MAIL				
Rodríguez, Rhonald	ROL	CA	AS	TU	JU X
	CVLAC:	V-14.077.185			
	E_MAIL	<u>rhoen2003@hotmail.com</u>			
	E_MAIL				
Veracierta, Gabriela	ROL	CA	AS	TU	JU X
	CVLAC:	V-14.616.683			
	E_MAIL	<u>gveracierta@hotmail.com</u>			
	E_MAIL				

FECHA DE DISCUSIÓN Y APROBACIÓN:

2009	10	22
AÑO	MES	DÍA

LENGUAJE. SPA

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

ARCHIVO (S): TESIS

NOMBRE DE ARCHIVO	TIPO MIME
TESIS.SAIME.helpdesk.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: _____ **HIDROCENTRO** _____ (OPCIONAL)

TEMPORAL: _____ **6 MESES** _____ (OPCIONAL)

TÍTULO O GRADO ASOCIADO CON EL TRABAJO:

_____ Ingeniero Computación _____

NIVEL ASOCIADO CON EL TRABAJO:

_____ Pre-Grado _____

ÁREA DE ESTUDIO:

_____ Departamento de Computación y Sistemas _____

INSTITUCIÓN:

_____ Universidad de Oriente – Núcleo de 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 sólo podrán ser utilizados a otros fines con el consentimiento del consejo de núcleo respectivo, quien lo participará al Consejo Universitario”

Suniaga S., José M.
AUTOR

García, Zulirais
TUTOR

Rodríguez, Rhonald
JURADO

Veracierta, Gabriela
JURADO

POR LA SUBCOMISIÓN DE TESIS