

Universidad de las Ciencias Informáticas

Facultad 4



Título: Desarrollo de un Sistema de Gestión y Control de Interrupciones de Software para la Dirección Territorial de ETECSA en Granma.

Trabajo de Diploma para optar por el Título de Ingeniero en
Ciencias Informáticas

Autor: Howard Caballero Daria

Tutor: Lic. Liusmer Martínez Quintana

Ciudad de la Habana, Junio 2009

AGRADECIMIENTOS

Agradezco a:

Lizandra Arza, Karina Sánchez, Rainer Batista y Liusmer Martínez Quintana por hacer posible la realización de este trabajo.

Mis amigos por compartir los buenos y los malos momentos.

Félix, Dayima, Rigoberto y todos aquellos que dieron su grano de arena.

A mi madre y hermano por su descomunal preocupación.

Y especialmente a mi esposa por toda su ayuda.

Para todos ustedes Gracias.

DEDICATORIA

En especial a mi Hijo por ser lo más grande y bello que me ha dado la vida.

A mi Esposa por todo su amor, entrega y comprensión.

A mi Madre y mi Hermano por haberme guiado por el mejor camino.

A Mima por todo y en especial por ese amor de madre que nunca escondiste.

A Yasmin por su comprensión.

A todos los que ya no están para ustedes también va dedicado mi trabajo.

Declaro ser autor de la presente Tesis y reconozco a la Universidad de las Ciencias Informáticas y la Dirección Territorial de ETECSA en Granma los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de ____ del año 2009.

Howard Caballero Daria

Autor

Ing. Liusmer Martínez Quintana

Tutor

RESUMEN

En la Dirección Territorial de ETECSA en la provincia Granma, la Unidad de Gestión y Administración de Aplicaciones (UGAA) decidió informatizar el proceso de gestión y control de las interrupciones de Software que reciben. Actualmente esta actividad se realiza de forma no automatizada. Esta investigación tiene como objetivo desarrollar una Sistema que permita gestionar y controlar las interrupciones de software en la Dirección Territorial de ETECSA en Granma. Para ello se realizó un estudio comparativo entre herramientas, tecnologías, lenguajes y metodologías de soporte al desarrollo, luego de obtener un entendimiento de la organización en la empresa en cuanto a este proceso. Permitiendo confeccionar la propuesta de la solución y la implementación del sistema. Se estimaron los esfuerzos humanos, el tiempo de desarrollo y los costos, obteniéndose una aplicación web accesible para todos los trabajadores de la empresa en el territorio. Los afectados pueden reportar las interrupciones, los responsables de solución atenderlas desde cualquier parte del región y el Jefe de la UGAA cuenta con una rápida accesibilidad a la información que se necesite. Se utilizó para el desarrollo del sistema el lenguaje **PHP**, el gestor de base de datos **MySQL**, el gestor de contenido **Joomla** y el uso de **Visual Paradigm** para el modelado UML, basado en la metodología **RUP**.

Palabras Claves:

Interrupción, trabajadores afectados, trabajadores responsable, ETECSA

ÍNDICE DE FIGURAS

FIGURA 1 DIAGRAMA DE CASO DE USO DEL NEGOCIO	40
FIGURA 2: DIAGRAMA DE ACTIVIDADES REPORTAR INTERRUPCIÓN	42
FIGURA 3: MODELO DE OBJETOS DE NEGOCIO	43
FIGURA 4: DIAGRAMA DE CASOS DE USO DEL SISTEMA.....	49
FIGURA 5: DIAGRAMA DE CLASES DE ANÁLISIS DEL CUS REPORTAR INTERRUPCIÓN.....	67
FIGURA 6: DIAGRAMA DE CLASES DE ANÁLISIS DEL CUS ATENDER INTERRUPCIÓN.....	67
FIGURA 7: DIAGRAMA DE CLASES DE ANÁLISIS DEL CUS GESTIONAR INTERRUPCIÓN.....	68
FIGURA 8: DIAGRAMA DE CLASES DE ANÁLISIS DEL CUS OBTENER REPORTE.	68
FIGURA 9: DIAGRAMA DE CLASES DE ANÁLISIS DEL CUS ASIGNAR RESPONSABLE.	69
FIGURA 10: DIAGRAMA DE CLASES DE ANÁLISIS DEL CUS GESTIONAR SOFTWARE.....	69
FIGURA 11: DIAGRAMA DE CLASES DE ANÁLISIS DEL CUS GESTIONAR RESPONSABLE.	70
FIGURA 12: DIAGRAMA DE CLASES WEB DEL CUS REPORTAR INTERRUPCIÓN.....	71
FIGURA 13: DIAGRAMA DE CLASES WEB DEL CUS ATENDER INTERRUPCIÓN.....	72
FIGURA 14: DIAGRAMA DE CLASES WEB DEL CUS GESTIONAR INTERRUPCIÓN.....	73
FIGURA 15: DIAGRAMA DE CLASES WEB DEL CUS GESTIONAR RESPONSABLE.	74
FIGURA 16: DIAGRAMA DE CLASES WEB DEL CUS ASIGNAR RESPONSABLE.	75
FIGURA 17: DIAGRAMA DE CLASES WEB DEL CUS OBTENER REPORTE.	76
FIGURA 18: DIAGRAMA DE CLASES WEB DEL CUS GESTIONAR SOFTWARE.	77
FIGURA 19: DIAGRAMA DE SECUENCIA DEL CUS REPORTAR INTERRUPCIÓN.....	78
FIGURA 20: DIAGRAMA DE SECUENCIA DEL CUS ATENDER INTERRUPCIÓN.....	79
FIGURA 21: DIAGRAMA DE SECUENCIA DEL CUS ASIGNAR RESPONSABLE.	80
FIGURA 22: DIAGRAMA DE SECUENCIA DEL CUS OBTENER REPORTE.....	81
FIGURA 23: DIAGRAMA DE CLASES PERSISTENTES.....	87
FIGURA 24: DIAGRAMA DE COMPONENTES DEL CUS REPORTAR INTERRUPCIÓN.....	94
FIGURA 25: DIAGRAMA DE COMPONENTES DEL CUS ATENDER INTERRUPCIÓN.....	94
FIGURA 26: DIAGRAMA DE COMPONENTES DEL CUS ASIGNAR RESPONSABLE.....	95
FIGURA 27: DIAGRAMA DE COMPONENTES DEL CUS OBTENER REPORTE.	95
FIGURA 28: DIAGRAMA DE COMPONENTES DEL CUS GESTIONAR INTERRUPCIÓN.....	96
FIGURA 29: DIAGRAMA DE COMPONENTES DEL CUS GESTIONAR RESPONSABLE.	96
FIGURA 30: DIAGRAMA DE COMPONENTES DEL CUS GESTIONAR SOFTWARE.....	97
FIGURA 31: DIAGRAMA DE DESPLIEGUE.	98

ÍNDICE DE TABLAS

TABLA 1: DESCRIPCIÓN DE LOS ACTORES DEL NEGOCIO. -----	39
TABLA 2: DESCRIPCIÓN DE LOS TRABAJADORES DEL NEGOCIO. -----	39
TABLA 3: DESCRIPCIÓN TEXTUAL DEL CUN REPORTAR INTERRUPCIÓN. -----	41
TABLA 4: DESCRIPCIÓN DE LOS ACTORES DEL SISTEMA. -----	48
TABLA 5: DESCRIPCIÓN DEL CUS CREAR INTERRUPCIÓN. -----	51
TABLA 6: DESCRIPCIÓN DEL CUS ATENDER INTERRUPCIÓN. -----	53
TABLA 7: DESCRIPCIÓN DEL CUS GESTIONAR INTERRUPCIÓN. -----	55
TABLA 8: DESCRIPCIÓN DEL CUS ASIGNAR RESPONSABLE. -----	56
TABLA 9: DESCRIPCIÓN DEL CUS OBTENER REPORTES -----	59
TABLA 10: DESCRIPCIÓN DEL CUS GESTIONAR SOFTWARE. -----	62
TABLA 11: DESCRIPCIÓN DEL CUS GESTIONAR RESPONSABLE. -----	65
TABLA 12: DESCRIPCIÓN DE LA CLASE CONTROLADORA CC_GESTIONAR_INTERRUPCION. -----	82
TABLA 13: DESCRIPCIÓN DE LA CLASE CONTROLADORA CC_GESTIONAR_RESPONSABLE. -----	82
TABLA 14: DESCRIPCIÓN DE LA CLASE DE ACCESO A DATO ACCESOS_DATO. -----	83
TABLA 15: DESCRIPCIÓN DE LA CLASE DE ACCESO A DATO AD_REPORTAR_INTERRUPCION. -----	83
TABLA 16: DESCRIPCIÓN DE LA CLASE DE ACCESO A DATO AD_ATENDER_INTERRUPCCION. -----	84
TABLA 17: DESCRIPCIÓN DE LA CLASE DE ACCESO A DATO AD_GESTIONAR_RESPONSABLE -----	84
TABLA 18: DESCRIPCIÓN DE LA CLASE DE ACCESO A DATO AD_GESTIONAR_SOFTWARE. -----	85
TABLA 19: DESCRIPCIÓN DE LA CLASE DE ACCESO A DATO AD_ASIGANAR_RESPONSABLE. -----	85
TABLA 20: DESCRIPCIÓN DE LA CLASE ENTIDAD CE_SOFTWARE. -----	85
TABLA 21: DESCRIPCIÓN DE LA CLASE ENTIDAD CE_RESPONSABLE. -----	86
TABLA 22: DESCRIPCIÓN DE LA CLASE ENTIDAD CE_INTERRUPCION. -----	87
TABLA 23: FACTOR DE PESO DE LOS ACTORES. -----	99
TABLA 24: FACTOR DE PESO DE LOS CASOS DE USO SIN AJUSTAR. -----	100
TABLA 25: TRANSACCIONES Y PESO DE LOS CASOS DE USO. -----	100
TABLA 26: FACTOR DE COMPLEJIDAD TÉCNICA. -----	102
TABLA 27: FACTOR DE AMBIENTE. -----	102
TABLA 28: DISTRIBUCIÓN PORCIENTO-HORAS-HOMBRE. -----	104

TABLA DE CONTENIDO

ÍNDICE DE FIGURAS..... 6

ÍNDICE DE TABLAS..... 7

INTRODUCCIÓN12

CAPITULO 1: FUNDAMENTACIÓN TEÓRICA16

1.1 Introducción.....16

1.2 Sistemas de Gestión de Interrupciones de Software16

 1.2.1 Sistemas de Gestión de Interrupciones de Software Internacionales.....16

 1.2.2 Sistema de Gestión de Interrupciones de Software en Cuba17

1.3 Tendencias y Tecnologías Actuales18

 1.3.1 Aplicaciones Web18

 1.3.2 Arquitectura Cliente-Servidor.....19

 1.3.3 Técnicas y tecnologías del lado del cliente19

 1.3.4 Tecnologías del lado del servidor.....21

 1.3.5 Gestor de Base de Datos.....23

 1.3.6 Servidor de aplicaciones Web24

 1.3.7 Metodologías de desarrollo de software.....24

 1.3.8 Herramientas CASE.28

 1.3.9 Herramientas de Desarrollo.30

 1.3.10 Sistema de Gestión de Contenido.....31

1.4 Conclusiones.....34

CAPITULO 2: CARACTERÍSTICAS DEL SISTEMA.....35

2.1 Introducción.....35

2.2 Objeto de estudio.....35

 2.2.1 Análisis de la Situación Problémica35

 2.2.2 Objeto de automatización.....36

 2.2.3 Información que se maneja.....37

2.3 Propuesta del sistema.37

2.4 Modelado del Negocio.38

2.5 Actor y Trabajadores del Negocio.....	39
2.6 Diagrama de Casos de Uso del Negocio.....	39
2.7 Descripciones Textuales.....	40
2.7.1 Descripción Textual del CUN Reportar Interrupción.....	40
2.8 Diagramas de Actividades.....	42
2.10 Requerimientos.....	43
2.10.1 Requisitos Funcionales.....	43
2.10.2 Requisitos no Funcionales.....	45
2.11 Modelación del Sistema.....	48
2.12 Descripción de los Actores del sistema.....	48
2.14 Descripción de los Casos de Uso del Sistema.....	49
2.14.1 Descripción del CUS “Reportar Interrupción”.....	50
2.14.2 Descripción del CUS “Atender Interrupción”.....	51
2.14.3 Descripción del CUS “Gestionar Interrupción”.....	53
2.14.4 Descripción del CUS “Asignar Responsable”.....	55
2.14.5 Descripción del CUS “Obtener Reportes”.....	56
2.14.6 Descripción del CUS “Gestionar Software”.....	60
2.14.7 Descripción del CUS “Gestionar Responsable”.....	62
2.15 Patrones de Casos de Usos.....	65
2.16 Conclusiones.....	65
CAPITULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.....	66
3.1 Introducción.....	66
3.2 Diagrama de Clases del Análisis.....	66
3.2.1 Diagrama de Clases del Análisis del CUS Reportar Interrupción.....	67
3.2.2 Diagrama de Clases del Análisis del CUS Atender Interrupción.....	67
3.2.3 Diagrama de Clases del Análisis del CUS Gestionar Interrupción.....	68
3.2.4 Diagrama de Clases del Análisis del CUS Obtener Reporte.....	68
3.2.5 Diagrama de Clases del Análisis del CUS Asignar Responsable.....	69
3.2.6 Diagrama de Clases del Análisis del CUS Gestionar Software.....	69
3.2.7 Diagrama de Clases del Análisis del CUS Gestionar Responsable.....	70

3.3 Diagrama de Clases del Diseño.	70
3.3.1 Diagrama de Clases Web para el CUS Reportar Interrupción.	71
3.3.2 Diagrama de Clases Web para el CUS Atender Interrupción.	72
3.3.3 Diagrama de Clases Web para el CUS Gestionar Interrupción.	73
3.3.4 Diagrama de Clases Web para el CUS Gestionar Responsable.	74
3.3.5 Diagrama de Clases Web para el CUS Asignar Responsable.	75
3.3.6 Diagrama de Clases Web para el CUS Obtener Reporte.	76
3.3.7 Diagrama de Clases Web para el CUS Gestionar Software.	77
3.4 Diagramas de Interacción.	78
3.4.1 Diagrama de Secuencia del CUS Reportar Interrupción.	78
3.4.2 Diagrama de Secuencia del CUS Atender Interrupción.	79
3.4.3 Diagrama de Secuencia del CUS Asignar Responsable.	80
3.4.4 Diagrama de Secuencia del CUS Obtener Reporte.	81
3.5 Descripción Textual de las clases Web.	81
3.5.1 Clase Controladora CC_Gestionar_Interrupcion.	82
3.5.3 Clase Controladora CC_Gestionar_Responsable.	82
3.5.8 Clase de acceso a dato Acceso_Dato.	82
3.5.9 Clase de acceso a dato AD_Reportar_Interrupcion.	83
3.5.10 Clase de acceso a dato AD_Atender_Interrupcion.	83
3.5.11 Clase de acceso a dato AD_Gestionar_Responsable.	84
3.5.13 Clase de acceso a dato AD_Gestionar_Software.	84
3.5.15 Clase de acceso a dato AD_Asignar_Responsable.	85
3.5.17 Clase entidad CE_Software.	85
3.5.17 Clase entidad CE_Responsable.	86
3.5.18 Clase entidad CE_Interrupcion.	86
3.6 Diagrama de clases persistentes.	87
3.7 Patrones Utilizados	88
3.7.1 Patrones de Distribución de Responsabilidades.	88
3.7.2 Patrón Modelo Vista Controlador (MVC).	88
3.8 Estándar de codificación.	89

3.9 Conclusiones	92
CAPITULO 4: IMPLEMENTACIÓN	93
4.1 Introducción	93
4.2 Modelo de Implementación	93
4.2.1 Diagrama de Componentes	93
4.3 Modelo de Despliegue	97
4.3.1 Diagrama de Despliegue.....	98
4.4 Estudio de Factibilidad	98
4.4.1 Planificación mediante el método de análisis por Puntos de Casos de Uso.....	98
4.4.2 Costo.....	105
4.4.2.1 Tangibles.....	105
4.4.2.2 Intangibles.....	105
4.4.3 Análisis costo-beneficio.....	106
4.4 Conclusiones	106
CONCLUSIONES.....	107
RECOMENDACIONES.....	108
BIBLIOGRAFIA	109
ANEXOS	112
Anexo #1: Ventajas y desventajas de los lenguajes de programación	112
Anexo #2: Ventajas y desventajas de los gestores de base de datos	113
Anexo #3: Ventajas y desventajas de las Metodologías de desarrollo de software	114
Anexo #4: Estructuras de Control	115
Anexo #5: Estructura Switch	115
Anexo #6: Llamadas de Funciones	116
Anexo #7: Definición de Funciones	116
Anexo #8: Convención de Nombres de Variables	117
Anexo #9: Convención de Nombres de Funciones	117
Anexo #10: Convención de Nombres de Clases	117
GLOSARIO DE TERMINOS	118

INTRODUCCIÓN

La actual revolución científico técnica y el proceso de globalización que están teniendo lugar en el mundo contemporáneo, conjuntamente con la alta competitividad e incertidumbre de las economías, exige cada vez más en el desarrollo de la informática y el auge del conocimiento, como factores decisivos en la búsqueda de soluciones alternativas, tanto para la economía global como local, así como para la sociedad en general.

Cuba propone la utilización de la Informática de forma justa y racional sobre principios éticos sostenibles. Identificándose desde muy temprano la ventaja y necesidad de dominar e introducir en la práctica social las Tecnologías de la Informática y las Comunicaciones (TIC), informatizando una amplia gama de esferas en nuestra sociedad para elevar la calidad de vida y nivel cultural de las personas.

Dentro de todos estos procesos se encuentra la Empresa de ETECSA, y en particular su Dirección Territorial en Granma, que se encuentra enfrascada en el proceso de migración a Software Libre. En consecuencia, se presentan incompatibilidades de aplicaciones antes usadas.

En aras de erradicar esta disyuntiva la Unidad de Gestión y Administración de Aplicaciones (UGAA) de ETECSA en la provincia Granma, se ha dado a la tarea de automatizar algunos procesos para agilizar el trabajo. Uno de estos es la gestión y control de las interrupciones de software que ocurren en el territorio.

Actualmente el proceso de gestión y control de las interrupciones se realiza de forma no automatizada, el trabajador afectado realiza una llamada telefónica o se presenta a la UGAA para dar a conocer el problema que presenta, aquí el jefe de la UGAA recoge los datos de la interrupción brindados por dicho trabajador en la planilla de control y asigna el responsable de atención para que dé solución al problema, este procede a solucionar la interrupción, actualiza la planilla con los datos que debe actualizar y notifica la atención al afectado por vía telefónica.

Al realizar el proceso de esta forma no es posible registrar más de una interrupción a la misma vez en tiempo real, y en caso de que los responsables no se encuentren no se podrá registrar la interrupción, se mantienen ocupadas por largos periodos de tiempo las líneas telefónica, se consume gran cantidad de materia prima en la impresión de planillas de control y se torna engorroso la recopilación de información de dichos documentos. En fin se retrasa la labor de todos los trabajadores involucrados en todo este proceso.

Teniendo en cuenta esta situación se plantea el siguiente **problema a resolver**:

¿Cómo lograr agilizar y organizar todo el proceso de Gestión y Control de las Interrupciones de Software en la Dirección Territorial de ETECSA en Granma usando un sistema automatizado?

Dado este problema surge como **Objeto de Estudio**: Estudio de los sistemas de Gestión de Interrupciones de Software. Y el **Campo de Acción**: Sistema de Gestión de Interrupciones de Software en la Dirección Territorial de ETECSA en Granma.

El **objetivo general es**: Desarrollar un sistema que permita gestionar y controlar las interrupciones de software para la Dirección Territorial de ETECSA en Granma.

A partir del análisis del objetivo general se derivaron los siguientes **objetivos específicos**:

1. Analizar situación actual del proceso de reporte de interrupciones de software en la Dirección Territorial de ETECSA en la provincia Granma.
2. Capturar los requisitos del cliente.
3. Realizar análisis y diseño del sistema para la Gestión y Control de Interrupciones de Software.
4. Implementar el sistema para la Gestión y Control de Interrupciones de Software.
5. Realizar el análisis de factibilidad del proyecto.

Para guiar esta investigación se plantean las siguientes **preguntas científicas**:

¿Cuáles son los fundamentos teóricos que respaldan los sistemas de gestión de interrupciones de software?

¿Cuáles son las tecnologías y herramientas de software utilizadas para la construcción de

una aplicación web?

¿Cómo implementar una aplicación web que responda a las necesidades existentes en el proceso de gestión de interrupciones de software en la Dirección Territorial de ETECSA en Granma?

Para dar cumplimiento a los objetivos específicos se llevarán a cabo las siguientes **tareas**:

- 1- Investigar con los trabajadores de esta empresa como realizan el proceso de gestión y control de reportes de interrupciones.
- 2- Análisis de las tendencias y las tecnologías existentes para la gestión y control de interrupciones de software.
- 3- Análisis y comparación de las metodologías existentes y selección de la más factible para el análisis y diseño de aplicaciones con tecnología Web.
- 4- Análisis, comparación y selección de las herramientas existentes para la implementación de aplicaciones con tecnología Web.
- 5- Modelación de análisis y diseño del sistema con las metodologías y herramientas seleccionadas.
- 6- Implementar el sistema con las herramientas seleccionadas.

Como **métodos científicos** se manejaron los métodos teóricos y los empíricos.

Como métodos teóricos se empleo el método analítico – sintético, logrando con este una mejor comprensión del trabajo, dividiéndolo en partes para su solución e integrando estas para lograr el resultado final.

Otro método seleccionado es el método histórico – lógico ya que partiendo de la observación y análisis de los antecedentes del objeto de estudio se dio inicio a esta

investigación.

Dentro de los métodos empíricos usados se encuentra la observación para conocer el avance y desarrollo del sistema.

Como método particular se usó la entrevista para conocer exactamente cuáles eran los requisitos necesarios para desarrollar la investigación, así como, el funcionamiento del negocio.

La investigación está estructurada en cuatro capítulos: La **Fundamentación Teórica** contiene un marco conceptual asociado a la información referente de las tecnologías, herramientas y metodologías unido a un estudio comparativo entre ellas para su selección, brindando un mejor soporte al desarrollo. Las **Características del Sistema** ofrecen un entendimiento de la organización del proceso y las necesidades esenciales que presenta el cliente. Se estructura el **Análisis y Diseño del Sistema** a través de diagramas que describen de forma entendible el funcionamiento del sistema y como se organiza la información manejada. En la **Implementación** se anexa un estudio de factibilidad del producto con una descripción de cómo está implementado el sistema. Culminando este documento se muestran las conclusiones generales, las recomendaciones, citas bibliográficas hechas a lo largo del trabajo, así como, un glosario de términos, siglas utilizadas y anexos.

CAPITULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción.

El siguiente capítulo aborda la fundamentación teórica acerca del proceso de gestión y control de las interrupciones de software en la Dirección Territorial de ETECSA en Granma, en este se realiza un estudio de los sistemas similares existentes, además se presentan las tendencias y tecnologías actuales y se realiza una selección de aquellas que serán utilizadas durante el desarrollo del proyecto.

1.2 Sistemas de Gestión de Interrupciones de Software

1.2.1 Sistemas de Gestión de Interrupciones de Software Internacionales

El **Sistema de Reporte de Interrupciones de Servicios Públicos de Telecomunicaciones** desarrollado por **OSIPTEL** tiene por objetivo facilitar a las empresas el reporte de interrupciones de servicios de telecomunicaciones a través de una interface Web. Dentro de sus funcionalidades se encuentran:

- Acceso al Sistema (inicio de sesión).
- Visualización de los servicios registrados para reporte de interrupción.
- Cambio de contraseña de acceso al sistema.
- Reporte de interrupción.
- Modificación de parámetros del reporte de interrupción.
- Adición de acreditaciones de la interrupción (archivos informáticos).
- Solicitud de prórroga de entrega de acreditación.
- Visualización de las interrupciones reportadas.
- Finalizar sesión.

De este sistema tomamos para el análisis la funcionalidad “Reporte de Interrupción” llegando a la conclusión de que no cumple con las expectativas de nuestro cliente. (1)

1.2.2 Sistema de Gestión de Interrupciones de Software en Cuba

AvilaQuid v4.2.0.30 Gestión de Incidencias:

Es una aplicación Web desarrollada por **Desoft S.A** destinada a la Atención a Clientes, que permite la automatización de la gestión de incidencias, a través de su seguimiento en una entidad o grupo de entidades interconectadas en una Intranet, a partir de que sean generadas tanto desde clientes externos como internamente.

He aquí sus principales cualidades:

- Permite abarcar todos los pasos de la gestión de la incidencia desde su registro, clasificación por diversos aspectos, asignación de responsabilidades, y procesamiento que posibilita describir la conclusión y la respuesta, llegando hasta la supervisión y la posible aplicación de medidas.
- Agrupa las incidencias según las Unidades Organizativas que intervienen en su tramitación, permitiendo el acceso pleno a la información que fluye con inmediatez.
- Es posible obtener el expediente de cada incidencia, donde se pueden conocer todos sus detalles y la trayectoria que ha seguido.
- Permite evaluar la satisfacción del cliente y la caracterización de los servicios brindados a éste, a partir de conocer de dónde procede cada incidencia.
- Facilita la búsqueda de incidencias según diferentes criterios y permite trabajar con subconjuntos de ellos basado en los resultados de las búsquedas.
- Se obtienen informes estadísticos que permiten evaluar resultados generales y específicos de la información.
- Notificación al Localizador (Beeper o Paging) y al Correo Electrónico del usuario responsable de la gestión y a su Unidad Organizativa.

El acceso a cada una de las opciones está restringido según los tipos de usuarios, definidos para cada una de las funciones posibles en el sistema (registrador, clasificador, procesador, supervisor, controlador, consultor, administrador, administrador principal). Esto unido a la agrupación por Unidades Organizativas proporciona una total fiabilidad. (2 - 4)

SisMon Sistema automatizado para el monitoreo de Servicios de Radiodifusión desarrollado por **LACETEL®**, Instituto de Investigación y Desarrollo de Comunicaciones, Cuba. Tiene como objetivo:

SisMon, automatiza el registro de la información sobre las tareas que realiza el técnico explorador, relacionadas con la exploración del espectro radioeléctrico y el registro de todos los datos sobre las transmisiones de los centros emisores remotos. Además, permite el registro y procesa de forma automatizada los datos introducidos sobre reportes realizados por diversos eventos y brinda información estadística para todos los usos.

Dentro de este sistema existe el bloque de “Reporte de eventos” el cual registra toda la información de la rotura y reparación de todos los equipos, ya sea la fecha y hora, así como la causa que provocó la rotura y quién es el técnico que reporta la misma. Cuando el equipo queda reparado, se registran todos los datos necesarios para posteriormente obtener un conjunto de salidas estadísticas relacionadas con esta actividad.

Y el bloque de “Salidas estadísticas” que está compuesto por un conjunto de opciones que permiten conocer las interrupciones que han ocurrido en un centro emisor o en un equipo en particular, especificando el período en que desee el resultado. (5)

1.3 Tendencias y Tecnologías Actuales

1.3.1 Aplicaciones Web

Una aplicación Web es aquella que los usuarios usan accediendo a un servidor Web a través de la red (Internet o intranet). Las aplicaciones Web son populares debido a la practicidad del navegador como cliente ligero. La habilidad para actualizar y mantener aplicaciones Web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su popularidad. Las interfaces de las páginas Web poseen ciertas limitantes en la funcionalidad del cliente. Métodos comunes en las aplicaciones de escritorio como dibujar en la pantalla, arrastrar o soltar no están soportados por las tecnologías Web estándar. Los desarrolladores utilizan generalmente lenguajes interpretados que se ejecutan en el cliente Web para obtener una mayor funcionalidad, así como tecnologías que se ejecutan en el servidor para no tener que recargar la página en su totalidad, algo que molesta mucho a los usuarios. (6)

1.3.2 Arquitectura Cliente-Servidor.

La arquitectura cliente-servidor es una relación entre procesos corriendo en máquinas separadas. El servidor es un proveedor de servicios. El cliente es un consumidor de servicios. Cliente y servidor interactúan por un mecanismo de pasaje de mensajes, el cliente realiza un "Pedido de servicio" y a su vez el servidor da una "Respuesta". Esta arquitectura es una forma de dividir las responsabilidades de un Sistema Informático separando la interfaz de usuario (Nivel de presentación) de la gestión de la información (Nivel de gestión de datos). (7)

1.3.3 Técnicas y tecnologías del lado del cliente.

Las técnicas y tecnologías del lado del cliente son aquellas que se utilizan para desarrollar los programas que se ejecutaran en los mismos, entiéndase por cliente a los navegadores o browser como: Firefox, Internet Explorer, Safari, Konqueror, Opera, Netscape Navigator etc. Estos clientes son los encargados de visualizar la información que han solicitado las personas, dicha información el navegador es capaz de leerla debido a que este interpreta código HTML, y algunos lenguajes script como son VBScript, Java Script. (8)

1.3.3.1 Lenguajes del lado del cliente.

El Cliente es un ordenador que accede a recursos y servicios brindados por otro llamado Servidor, generalmente en forma remota. Los lenguajes de lado cliente (entre los cuales no sólo se encuentra el HTML sino también el Java y el JavaScript los cuales son simplemente incluidos en el código HTML) son aquellos que pueden ser directamente "digeridos" por el navegador y no necesitan un pretratamiento. (8)

HTML (HyperText Markup Language)

Es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces (hipervínculos) que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia (gráficos, sonido, entre otros). Este lenguaje es el que se utiliza para presentar

información en la World Wide Web. La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc.), así como los diferentes efectos que se quieren dar (cursiva, negrita, o un gráfico determinado) y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado. (9 - 10)

XML (Extensible Markup Language).

XML es un lenguaje de marco que ofrece un formato para la descripción de datos estructurados, esto facilita una organización más precisa de los contenidos y unos resultados de búsquedas más significativos en varias plataformas. Además, XML habilita una nueva generación de aplicaciones para ver y manipular datos basadas en la Web. Se trata de una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil. (11)

Hojas de Estilo en Cascada (Cascading Style Sheets).

Son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). La virtud del desarrollo de CSS es separar la estructura de un documento de su presentación. La información de estilo puede ser adjuntada tanto como un documento separado o en el mismo documento HTML. En este último podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "style". Las ventajas de utilizar CSS son:

- Control centralizado de la presentación de un sitio Web completo con lo que se agiliza de forma considerable la actualización de la misma.
- Los navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio Web remoto. Así por ejemplo, personas con deficiencias visuales pueden configurar su propia hoja de estilo para aumentar el tamaño del texto o remarcar más los enlaces.

- El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño. (12)

1.3.4 Tecnologías del lado del servidor.

Las tecnologías del lado del servidor son aquellos programas o servicios que corren en un servidor remoto y que brindan funcionalidades al sistema. (13)

1.3.4.1 Lenguajes de Programación

PHP

Es un lenguaje de programación usado generalmente para la creación de contenido para sitios Web. PHP es un acrónimo recurrente que significa "PHP Hypertext Pre-processor" (inicialmente PHP Tools, o, Personal Home Page Tools), y se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web. (14 -18). [\(Ver Anexo#1\)](#)

Python

Es un lenguaje interpretado, lo que ahorra un tiempo considerable en el desarrollo de programas, pues no es necesario compilar ni enlazar con librerías que usen la memoria de la computadora. El intérprete se puede utilizar de modo interactivo, lo que facilita experimentar con características del lenguaje, escribir programas desechables o probar funciones durante el desarrollo del programa. (19,20) [\(Ver Anexo#1\)](#)

Perl

Es un lenguaje de propósito general, inicialmente fue desarrollado para la manipulación de texto, pero actualmente es utilizado para el desarrollo de varias tareas, entre las que se encuentran administración de sistemas, desarrollo Web, programación en red y desarrollo de GUI. Es fácil de usar, eficiente y completo. Soporta tanto la programación estructurada

como la programación orientada a objetos y la programación funcional, tiene incorporado un poderoso sistema de procesamiento de texto y una enorme colección de módulos disponibles. (21) ([Ver Anexo#1](#))

¿Por qué PHP?

Para el desarrollo de la aplicación Web se debe tener en cuenta cuestiones esenciales como: garantizar que el sistema sea multiplataforma y que pueda ser desarrollado en software libre por las características del país, ya que este está bloqueado y de las restricciones de las principales compañías de desarrollo informático. Analizando esto el más adecuado es el PHP por ser libre, multiplataforma, su flexibilidad de comunicación con los principales gestores de bases de datos y por su potencialidad en funcionalidades y rapidez. Ventajas de su uso:

- Es un lenguaje multiplataforma y permite utilizar programación Orientada a Objetos.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad como son: InterBase, mSQL, MySQL, Oracle, Informix, PostgreSQL.
- Permite leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.
- Es un lenguaje libre de licencias de uso, por lo que se presenta como una alternativa de fácil acceso para todos los desarrolladores.
- Permite crear los formularios para la web.
- Biblioteca nativa de funciones sumamente amplia que permiten que el desarrollador haga casi cualquier cosa desde generar documentos en .pdf hasta analizar código XML.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

1.3.5 Gestor de Base de Datos.

Los Sistemas de gestión de base de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. En los textos que tratan este tema, o temas relacionados, se mencionan los términos SGBD y DBMS, siendo ambos equivalentes y acrónimos respectivamente, de Sistema Gestor de Bases de Datos y DataBase Management System, por sus siglas en inglés. (22-25)

PostgreSQL

Servidor de base de datos relacional libre, liberado bajo la licencia BSD. Es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL, Firebird y MaxDB), así como sistemas propietarios como Oracle o DB2. (26) [\(Ver Anexo#2\)](#)

MySQL

Es un sistema de gestión de base de datos relacional, multihilo y multiusuario desarrollado como software libre, es el sistema de base de datos open source más utilizado. Dentro de las características más importantes de este software de base de datos están:

- Es desarrollado en C y C++.
- Funciona en diferentes plataformas.
- Es compatible con lenguajes de programación como C, C++, Java, Perl, PHP, Python.
- Proporciona sistemas de almacenamientos transaccionales y no transaccionales.
- Las funciones SQL están implementadas usando una librería altamente optimizada y debe ser tan rápida como sea posible. (27) [\(Ver Anexo#2\)](#)

1.3.6 Servidor de aplicaciones Web

Apache:

El servidor HTTP Apache es un software gratuito de código fuente abierto que funciona sobre una multitud de sistemas operativos como: Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementan el protocolo HTTP/1.1. Es un servidor altamente configurable de diseño modular, es muy sencillo ampliar sus capacidades. Actualmente existen muchos módulos para Apache que son adaptables a este. Apache trabaja con PHP y otros lenguajes de script. También trabaja con Java. Teniendo todo el soporte que se necesita para tener páginas dinámicas. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Se escoge este servidor por políticas de la institución. (28-30)

Internet Information Server (IIS):

Es una serie de servicios para los ordenadores que funcionan con Windows. Originalmente era parte del *Option Pack* para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS. (31)

1.3.7 Metodologías de desarrollo de software.

Todo desarrollo de software se torna riesgoso y difícil de controlar, pero si no se utiliza una metodología, existe una mayor exposición a obtener clientes y desarrolladores insatisfechos con el resultado. Muchas veces no se toma en cuenta la utilización de una metodología adecuada y por esta causa se desarrolla el software solo tomando en cuenta los requerimientos del cliente de manera que en la etapa final si este solicita algún cambio en los requerimientos se hace muy difícil de realizar y es justo éste, uno de los factores que ocasiona un atraso en el proyecto y por tanto la incomodidad del desarrollador por no cumplir con el cambio solicitado y el malestar por parte del cliente por no tomar en cuenta su pedido. Obviamente para evitar estos incidentes se debe llegar a un acuerdo formal con el cliente, al inicio del proyecto, de tal manera que cada cambio o modificación no

perjudique el desarrollo del mismo, así como escoger una metodología adecuada para el desarrollo del software que sirva como guía para realizar de forma disciplinada y eficiente el producto deseado. (32)

1.3.7.1 Programación Extrema (XP).

Es un enfoque de la ingeniería de software que pone más énfasis en la adaptabilidad que en la previsibilidad. Ésta metodología promueve la idea de que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos. Las características fundamentales de esta metodología son: (33) [\(Ver Anexo#3\)](#)

- Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- Pruebas unitarias continuas.
- Programación en parejas.
- Frecuente integración del equipo de programación con el cliente.
- Corrección de todos los errores.
- Refactorización del código.
- Propiedad del código compartida.
- Simplicidad.

1.3.7.2 El Proceso Unificado de Modelado (RUP).

La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, divide en 4 fases el desarrollo del software:

- 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 llegar a obtener la capacidad

operacional inicial.

- Transición: El objetivo es llegar a obtener el release del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

(34,35) [\(Ver Anexo#3\)](#)

El ciclo de vida que se desarrolla por cada iteración, es llevado bajo dos disciplinas:

Disciplina de Desarrollo:

- Ingeniería de Negocios: Entendiendo las necesidades del negocio.
- Requerimientos: Traslado de las necesidades del negocio a un sistema automatizado.
- Análisis y Diseño: Traslado de los requerimientos dentro de la arquitectura de software.
- Implementación: Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- Pruebas: Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.

Disciplina de Soporte:

- Configuración y administración del cambio: Guardando todas las versiones del proyecto.
- Administrando el proyecto: Administrando horarios y recursos.
- Ambiente: Administrando el ambiente de desarrollo.
- Distribución: Hacer todo lo necesario para la salida del proyecto.

Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que cada una se convierte luego en un entregable al cliente. Esto trae como beneficio la retroalimentación que se tendría en cada entregable o en cada iteración.

Los elementos del RUP son:

- **Actividades:** Son los procesos que se llegan a determinar en cada iteración.
- **Trabajadores:** Vienen hacer las personas o gentes involucradas en cada proceso.
- **Artefactos:** Un artefacto puede ser un documento, un modelo, o un elemento de modelo.

Características del Proceso Unificado.

Dirigido por casos de uso:

Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. Un caso de uso es un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante. Representan requerimientos funcionales. Los casos de uso guían la arquitectura del sistema y la arquitectura del sistema influye en la selección de los casos de uso.

Centrado en la arquitectura:

La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura. La arquitectura se refleja en los casos de uso pues cada producto tiene tanto una función como una forma, ninguna es suficiente por sí sola.

Iterativo e Incremental:

RUP propone dividir el trabajo en partes más pequeñas o mini proyectos, donde cada mini proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto. Cada iteración se realiza de forma planificada es por eso que se dice que son miniproyectos. Una iteración es una secuencia de actividades con un plan establecido y criterios de evaluación las cuales están estrechamente relacionadas, cuyo resultado es una versión del software.

Beneficios de la iteración:

- Reduce el coste del riesgo al coste de un solo incremento.
- Menos riesgo de no sacar el producto al mercado en fecha.
- Acelera el ritmo de desarrollo.
- Las necesidades del usuario y correspondientes requisitos no pueden definirse completamente al principio. Se requieren iteraciones sucesivas.

1.3.8 Herramientas CASE.

La tecnología CASE supone la automatización del desarrollo del software, contribuyendo a mejorar la calidad y la productividad en el desarrollo de sistemas de información y se plantean los siguientes objetivos: (36)

- Permitir la aplicación práctica de metodologías estructuradas, las cuales al ser realizadas con una herramienta se consigue agilizar el trabajo.
- Facilitar la realización de prototipos y el desarrollo conjunto de aplicaciones.
- Simplificar el mantenimiento de los programas.
- Mejorar y estandarizar la documentación.
- Aumentar la portabilidad de las aplicaciones.

- Facilitar la reutilización de componentes software.
- Permitir un desarrollo y un refinamiento visual de las aplicaciones, mediante la utilización de gráficos.

Automatizar:

- El desarrollo del software.
- La documentación.
- La generación del código.
- El chequeo de errores.
- La gestión del proyecto.

Permitir:

- La reutilización del software.
- La portabilidad del software.
- La estandarización de la documentación.

Visual Paradigm

Es una herramienta CASE que ofrece un entorno de creación de diagramas para UML, diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad; uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación; capacidades de ingeniería directa (versión profesional) e inversa; modelo y código que permanece sincronizado en todo el ciclo de desarrollo; disponibilidad de múltiples versiones, para cada necesidad; disponibilidad de integrarse en los principales IDEs; disponibilidad en múltiples plataformas, y muy útil para la generación de código fuente en PHP, también con el Paradigm se generan script de las tablas de salidas para las clases persistentes. (37-40)

Beneficios:

- Navegación intuitiva entre el modelo visual y el código.
- Poderosa herramienta de generación de PDF/HTML a partir de diagramas UML.
- Sincronización entre el código fuente y el modelo en tiempo real o bajo demanda.
- Entorno visual de modelado superior.
- Soporte para toda la notación UML.
- Sofisticados y automáticos diagramas de capas.
- Análisis de textos.
- Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la
- comunicación.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de integrarse en los principales IDEs.
- Disponibilidad en múltiples plataformas.

1.3.9 Herramientas de Desarrollo.

Komodo

Es un IDE multilenguaje (Perl, PHP, Python, Ruby, Tcl y XSLT) y multiplataforma (Windows, Mac OS X, Solaris y Linux). Dentro de sus principales características está el completamiento de código, depurador integrado, resumen de funciones (lo que algunos conocen como “outline”), funciones potentes de búsqueda y generador de expresiones regulares. (41)

Zend Studio

Es una herramienta de desarrollo utilizada para facilitar la programación web. Es propietario, compatible con las plataformas Linux, MAC y Windows. Incluye todos los componentes necesarios durante el ciclo de vida de una aplicación en PHP. Incluye editor, análisis, depuración, optimizadores de código y herramientas de base de datos. Zend Studio permite agilizar el desarrollo web y simplificar proyectos complejos.

Entre sus principales características se puede citar que cuenta con un excelente completamiento de código. Contiene coloreado en la sintaxis del código, administración avanzada de proyectos, múltiples lenguajes, incorpora el Framework de Zend, PHP Documenter, manual de PHP. Integración con subversión, los navegadores, integración avanzada con FTP. Soporte para Servicios Web, PHP4, PHP5 y SQL. (42,43)

Ventajas: Agiliza el trabajo, cuenta con un buen Depurador, infinitas opciones que permiten un desarrollo profesional de nuestras aplicaciones.

Desventajas: Requiere Licencia de pago, no incluye editor visual HTML, un poco complejo.

¿Por qué Zend Studio?

Teniendo en cuenta los argumentos antes expuestos y dado a que se va a hacer uso de tecnologías cliente/servidor sobre plataforma web, luego de haberse realizado un análisis de los principales editores de PHP más utilizados en la actualidad, se llega a la conclusión de que el más apropiado para el desarrollo del sistema es el Zend Studio, debido a su gran integración con el gestor de base de datos que se usará (MySQL), su multiplataforma, el completamiento de código que presenta, la integración a ayudas online y una gran herramienta de seguimiento de errores lo que lo hace muy robusto.

1.3.10 Sistema de Gestión de Contenido

Un CMS es un sistema de gestión de contenidos (Content Management Systems) que se utiliza principalmente para facilitar la gestión en la web, ya sea en Internet o en una

intranet, y por eso también son conocidos como gestores de contenido web (Web Content Management o WCM), constituyen una herramienta que permite a compañías o individuos crear y modificar el contenido de su página web, con poco o nada de conocimiento técnico. (44)

Principales sistemas de gestión de contenidos del mercado

Los CMS de código abierto más usados pueden dividirse en dos grandes grupos: los desarrollados bajo la plataforma J2EE, y los desarrollados sobre la plataforma PHP. Dentro de los de la plataforma J2EE sobresalen el EXOPlatform, el Apache Lenya, y el LifeRay; mientras que en la plataforma PHP encabezan la lista Mambo, Drupal, XOOPS, Postnuke, PHP-Nuke, Plone, entre otros. Existen otros CMS también populares desarrollados en Pyton, Perl, y otros lenguajes. (44)

Entre algunos de los CMS más utilizados sobre la plataforma PHP se pueden encontrar:

Joomla:

Joomla es uno de los más potentes CMS del planeta. Es usado en todo el mundo para todo, desde los sitios web más simples hasta las más complejas aplicaciones corporativas. Joomla es muy sencillo de instalar, simple de administrar y además es confiable.

Incluye características como: hacer caché de páginas para mejorar el rendimiento, indexamiento web, versiones imprimibles de páginas, flash con noticias, etc. La primera versión de Joomla fue publicada en septiembre de 2005. Se trataba de una versión mejorada de Mambo 4.5.2.3 combinada con otras modificaciones de seguridad y anti-bugs. Actualmente los programadores han publicado Joomla 1.5 bajo un código completamente reescrito y construido bajo PHP 5. (45)

Plone:

Es un CMS muy flexible y poderoso, que presenta una excelente interfaz de usuario, una instalación muy limpia, un impresionante grado de personalización, y una integración con

LDAP u otros sistemas de login. Posibilita además construir un sitio cuyo contenido se puede enriquecer constantemente, debido a sus facilidades de actualización e interacción, y con ello, obtener ventajas competitivas. Pero, probablemente su característica más importante es que es un sistema de código abierto y libre.

Plone es un sistema de gestión de contenidos basado en Zope⁴⁵ y programado en Python, que está basado en código abierto. Puede utilizarse como servidor intranet o extranet, un sistema de publicación de documentos y una herramienta de trabajo en grupo para colaborar entre entidades distantes. Es un CMS muy reciente lo que no lo hace menos portable y poderoso. Presenta características similares al resto de los CMS como completamente modularizado donde los módulos se pueden gestionar, personalización extensible para los principales elementos de seguridad, gestión de usuarios y además tiene un sistema de permisos de grupo de gran alcance y de uso fácil. (46)

Drupal

Drupal es un sistema de gestión de contenido para sitios web. Permite publicar artículos, imágenes, u otros archivos y servicios añadidos como foros, encuestas, votaciones, blogging, administración de usuarios y permisos basados en PHP. El contenido textual de las páginas y otras configuraciones son almacenados en una base de datos y se editan utilizando un entorno web incluido en el producto.

Lo que hace especial a Drupal es su modularidad, se elige qué es lo que se quiere que la web tenga, estos módulos ofrecen un amplio conjunto de funciones, incluyendo sistemas de comercio electrónico, galerías de fotos, administración de listas de correo electrónico, etc. El punto que hace de Drupal interesante, es que en sus requerimientos utiliza un sistema LAMP normal (Linux+Apache+MySQL+PHP). Esas son a priori las grandes ventajas que tiene Drupal sobre el resto de los CMS.

Además introduce el concepto de nodo como sinónimo de tipos de contenido, cualquier recurso que se ingrese al sistema pasa a ser un nodo, que puede ser variable e incluir artículos, historias, posts, encuestas, imágenes, libros colaborativos, reseñas, recetas, etc. Este nuevo concepto nos permite estandarizar la información asignándoles las mismas características a distintos tipos de objetos y la posibilidad de tener toda la información centralizada y a la vez catalogada.

El sistema de taxonomía de Drupal es robusto y potente, el contenido se organiza en categorías que se arman a través del módulo taxonomías, el cual permite generar vocabularios controlados con términos que pueden ordenarse jerárquicamente y asociarse a un tipo de nodo en particular si hiciera falta. (47)

1.4 Conclusiones.

En este capítulo se detallaron las condiciones y problemas que rodean el objeto de estudio; y a través de los conceptos y definiciones planteadas, se determinaron las condiciones específicas que envuelven al mismo. Se analizaron las características de diferentes herramientas, para la creación de un software o aplicación así como algunas metodologías. Después de este análisis y la fundamentación realizada, el lenguaje de programación que se utilizará es PHP, el gestor de base de datos, el gestor de contenido CMS y el servidor de aplicaciones se rige por las políticas de la empresa, seleccionándose MySQL, Joomla y Apache respectivamente. Se empleará la metodología RUP por ser un proceso iterativo e incremental, que permite dirigir el proceso por casos de uso y tener un mejor control de la calidad en cada etapa del proceso, como lenguaje de modelado UML y la herramienta que se usará como entorno de creación de diagramas para UML el Visual Paradigm.

Una vez conocidas las herramientas óptimas y los conceptos a utilizar se puede empezar a desarrollar la propuesta de sistema.

CAPITULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción.

En el presente capítulo, se dará una visión de lo que funcionalmente debe realizar el sistema. Se transitará a lo largo de la fase de inicio por cada uno de los flujos de trabajo desde el negocio hasta el sistema. Todo esto se desarrollará por medio de un conjunto de artefactos y modelos resultantes de la aplicación de la metodología de desarrollo de software. Estos modelos deben cumplir una serie de propiedades entre las que se encuentran la de ser coherentes y que tengan relación con el resultado final del producto. Uno de los modelos útiles previo al desarrollo de un software es el modelo del negocio.

2.2 Objeto de estudio

2.2.1 Análisis de la Situación Problemática

Muchos son los procesos, que en las empresas, se han automatizado con el objetivo de optimizar actividades que van desde las más sencillas hasta las más complejas, logrando con ello facilitar el trabajo de varias personas. La UGAA de la Dirección Territorial ETECSA en Granma también está llevando a cabo este proceso de informatización y el sistema que da título a este trabajo es un ejemplo de ello.

- En la actualidad, la Dirección Territorial de ETECSA en Granma no cuenta con un sistema automatizado que permita la gestión y control de las interrupciones de software que se producen en el territorio.
- El trabajador afectado no cuenta con un sistema automatizado para hacer el reporte de interrupciones de software.
- El jefe de la UGAA no cuenta con un software para la obtención rápida y segura de la información relacionada con este proceso.
- Utilización de las líneas telefónicas por largo periodo de tiempo.

- Se consume gran cantidad de combustible para el traslado de los medios afectados a la UGAA.
- Gasto de recursos para la impresión de “Planillas de Control”.

Al realizarse la actividad de reporte de interrupción de forma no automatizada trae consigo que ocurra, la pérdida de la información, errores, lentitud y poca eficiencia. Por la importancia que tiene este proceso surge la necesidad de crear un sistema automatizado para la gestión de toda la información que en él se maneja.

2.2.2 Objeto de automatización

Al realizar este trabajo se podrán ejecutar de forma automatizadas ciertas actividades que actualmente se realizan de forma manual, como es el caso de la gestión y control de toda la información relacionada con las interrupciones de software, las de los trabajadores responsables en la Dirección Territorial de ETECSA en Granma y la información necesaria de los software que se utilizan dentro de esta institución. Además el proceso de atención y recepción de las interrupciones podrá ser realizado por los trabajadores responsables desde cualquier parte de la provincia y así lograr un tiempo de respuesta mucho menor.

La obtención rápida y segura de los datos referentes al estado de las interrupciones en tiempo real brindará la posibilidad de elaborar reportes de acuerdo a los criterios que se deseen.

Otra actividad a automatizar es la asignación de responsables a las interrupciones y la notificación de diferentes acciones por vía e-mail.

2.2.3 Información que se maneja

Durante la realización del proceso de gestión de las interrupciones de software que se producen en la Dirección Territorial de ETECSA en la provincia Granma, es necesario contar con información referente de los trabajadores de dicha institución como su nombre y apellidos, usuario, el área a la que pertenece y el medio en cual trabaja.

Se manejan las siguientes entidades:

Planilla de Control de Interrupciones (Interrupción).

Datos de los trabajadores responsables (Responsable).

Datos de los software que se utilizan (Software).

Esta planilla está conformada por el nombre del trabajador afectado, el medio donde se produjo la interrupción (IP, Mascara de Subred), fecha del reporte, nombre del responsable, fecha de atención, nombre del software, área del reporte, descripción de la interrupción y estado. De los trabajadores responsables se registra su nombre y usuario del dominio. Y de los software si su identificador y versión.

2.3 Propuesta del sistema.

Para Informatizar el proceso de gestión y control de las interrupciones de software que se producen en la Dirección Territorial de ETECSA en la provincia Granma el sistema debe comenzar autenticando a los usuarios (Trabajadores Responsables, Afectado y Jefe de la UGAA) para tener acceso a las diferentes funcionalidades.

El sistema le permite al jefe de la UGAA gestionar la información de los responsables, software e interrupciones y almacenarla en una base datos para lograr una mejor organización y evitar la pérdida de información. Además le permite conocer el estado de las interrupciones en cuanto a criterios como intervalos de fechas, responsable, área y el software más involucrados en las interrupciones. El sistema asigna de forma automática un responsable a cada interrupción y le ofrece la posibilidad de cambiarlo en caso necesario.

Esta aplicación permite que todos los trabajadores afectados del territorio reporten simultáneamente las interrupciones. Le brinda la posibilidad de obtener una notificación en el momento del reporte y al ser atendida la interrupción.

El sistema le da a conocer al trabajador responsable las interrupciones que le son asignadas y la posibilidad de actualizar las mismas al ser atendidas.

El sistema permite la corrección de errores de forma rápida al manipular los datos.

Por la estandarización en la organización de la estructura que exige ETECSA en sus instituciones es posible que este sistema sea implantado en todas las Direcciones Territoriales del país, sin ningún tipo de inconveniente.

2.4 Modelado del Negocio.

El modelado del negocio es una técnica para comprender y definir los procesos del negocio, roles y responsabilidades de la organización en los modelos de casos de uso del negocio y de objetos. Los propósitos que se persiguen al modelar el negocio son:

- Comprender la dinámica de la organización en la cual se va a implementar el sistema.
- Comprender los problemas actuales de la organización e identificar las mejoras potenciales.
- Asegurar que los usuarios y desarrolladores tengan un entendimiento común de la organización.

Por tanto la finalidad del modelado del negocio es describir cada proceso del negocio especificando sus datos, actividades, roles y reglas del negocio. Este proceso está soportado por dos tipos de modelos de UML: modelos de casos de uso y modelo de objetos. Un modelo de casos de uso del negocio describe los procesos de una empresa en términos de casos de uso y actores del negocio. Un modelo de objetos del negocio es un modelo que describe cómo colaboran los trabajadores y las entidades del negocio dentro del flujo (realización). Para lograr un mejor entendimiento del modelado del negocio, se presenta una descripción de cuáles serán los actores y trabajadores del mismo.

2.5 Actor y Trabajadores del Negocio.

Un actor puede representar a una persona física, otro sistema, un dispositivo, siendo siempre un tercero fuera del sistema que colabora con él. La definición de actores, sirve para definir el contexto externo del sistema, esto es, delimitar los elementos que se encuentran fuera y dentro del mismo, y un actor juega un rol para cada caso de uso en el que colabora. .

Actores del Negocio	Justificación
Trabajadores Afectados	Es el encargado de reportar las interrupciones de software que presente.

Tabla 1: Descripción de los actores del negocio.

El trabajador, por su parte, representa una posición que se puede asignar a una persona o equipo en una organización de desarrollo de software, y para cada trabajador se especifican las responsabilidades y habilidades requeridas, ya que este define el comportamiento y las responsabilidades de un individuo.

Trabajadores del Negocio	Justificación
Jefe de la UGAA	Es el encargado de gestionar y controlar todo el proceso de las interrupciones.
Servidor LDAP	Encargado de verificar la autenticación de los usuarios.
Trabajador Responsable	Es el encargado de darle solución a las interrupciones.

Tabla 2: Descripción de los trabajadores del negocio.

2.6 Diagrama de Casos de Uso del Negocio.

Con el objetivo de obtener una visión general de los diferentes procesos del negocio se realiza el diagrama de casos de uso del negocio, donde cada proceso se representa como un caso de uso y solo aparecerán los actores del negocio correspondientes a los roles

externos al sistema este diagrama permite mostrar los límites y el entorno de la organización (Ver figura 1).

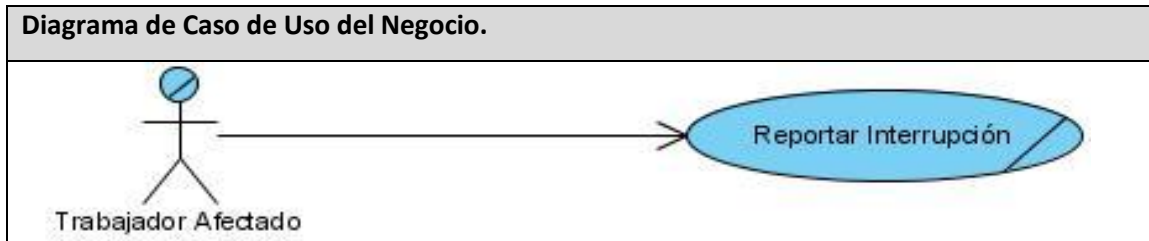


Figura 1 Diagrama de Caso de Uso del Negocio

2.7 Descripciones Textuales.

La descripción textual de los casos de uso del negocio se formaliza en un documento generalmente llamado “Especificación del caso de uso de negocio”. Para realizar la misma, inicialmente se rellena una plantilla de descripción, y después, a partir de la información reflejada en dicha plantilla, se construye un conjunto de diagramas (diagramas de actividades) que describen completamente el caso de uso del negocio.

2.7.1 Descripción Textual del CUN Reportar Interrupción.

Caso de Uso	Crear Interrupción	
Actores	Trabajador Afectado	
Resumen	El Caso de Uso comienza cuando el trabajador afectado se presenta a la UGAA o realiza una llamada telefónica al mismo para que se le registre el reporte. Luego de que el Jefe de la UGAA le asigna un responsable y registra los datos en la “Planilla de Control” este ultimo procede a la atención de la interrupción, al finalizar actualiza la planilla con los datos que requieran actualizar. Y le notifica al afectado por vía telefónica.	
Curso Normal de Eventos		
Acciones del Actor	Respuesta del Negocio	

1- Se presenta a la UGAA o realiza una llama telefónica al mismo con el fin de reportar la interrupción.	2- Se le piden los datos del reporte de la interrupción
3- Ofrece los datos de la interrupción.	4- El Jefe de la UGAA registra los datos en un documento "Planilla de Control" y le asigna un responsable luego este último procede a darle solución al problema.
	5- Luego actualiza la planilla con los datos restantes.
	6- Entrega la planilla actualizada al jefe de la UGAA
8- Obtiene notificación por vía telefónica. Finalizando así el caso de uso.	7- Y le notifica al afectado que ha sido atendido su reporte. Pudiendo quedar pendiente a solución o solucionado.
Curso Alterno de los Eventos	
	5- En caso de que la interrupción quede pendiente porque debe ser atendido por un responsable de hardware se traslada la planilla al dicho responsable y finaliza el caso de uso.

Tabla 3: Descripción textual del CUN Reportar Interrupción.

2.8 Diagramas de Actividades.

Con el objetivo de mostrar el flujo de trabajo que realiza cada proceso del negocio se utilizarán los diagramas de actividades que no son más que diagramas de flujos (paralelos o secuenciales) que muestran el flujo de control de una actividad a otra, donde los datos aparecen como objetos que fluyen entre las actividades como se muestra en las siguientes figuras.

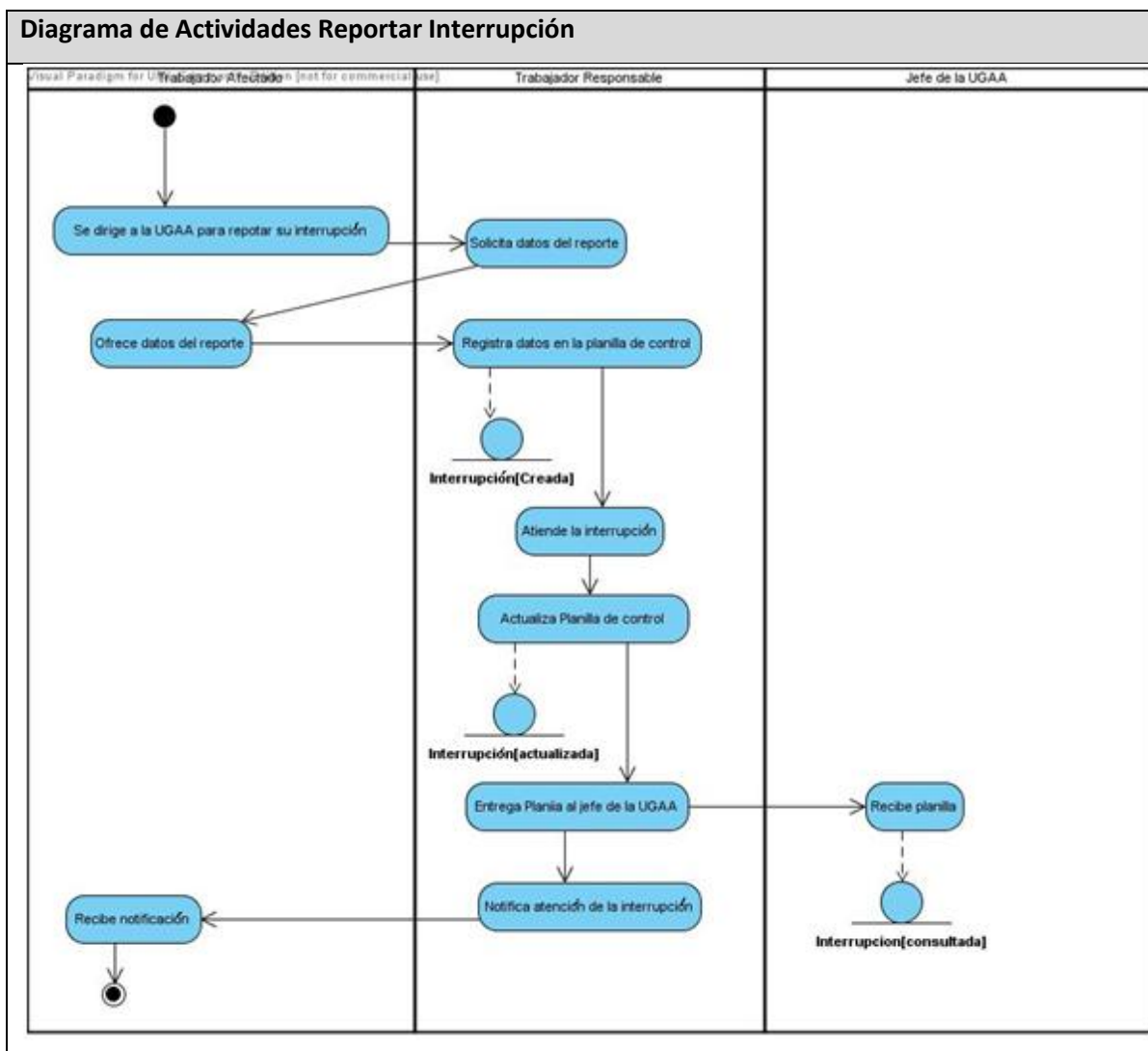


Figura 2: Diagrama de actividades Reportar Interrupción

2.9 Diagramas de Clases del Modelo de Objetos.

Este diagrama muestra la relación existente entre los trabajadores y las entidades del negocio y proporciona un acercamiento a la identificación de los futuros actores y entidades del sistema.

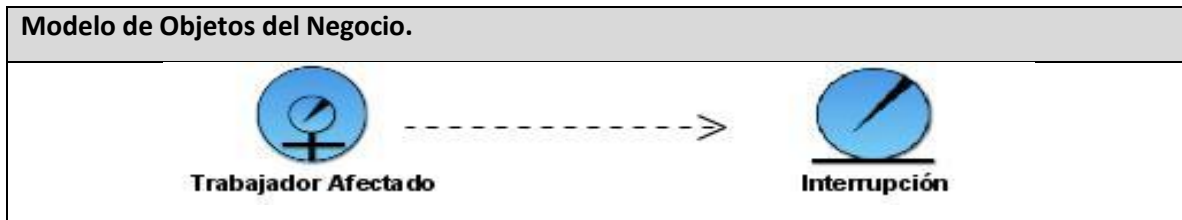


Figura 3: Modelo de Objetos de Negocio

2.10 Requerimientos.

Una de las principales tareas en el ciclo de desarrollo de un sistema es la de determinar los requerimientos del sistema de información (es decir, las condiciones o capacidades que el sistema debe cumplir). El propósito principal del flujo de trabajo de RUP captura de requisitos es guiar el desarrollo hacia el sistema correcto donde se describan con claridad y sin ambigüedades el comportamiento del mismo. Así como lograr una comunicación efectiva entre el cliente (incluyendo a los usuarios) y a los desarrolladores sobre qué debe y qué no debe hacer el sistema. Los requisitos se pueden clasificar en: funcionales y no funcionales.

2.10.1 Requisitos Funcionales.

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Su meta principal es identificar y documentar las acciones que en realidad debe ejecutar el sistema para que cumpla con los objetivos planteados al inicio de este trabajo.

Todas estas acciones se convierten en requisitos funcionales y de acuerdo a los objetivos propuestos el sistema debe ser capaz de:

RF 1 Autenticar.

RF 1.1 Autenticar usuarios: El sistema en conjunto con el servicio de identificación de usuarios autentica los mismos de acuerdo a los permisos que presenta.

RF 1.2 Cerrar Sesión.

RF 2 Gestionar Interrupciones.

RF 2.1 Insertar Interrupciones: Se registra la interrupción con los datos necesarios.

RF 2.2 Actualizar Interrupciones: Se modifica la interrupción con los datos necesarios.

RF 2.3 Mostrar información del Interrupciones: El sistema muestra toda la información de la interrupción.

RF 2.4 Eliminar Interrupciones: Se selecciona la interrupción deseada y se elimina de la base de datos.

RF 3 Reportar Interrupción: Los datos necesarios para realizar esta acción: id-interrupción, nombre del afectado, nombre del medio, IP, máscara de subred, nombre del software, nombre del responsable, estado, descripción, fecha del reporte, fecha de solución, área. Se verifican los datos introducidos en caso de estar correctos se registra la interrupción.

RF 4 Atender Interrupción: Los datos necesarios para realizar esta acción son: fecha de atención, estado. Se procede a actualizar la interrupción con estos nuevos valores.

RF 5 Asignar Responsable: Se selecciona la interrupción a la que se le desea asignar el nuevo responsable y se actualiza con el nuevo valor

RF 6 Obtener Reporte.

RF 6.1 Obtener Reporte por Fecha: Los datos necesarios para realizar esta acción son: fecha inicial y fecha final. El sistema verifica que el intervalo de fecha sea correcto en caso de estarlo muestra todas las interrupciones que se reportaron dentro de ese intervalo.

RF 6.2 Obtener Reporte por Responsable: Los datos necesarios para realizar esta acción son: nombre del responsable. Se verifica que existe dicho responsable y se

muestra todas las interrupciones que le han sido asignada.

RF 6.3 Obtener Reporte por Área: Dado el nombre del área se muestran todas las interrupciones que se reportaron desde dicha área.

RF 6.4 Obtener Software más involucrado en Interrupciones: Se muestran en orden los software que más se ven involucrados en el proceso de reportes de interrupciones.

RF 7 Gestionar Software:

RF 7.1 Insertar Software: Se registra el software con los datos necesarios.

RF 7.2 Eliminar Software: Se modifica el software con los datos necesarios.

RF 7.3 Mostrar Software: El sistema muestra toda la información del software.

RF 7.3 Editar Software: Se selecciona el software deseado y se elimina de la base de datos.

RF 8 Gestionar Responsable:

RF 8.1 Insertar Responsable: Se registra el responsable con los datos necesarios.

RF 8.2 Eliminar Responsable: Se modifica el responsable con los datos necesarios.

RF 8.3 Mostrar Responsable: El sistema muestra toda la información del responsable.

RF 8.3 Editar Responsable: Se selecciona el responsable deseado y se elimina de la base de datos.

2.10.2 Requisitos no Funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Los requerimientos no funcionales forman una parte significativa de la especificación. Son importantes para que clientes y usuarios

puedan valorar las características no funcionales del producto y de esta forma poder decir si el producto tiene la calidad requerida.

Eficiencia:

- Uso de Patrones de Diseño.
- Base de Datos normalizada.
- Gestor de Base de Datos MySQL.

Software:

Para la instalación del servidor:

- Servidor Apache versión 2 o superior.
- Servidor de bases de datos MySQL versión 4 o superior.
- Sistemas Operativos GNU/Linux (recomendado) o Microsoft Windows.

Para la interpretación por clientes (Navegadores Web):

- Internet Explorer v6 o superior.
- Mozilla Firefox 2.* (recomendado) o superior (recomendado).

Hardware:

- Mínimo Requerido: RAM 128 Mb, Disco duro 100 Mb.
- Mínimo Recomendado: RAM 256 Mb, Disco duro 1Gb.
- Conexión de red por MODEM o fibra óptica.

Apariencia o Interfaz Externa

- El diseño de la interfaz debe ser lo más sencillo y claro posible y contar con aspectos que definan claramente cada una de sus funcionalidades.
- La navegabilidad debe ser fácil y sencilla.

Usabilidad:

- El sistema podrá ser utilizado por cualquier usuario que acceda al mismo.
- Proporcionará rápido acceso de búsqueda de la información, en tiempos cortos.
- Contará con zonas de acceso limitado, a la cual solo accederán los usuarios

autorizados con roles determinados.

- Fácil gestión de la información

Rendimiento

- El sistema deberá ser capaz de gestionar toda la información y dar respuestas rápidas, casi inmediatas, a las solicitudes realizadas.
- Debe ser eficiente en la gestión de las solicitudes que se realicen, para que el usuario obtenga, sin mucha navegación, el resultado que desea.

Diseño e implementación:

- El sistema es recomendado que funcione sobre una plataforma de sistema operativo GNU/Linux a modo texto, basado en el modelo cliente/servidor. Los lenguajes necesarios para trabajar son HTML, JavaScript y PHP4 o superior. Como gestor de Base Datos se utilizará MySQL, se desarrollará sobre un sistema GNU/Linux, utilizándose herramientas como el ZendStudio, phpmyadmin, Visual Paradigm.

Portabilidad:

- Es un sistema multiplataforma. Podrá ser montado sobre los sistemas operativos: Unix, Linux, Windows.

Seguridad:

- Verificar la autenticidad y niveles de privilegio del usuario antes de hacer cualquier operación dentro del sistema.
- Verificar que las funcionalidades sean mostradas en correspondencia con los niveles de privilegios.
- Protección de acciones no autorizadas que puedan afectar la integridad de los datos.

Confidencialidad:

- Solo se permitirá conexiones mediante usuarios del directorio activo.

Soporte:

- El sistema debe ser de fácil instalación y configuración, con vista a poder darle un mantenimiento relativamente fácil y sencillo en caso de fallos.

Legales:

- La plataforma está basada en la licencia GNU/GPL.
- El sistema es reconocido y autorizado por entidades superiores tales como la directiva de la UCI y la Dirección Territorial de ETECSA en Granma.

2.11 Modelación del Sistema.

Los actores del sistema definen el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que interactúan con el mismo intercambiando información con este. A continuación se detallan los actores del sistema:

2.12 Descripción de los Actores del sistema.

Actores del Sistema	Justificación
Jefe de la UGAA	Es el encargado de que gestionar y controlar toda la información del sistema.
Trabajador Responsable	Es el encargado de darle solución a las interrupciones.
Trabajador Afectado	Es el encargado de reportar la interrupción.
Servidor LDAP	Es el encargado de autenticar a los usuarios.

Tabla 4: Descripción de los Actores del Sistema.

2.13 Diagrama de casos de uso del sistema.

Los Casos de Uso del sistema (CUS) son un conjunto de secuencia de acciones que un sistema ejecuta y que produce un resultado observable para un actor. Con otras palabras, son “fragmentos” de funcionalidad que el sistema ofrece a los actores que interactúan con el mismo, reportándoles algunos que otro beneficio.

A continuación se muestra la figura 2.10 correspondiente al diagrama de casos de uso del sistema:

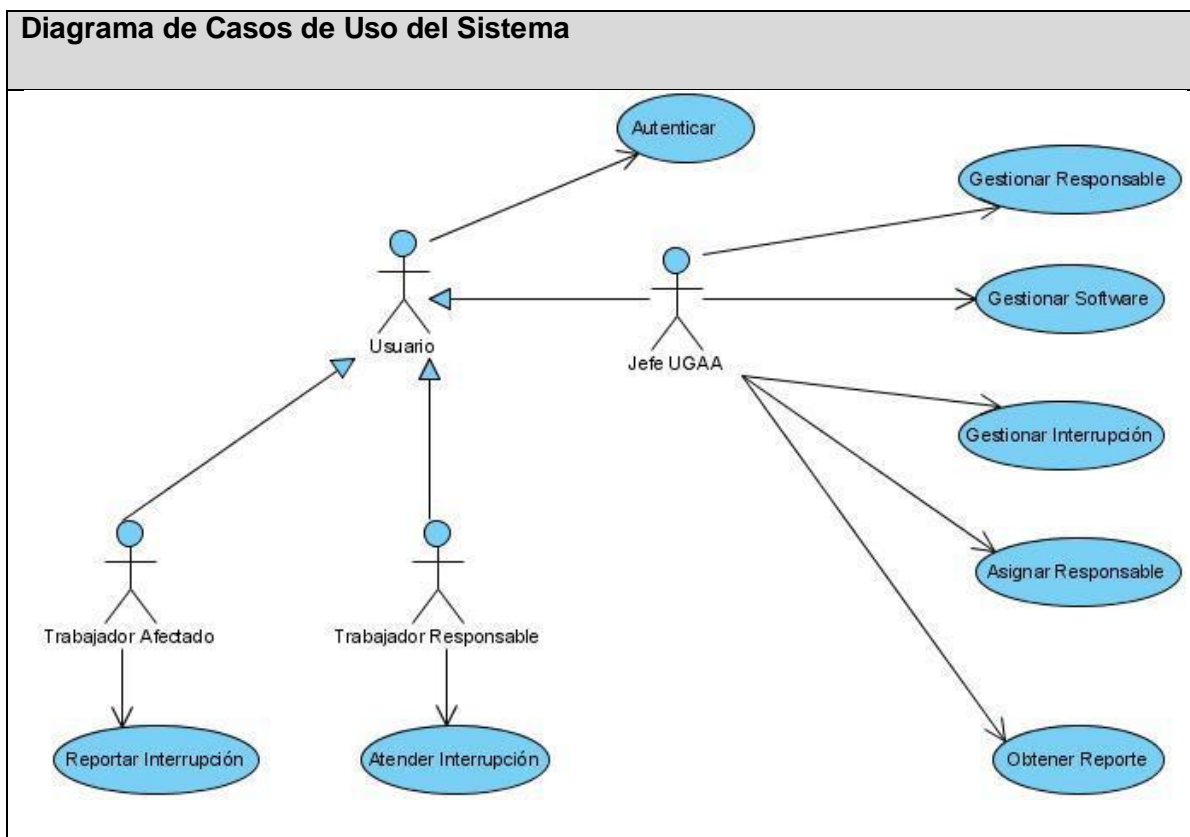


Figura 4: Diagrama de Casos de Uso del Sistema.

2.14 Descripción de los Casos de Uso del Sistema.

La descripción de los casos de uso del sistema, detallan las acciones que tienen lugar durante la interacción actor-sistema, es decir, describe el flujo de actividades que realiza el actor al hacer uso del sistema y las correspondientes respuestas del mismo. Por lo tanto, establece un acuerdo entre clientes y desarrolladores sobre qué es lo que el sistema debe hacer (requisitos).

2.14.1 Descripción del CUS “Reportar Interrupción”.

Caso de Uso	Reportar Interrupción	
Actores	Trabajador Afectado	
Propósito	Ofrece la posibilidad de reportar una interrupción con los problemas que presenta el software a reportar.	
Resumen	El Caso de Uso comienza cuando el trabajador afectado selecciona la opción “Reportar Interrupción” e introduce los datos necesario para registrar la misma.	
Referencias	RF 1, RF 3	
Precondiciones	- El trabajador debe estar logueado en el sistema.	
Poscondiciones	<ul style="list-style-type: none"> - Interrupción adicionada a la Base de Datos. - Notificación de la acción mediante el envío de un correo electrónico. 	
Curso Normal de Eventos		
Acciones del Actor	Respuesta del Sistema	
1- Selecciona la opción de Reportar Interrupción.	2- El sistema muestra un formulario para que el Trabajador Afectado introduzca los datos necesario para insertar una Interrupción al sistema.	
3- Introduce los datos necesarios para reportar la interrupción y pulsa el botón Reportar.	4- El sistema verifica que los datos son correctos y en caso afirmativo inserta los datos en la base de datos.	
	5- El sistema envía un correo electrónico con los datos necesarios del reporte tanto al	

	Trabajador Afectado como al Responsable como forma de notificación.
	6- Muestra un mensaje de que el correo ha sido enviado.
Curso Alternativo de los Eventos	
	En caso de que los datos no sean validos el sistema muestra un mensaje en los campos incorrectos del formulario. Se procede al flujo normal de eventos desde la acción 3.
Prioridad	Crítico

Tabla 5: Descripción del CUS Crear Interrupción.

2.14.2 Descripción del CUS “Atender Interrupción”.

Caso de Uso	Atender Interrupción
Actores	Trabajador Responsable.
Propósito	Ofrece la posibilidad de solucionar el problema lo antes posible por el cual se reportó la interrupción.
Resumen	El Caso de Uso comienza cuando el trabajador responsable selecciona la opción “Atender Interrupción” e introduce los datos necesario de acuerdo a la prestación del servicio de solución.
Referencias	RF 1, RF 2, RF 4
Precondiciones	El trabajador debe estar logueado en el sistema.
Poscondiciones	<ul style="list-style-type: none"> ➤ Interrupción actualizada en la Base de Datos. ➤ Notificación de la acción mediante el envío de un correo

	electrónico.
Curso Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1- Selecciona la opción de Atender Interrupción.	2- El sistema muestra un listado con todas las interrupciones que le corresponde atender a dicho responsable.
3- Selecciona la interrupción que desee atender.	4- El sistema muestra los datos de dicha interrupción.
5- El trabajador responsable verifica los datos y realiza una conexión remota al medio desde donde fue reportada la interrupción.	
6- Al atender la interrupción selecciona la opción "Actualizar Interrupción".	7- El sistema muestra el formulario con los campos a actualizar.
8- Introduce los datos necesarios para la actualización.	9- El sistema verifica los datos introducidos en caso de estar correctos actualiza la interrupción con los nuevos valores.
	10- Realiza la notificación mediante el envío de un correo electrónico al Trabajador Afectado.
Curso Alternativo de los Eventos	
	9- En caso de que los datos no sean válidos el sistema muestra un mensaje en los campos incorrectos del formulario. Se procede al flujo normal de eventos desde la acción 6.
Prioridad	Crítico

Tabla 6: Descripción del CUS Atender Interrupción.

2.14.3 Descripción del CUS “Gestionar Interrupción”.

Caso de Uso	Gestionar Interrupción	
Actores	Jefe de la UGAA	
Propósito	Ofrece la posibilidad al Jefe de la UGAA gestionar la información relacionada con los Responsables como Insertar Interrupción, Modificar Interrupción, Eliminar Interrupción y Buscar Interrupción.	
Resumen	El Caso de Uso comienza cuando el jefe de la UGAA selecciona la opción “Gestionar Interrupción”, luego escoge la acción a ejecutar e introduce los datos necesarios, el sistema ejecuta dicha acción y finaliza el CU.	
Referencias	RF 1, RF 2	
Precondiciones	- El jefe de la UGAA debe estar logueado en el sistema.	
Poscondiciones	<ul style="list-style-type: none"> - Interrupción adicionada a la Base de Datos. - Interrupción eliminada de la Base de Datos. - Datos del Interrupción actualizados. 	
Curso Normal de Eventos		
Acciones del Actor	Respuesta del Negocio	
1- Selecciona la opción de Gestionar Interrupción.	2- El sistema muestra las opciones: Insertar Interrupción Editar Interrupción Eliminar Interrupción.	

Escenario 1: Insertar Interrupción	
1- Selecciona la opción Insertar Interrupción	2- El sistema muestra un formulario con todos los datos que el jefe de la UGAA debe llenar para insertar la nueva Interrupción.
3- Introduce los datos necesarios para registrar la Interrupción y pulsa el botón Aceptar.	4- El sistema verifica que los datos son correctos y en caso afirmativo inserta los datos en la base de datos confluendo así el CUS.
Curso Alternativo de los Eventos	
	En caso de que los datos no sean válidos el sistema muestra un mensaje en los campos incorrectos del formulario. Se procede al flujo normal de eventos desde la acción 3.
Escenario 2: Editar Interrupción	
1- Selecciona la opción Editar Interrupción	2- El sistema brinda la posibilidad de que el jefe de la UGAA busque el Interrupción que desea editar devolviendo todas las Interrupciones que cumplan con el criterio de búsqueda.
3- El jefe de la UGAA selecciona la Interrupción que desea editar.	4- El sistema muestra un formulario con todos los datos de la Interrupción.
5- Modifica los datos deseados y pulsa el botón Editar.	6- El sistema verifica que los datos son correctos y en caso afirmativo inserta los datos en la base de datos confluendo así el CUS.
Curso Alternativo de los Eventos	
	En caso de que los datos no sean válidos el sistema muestra un mensaje en los campos incorrectos del formulario. Se procede al flujo

		normal de eventos desde la acción 5.
Escenario 3: Eliminar Interrupción		
1- Selecciona la opción Eliminar Interrupción	2- El sistema brinda la posibilidad de que el jefe de la UGAA busque la Interrupción que desea eliminar devolviendo todos las Interrupciones que cumplan con el criterio de búsqueda.	
3- El jefe de la UGAA selecciona la Interrupción que desea eliminar y presiona el botón eliminar.	3- El sistema elimina la Interrupción de la base de datos finalizando así el CUS.	
Prioridad	Crítico	

Tabla 7: Descripción del CUS Gestionar Interrupción.

2.14.4 Descripción del CUS “Asignar Responsable”.

Caso de Uso	Asignar Responsable
Actores	Jefe de la UGAA
Propósito	Ofrece la posibilidad de cambiar el responsable de una interrupción en caso que sea necesario.
Resumen	El Caso de Uso comienza cuando el jefe de la UGAA selecciona la opción “Cambiar Responsable” y le asigna un nuevo responsable a la interrupción deseada, el sistema actualiza los datos y finaliza así el CU.
Referencias	RF 1, RF 2.
Precondiciones	- El jefe de la UGAA debe estar logueado en el sistema.
Poscondiciones	- Interrupción actualizada en la Base de Datos. - Notificación de la acción mediante el envío de un correo

	electrónico.
Curso Normal de Eventos	
Acciones del Actor	Respuesta del Negocio
1- Selecciona la opción de Asignar Responsable.	2- El sistema muestra un listado de los responsables con más interrupciones pendientes en orden descendente y otro con los que no tienen.
3- El jefe de la UGAA selecciona el responsable más cargado y presiona el botón "Ver Interrupciones"	4- El sistema muestra todas las interrupciones pendientes que presenta el responsable seleccionado.
5- El jefe de la UGAA selecciona de la lista la interrupción deseada.	6- El sistema muestra los datos de la interrupción a actualizar.
7-Y selecciona de la lista de trabajadores sin interrupciones pendientes el que desea asignar.	8- El sistema actualiza la interrupción con los nuevos valores.
	9- Realiza la notificación mediante el envío de un correo electrónico al nuevo Trabajador Responsable.
Curso Alternativo de los Eventos	
Prioridad	Crítico

Tabla 8: Descripción del CUS Asignar Responsable.

2.14.5 Descripción del CUS "Obtener Reportes".

Caso de Uso	Obtener Reportes
--------------------	-------------------------

Actores	Jefe de la UGAA	
Propósito	Ofrece la posibilidad al Jefe de la UGAA obtener reportes de las interrupciones de acuerdo a ciertos criterios como son Reportes por fecha, por responsables, por área y por software así como obtener el software mas involucrado en las interrupciones.	
Resumen	El Caso de Uso comienza cuando el jefe de la UGAA selecciona la opción "Obtener Reporte", luego escoge la acción a ejecutar e introduce los datos necesarios, el sistema ejecuta dicha acción y finaliza el CU.	
Referencias	RF 1, RF 6	
Precondiciones	- El jefe de la UGAA debe estar logueado en el sistema.	
Poscondiciones	- Muestra de reportes de acuerdo el criterio seleccionado.	
Curso Normal de Eventos		
Acciones del Actor	Respuesta del Negocio	
1- Selecciona la opción de Obtener Reporte.	2- El sistema muestra las opciones: Reporte por Fecha Reporte por Responsable Reporte por Área Reporte por Software. Software más involucrado.	
Escenario 1: Reporte por Fecha		
1- Selecciona la opción Reporte por Fecha.	2- El sistema brinda la opción de introducir del rango de fecha.	

3- Introduce los datos necesarios para obtener el reporte y pulsa el botón Aceptar.	4- El sistema verifica que los datos son correctos y en caso afirmativo muestra una lista con las interrupciones que cumplen con el criterio seleccionado.
Curso Altero de los Eventos	
	En caso de que los datos no sean validos el sistema muestra un mensaje en los campos incorrectos del formulario. Se procede al flujo normal de eventos desde la acción 3.
Escenario 2: Reporte por Responsable	
1- Selecciona la opción Reporte por Responsable	2- El sistema brinda la opción de introducir el nombre del responsable.
3- Introduce los datos necesarios para obtener el reporte y pulsa el botón Aceptar.	4- El sistema verifica que los datos son correctos y en caso afirmativo muestra una lista con las interrupciones que cumplen con el criterio seleccionado.
Curso Alterno de los Eventos	
	En caso de que los datos no sean validos el sistema muestra un mensaje en los campos incorrectos del formulario. Se procede al flujo normal de eventos desde la acción 3.
Escenario 3: Reporte por Área	
1- Selecciona la opción Reporte por Área.	2- El sistema brinda la opción de introducir el nombre del área.
3- Introduce los datos necesarios para obtener el reporte y pulsa el botón	4- El sistema verifica que los datos son correctos y en caso afirmativo muestra una lista

Aceptar.	con las interrupciones que cumplen con el criterio seleccionado.
Curso Alternativo de los Eventos	
	En caso de que los datos no sean válidos el sistema muestra un mensaje en los campos incorrectos del formulario. Se procede al flujo normal de eventos desde la acción 3.
Escenario 4: Reporte por Software.	
1- Selecciona la opción Reporte por Software.	2- El sistema brinda la opción de introducir el nombre del software.
3- Introduce los datos necesarios para obtener el reporte y pulsa el botón Aceptar.	4- El sistema verifica que los datos son correctos y en caso afirmativo muestra una lista con las interrupciones que cumplen con el criterio seleccionado.
Curso Alternativo de los Eventos	
	En caso de que los datos no sean válidos el sistema muestra un mensaje en los campos incorrectos del formulario. Se procede al flujo normal de eventos desde la acción 3.
Escenario 4: Software más involucrado.	
1- Selecciona la opción Software más involucrado.	2- El sistema muestra los Software que se repiten en las interrupciones registradas.
Prioridad	Crítico

Tabla 9: Descripción del CUS Obtener Reportes

2.14.6 Descripción del CUS “Gestionar Software”.

Caso de Uso	Gestionar Software	
Actores	Jefe de la UGAA	
Propósito	Ofrece la posibilidad al Jefe de la UGAA gestionar las información relacionada con los Software que se usan como Insertar Software, Modificar Software, Eliminar Software y Buscar Software.	
Resumen	El Caso de Uso comienza cuando el Jefe de la UGAA selecciona la opción “Gestionar Software”, luego escoge la acción a ejecutar e introduce los datos necesarios, el sistema ejecuta dicha acción y finaliza el CU.	
Referencias	RF 7	
Precondiciones	El jefe de departamento debe estar logueado en el sistema.	
Poscondiciones	<ul style="list-style-type: none"> ➤ Software adicionado a la Base de Datos. ➤ Software eliminado a la Base de Datos. ➤ Datos del Software actualizados. 	
Curso Normal de Eventos		
Acciones del Actor	Respuesta del Negocio	
1- Selecciona la opción de Gestionar Software.	2- El sistema muestra las opciones: Insertar Software Editar Software Eliminar Software.	

Escenario 1: Insertar Software	
1- Selecciona la opción Insertar Software	2- El sistema muestra un formulario con todos datos que el Jefe de la UGAA debe llenar para insertar el nuevo Software.
3- Introduce los datos necesarios para insertar el Software y pulsa el botón Aceptar.	4- El sistema verifica que los datos son correctos y en caso afirmativo inserta los datos en la base de datos confluendo así el CUS.
Curso Altero de los Eventos	
	En caso de que los datos no sean validos el sistema muestra un mensaje en los campos incorrectos del formulario. Se procede al flujo normal de eventos desde la acción 3.
Escenario 2: Editar Software	
1- Selecciona la opción Editar Software.	2- El sistema brinda la posibilidad de que el Jefe de la UGAA busque el Software que desea editar devolviendo todos los Software que cumplan con el criterio de búsqueda.
3- El Jefe de la UGAA selecciona el Software que desea editar.	4- El sistema muestra un formulario con todos los datos del Software.
5- Modifica los datos deseados y pulsa el botón Editar.	7- El sistema verifica que los datos son correctos y en caso afirmativo inserta los datos en la base de datos confluendo así el CUS.
Curso Alterno de los Eventos	

		En caso de que los datos no sean validos el sistema muestra un mensaje en los campos incorrectos del formulario. Se procede al flujo normal de eventos desde la acción 5.
Escenario 3: Eliminar Software		
1- Selecciona la opción Eliminar Software.		2- El sistema brinda la posibilidad de que el Jefe de la UGAA busque el Software que desea eliminar devolviendo todos los Software que cumplan con el criterio de búsqueda.
3- El Jefe de la UGAA selecciona el Software que desea eliminar y presiona el botón Eliminar.		4- El sistema elimina el Software de la base de datos finalizando así el CUS.
Prioridad	Crítico	

Tabla 10: Descripción del CUS Gestionar Software.

2.14.7 Descripción del CUS “Gestionar Responsable”.

Caso de Uso	Gestionar Responsable
Actores	Jefe de la UGAA
Propósito	Ofrece la posibilidad al Jefe de la UGAA gestionar las información relacionada con los Trabajadores Responsables como Insertar Responsable, Modificar Responsable, Eliminar Responsable y Buscar Responsable.
Resumen	El Caso de Uso comienza cuando el Jefe de la UGAA selecciona la opción “Gestionar Responsable”, luego escoge la acción a ejecutar e introduce los datos necesarios, el sistema ejecuta dicha acción y finaliza el CU.

Referencias	RF 1, RF 8	
Precondiciones	El Jefe de la UGAA debe estar logueado en el sistema.	
Poscondiciones	<ul style="list-style-type: none"> ➤ Responsable adicionado a la Base de Datos. ➤ Responsable eliminado a la Base de Datos. ➤ Datos del Responsable actualizados. 	
Curso Normal de Eventos		
Acciones del Actor	Respuesta del Negocio	
1- Selecciona la opción de Gestionar Responsable.	2- El sistema muestra las opciones: Insertar Responsable Editar Responsable Eliminar Responsable.	
Escenario 1: Nuevo Responsable		
1- Selecciona la opción Insertar Responsable	2- El sistema muestra un formulario con todos datos que el Jefe de la UGAA debe llenar para insertar en el nuevo responsable.	
3- Introduce los datos necesarios para crear responsable y pulsa el botón Aceptar.	4- El sistema verifica que los datos son correctos y en caso afirmativo inserta los datos en la base de datos confluendo así el CUS.	
Curso Altero de los Eventos		
	En caso de que los datos no sean validos el sistema muestra un mensaje en los campos incorrectos del formulario. Se procede al	

	flujo normal de eventos desde la acción 3.
Escenario 2: Editar Responsable	
1- Selecciona la opción Editar Responsable	2- El sistema brinda la posibilidad de que el Jefe de la UGAA busque el responsable que desea editar devolviendo todos los responsable que cumplan con el criterio de búsqueda.
3- El Jefe de la UGAA selecciona el Responsable que desea editar.	4- El sistema muestra un formulario con todos los datos del Responsable.
5- Modifica los datos deseados y pulsa el botón Editar.	6- El sistema verifica que los datos son correctos y en caso afirmativo inserta los datos en la base de datos confluyendo así el CUS.
Curso Alterno de los Eventos	
	En caso de que los datos no sean validos el sistema muestra un mensaje en los campos incorrectos del formulario. Se procede al flujo normal de eventos desde la acción 5.
Escenario 3: Eliminar Responsable	
1- Selecciona la opción Eliminar Responsable	2- El sistema brinda la posibilidad de que el Jefe de la UGAA busque el responsable que desea eliminar devolviendo todos los responsable que cumplan con el criterio de búsqueda.
3- El Jefe de la UGAA selecciona el Responsable que desea eliminar y	4- El sistema elimina el responsable de la base de datos finalizando así el CUS.

presiona el botón eliminar.	
Prioridad	Crítico

Tabla 11: Descripción del CUS Gestionar Responsable.

2.15 Patrones de Casos de Usos

Permiten con más precisión reflejar los requisitos reales, haciendo más fácil el trabajo con los sistemas, y mucho más simple su mantenimiento. Los utilizados se muestran a continuación:

➤ **Patrón CRUD (Creating, Reading, Updating, Deleting):**

Este patrón se basa en la fusión de casos de uso simples para formar una unidad conceptual. Consta de un caso de uso, llamado Información CRUD o Gestionar información modela todas las operaciones que pueden ser realizadas sobre una parte de la información de un tipo específico, tales como creación, lectura, actualización y eliminación.

➤ **Patrón Múltiples Actores:**

Roles comunes:

Puede suceder que los dos actores jueguen el mismo rol sobre el CU. Este rol es representado por otro actor, heredado por los actores que comparten este rol.

2.16 Conclusiones.

Durante este capítulo fueron expuestas las características que contendrá el sistema, apoyándose para ello en el análisis de los actuales procesos de negocio, siendo identificado, quiénes son los actores y trabajadores que intervienen en el mismo y con cuáles actividades y entidades interactúan. Se identificaron además, los requisitos funcionales y no funcionales que debe cumplir el sistema en cuestión, y con ello fueron expuestos los casos de uso a tratar durante el desarrollo del mismo, con la correspondiente descripción textual de cada uno, lo cual provee de una visión general de qué es lo que el sistema debe hacer, por lo que se está en condiciones de pasar a ver cómo es que el mismo va a realizar las operaciones antes descritas.

CAPITULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Introducción.

En este capítulo se procede a la confección de la propuesta de solución, basándose en el análisis hecho anteriormente. Para el desarrollo de este capítulo se tuvieron en cuenta los requisitos funcionales, como no funcionales obtenidos durante el flujo de la captura de requisitos. El mismo constará con varios artefactos que tienen como objetivos mostrar el diseño y la implementación del sistema propuestos como solución.

3.2 Diagrama de Clases del Análisis

En el análisis se presentan los siguientes estereotipos de clases:

Clase de frontera o interfaz: Modela la interfaz del sistema, y manejan la comunicación entre el entorno y el interior del mismo. Durante el diseño, estas clases son refinadas para tomar en consideración los mecanismos de interfaz seleccionados o implementados, además de facilitar la comunicación con otros sistemas, etc.

Clases de entidad o sistema: Representan la información manejada en el caso de uso, además de que modelan información y comportamiento asociado que generalmente es de larga duración. Reflejan entidades del mundo real, que resultan necesarias para realizar tareas internas del sistema.

Clases de control o software: Coordinan los eventos necesarios para la realización o especificación del caso de uso, con otras palabras, son las que ejecutan el caso de uso. Usualmente son dependientes de la aplicación, además de tener un control sobre todas las acciones a realizar.

3.2.1 Diagrama de Clases del Análisis del CUS Reportar Interrupción.

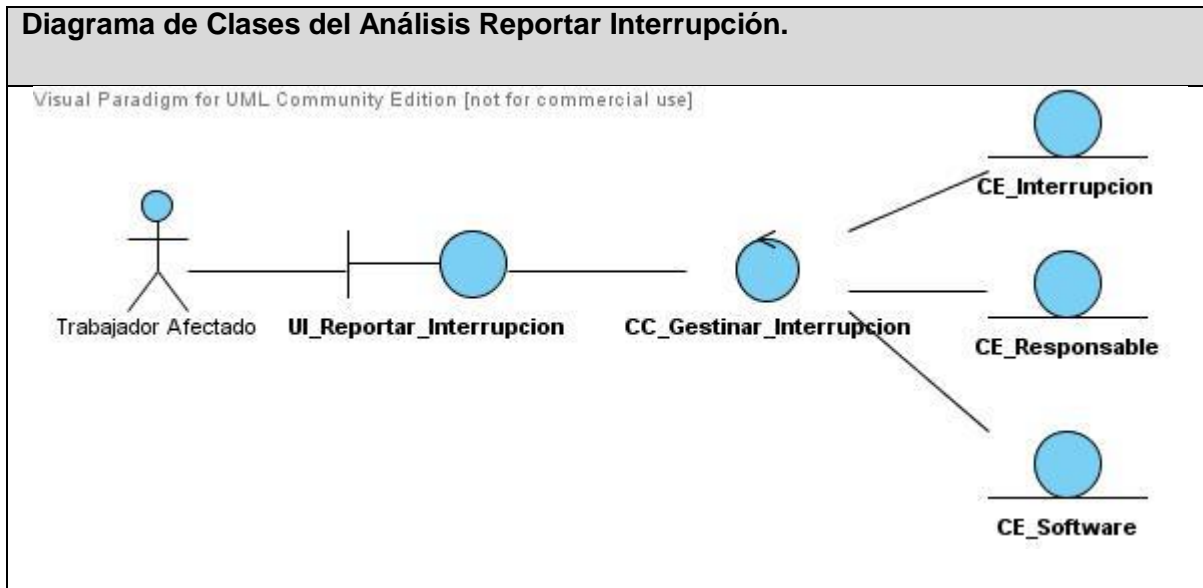


Figura 5: Diagrama de clases de análisis del CUS Reportar Interrupción.

3.2.2 Diagrama de Clases del Análisis del CUS Atender Interrupción.

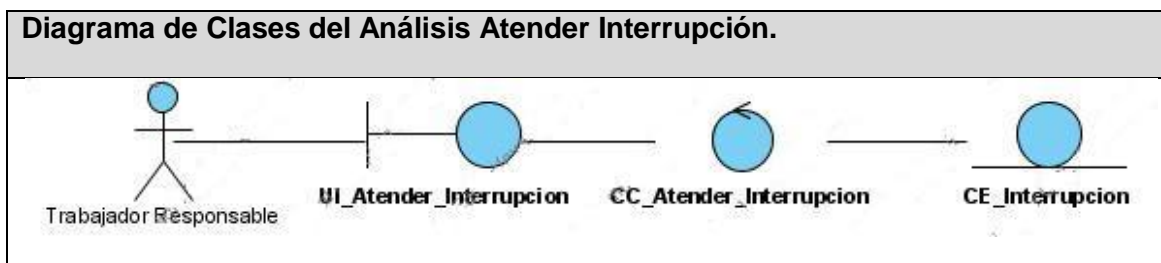


Figura 6: Diagrama de clases de análisis del CUS Atender Interrupción.

3.2.3 Diagrama de Clases del Análisis del CUS Gestionar Interrupción.

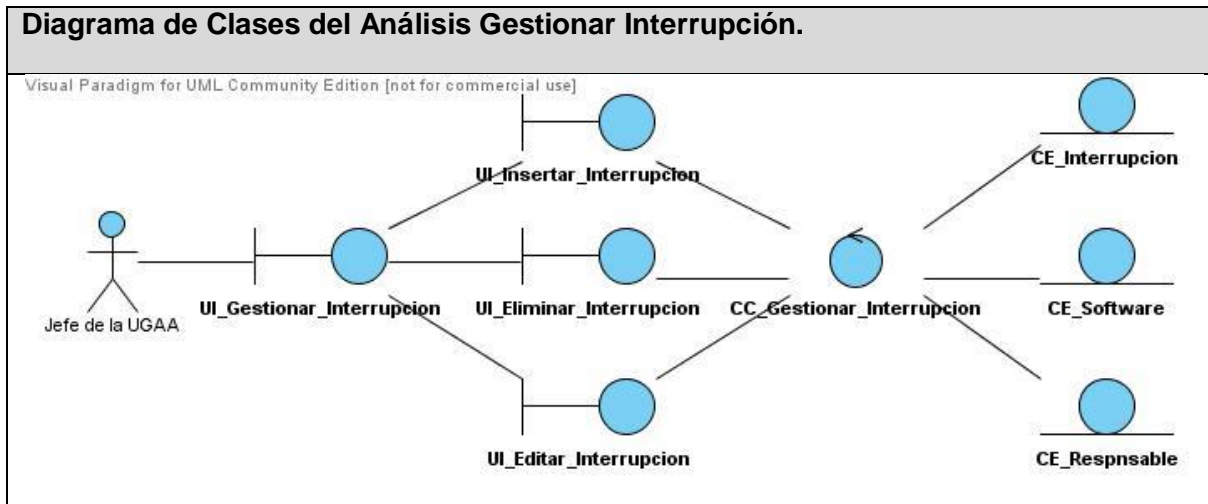


Figura 7: Diagrama de clases de análisis del CUS Gestionar Interrupción.

3.2.4 Diagrama de Clases del Análisis del CUS Obtener Reporte.

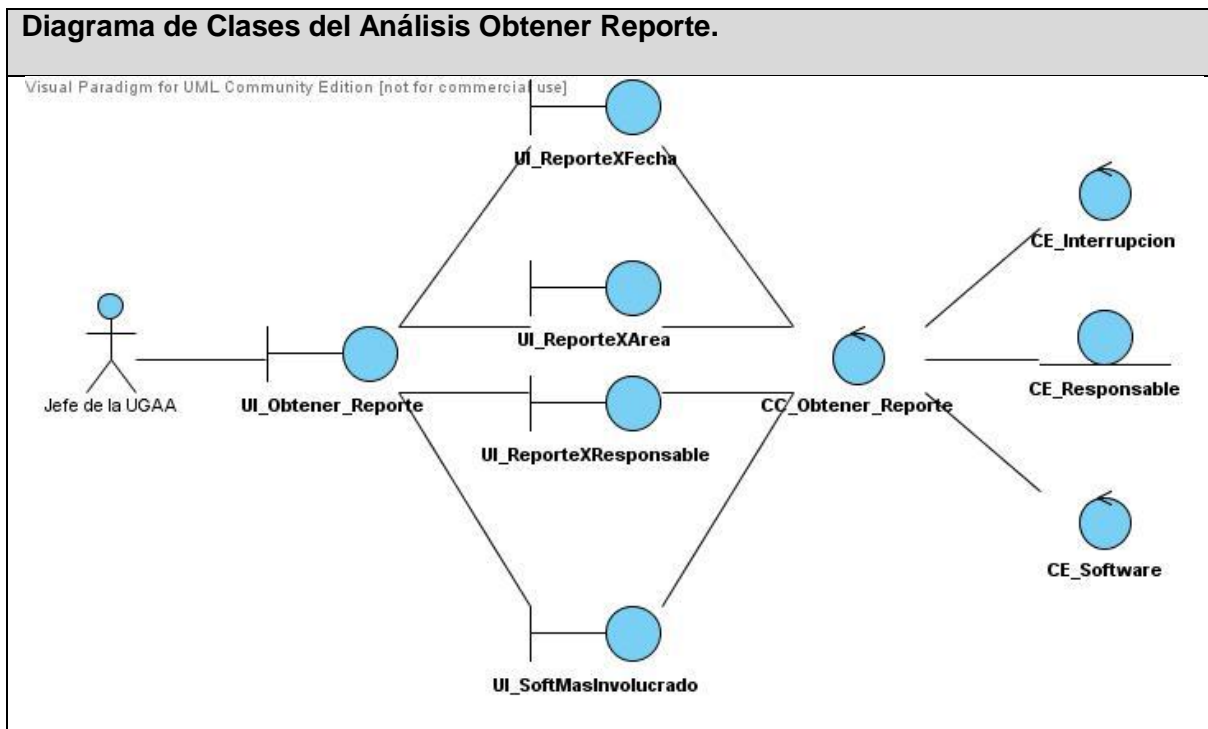


Figura 8: Diagrama de clases de análisis del CUS Obtener Reporte.

3.2.5 Diagrama de Clases del Análisis del CUS Asignar Responsable.

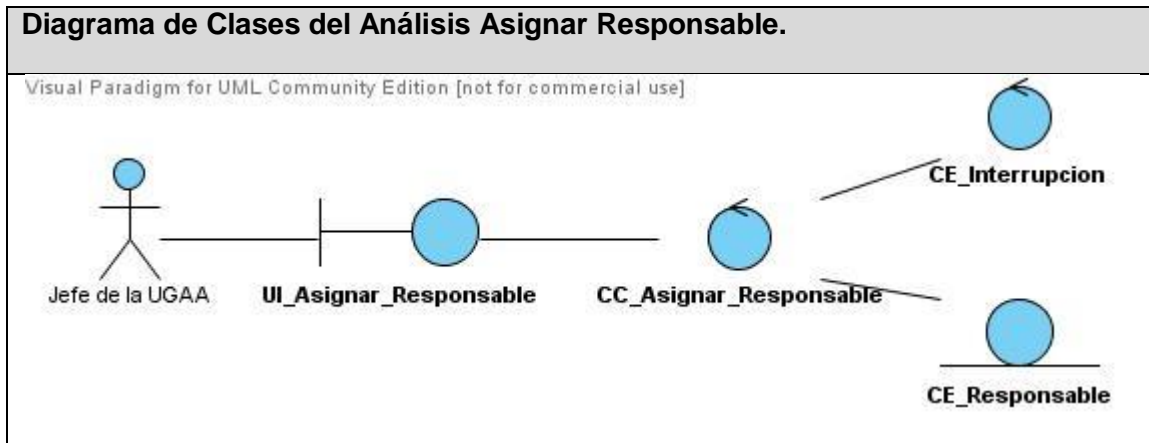


Figura 9: Diagrama de clases de análisis del CUS Asignar Responsable.

3.2.6 Diagrama de Clases del Análisis del CUS Gestionar Software.

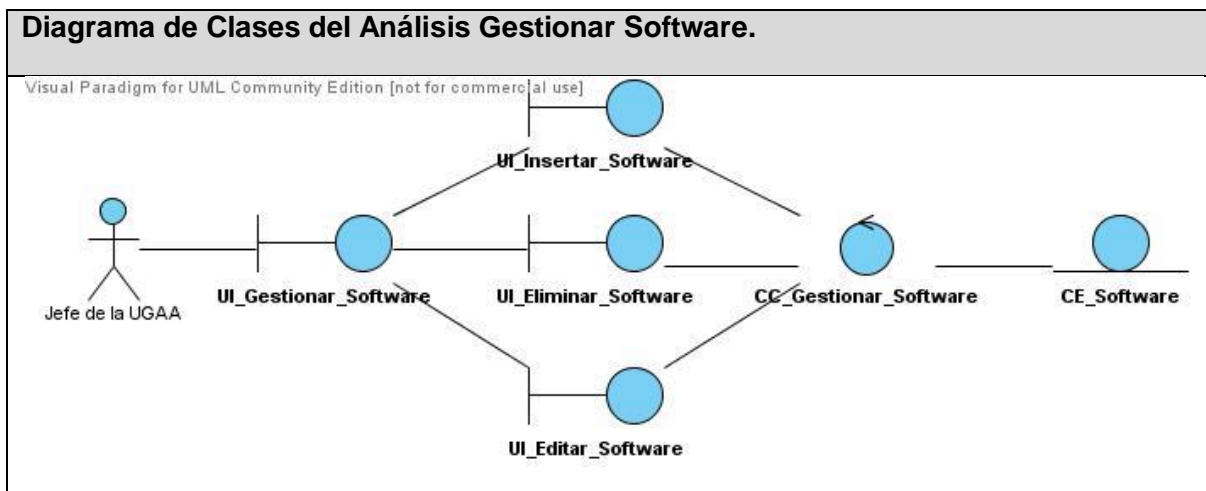


Figura 10: Diagrama de clases de análisis del CUS Gestionar Software.

3.2.7 Diagrama de Clases del Análisis del CUS Gestionar Responsable.

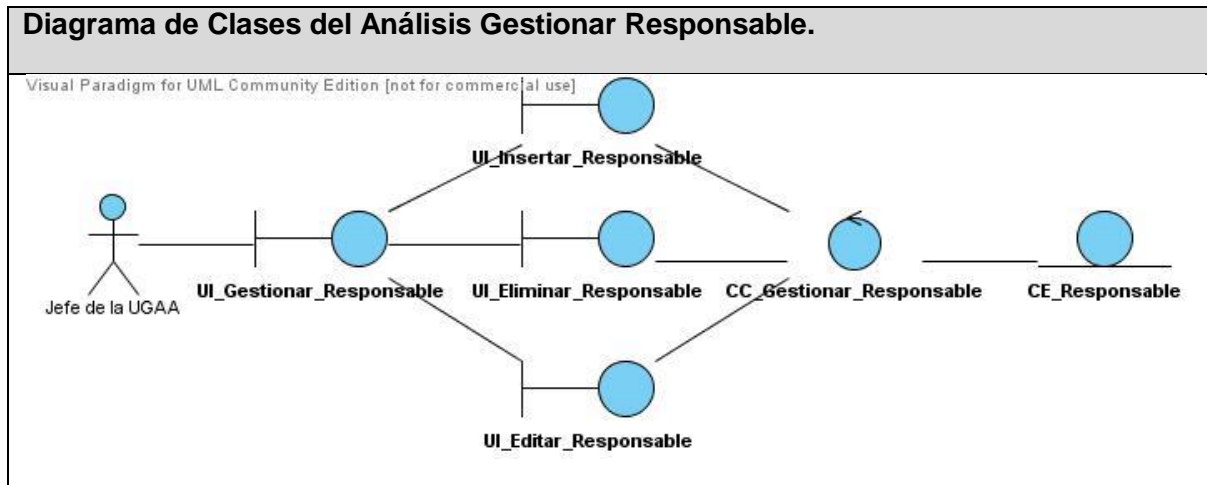


Figura 11: Diagrama de clases de análisis del CUS Gestionar Responsable.

3.3 Diagrama de Clases del Diseño.

Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia. En el caso de las aplicaciones Web, el diagrama de clases representa las colaboraciones que ocurren entre las páginas, donde cada página lógica puede ser representada como una clase.

3.3.1 Diagrama de Clases Web para el CUS Reportar Interrupción.

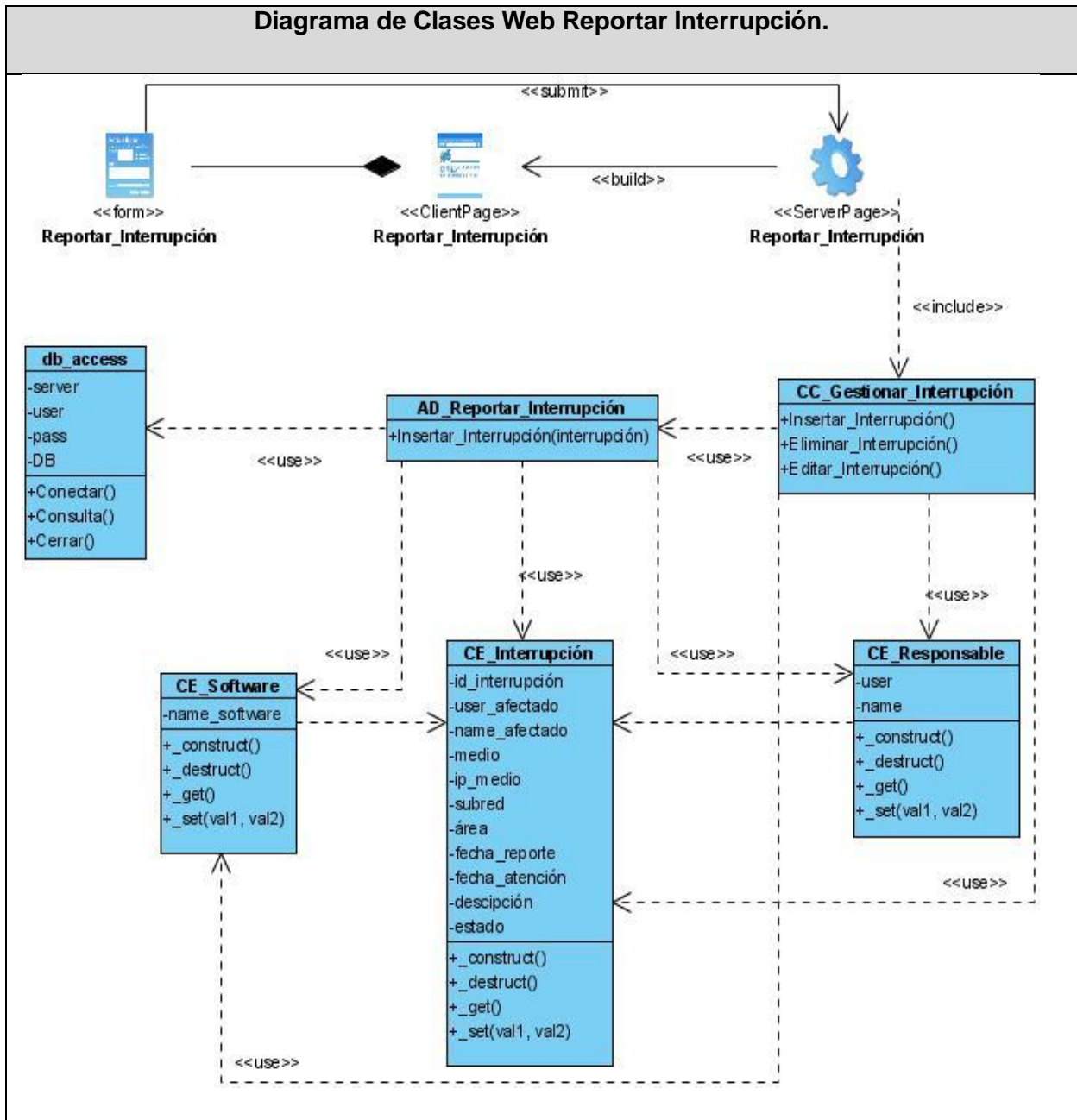


Figura 12: Diagrama de Clases Web del CUS Reportar Interrupción.

3.3.2 Diagrama de Clases Web para el CUS Atender Interrupción.

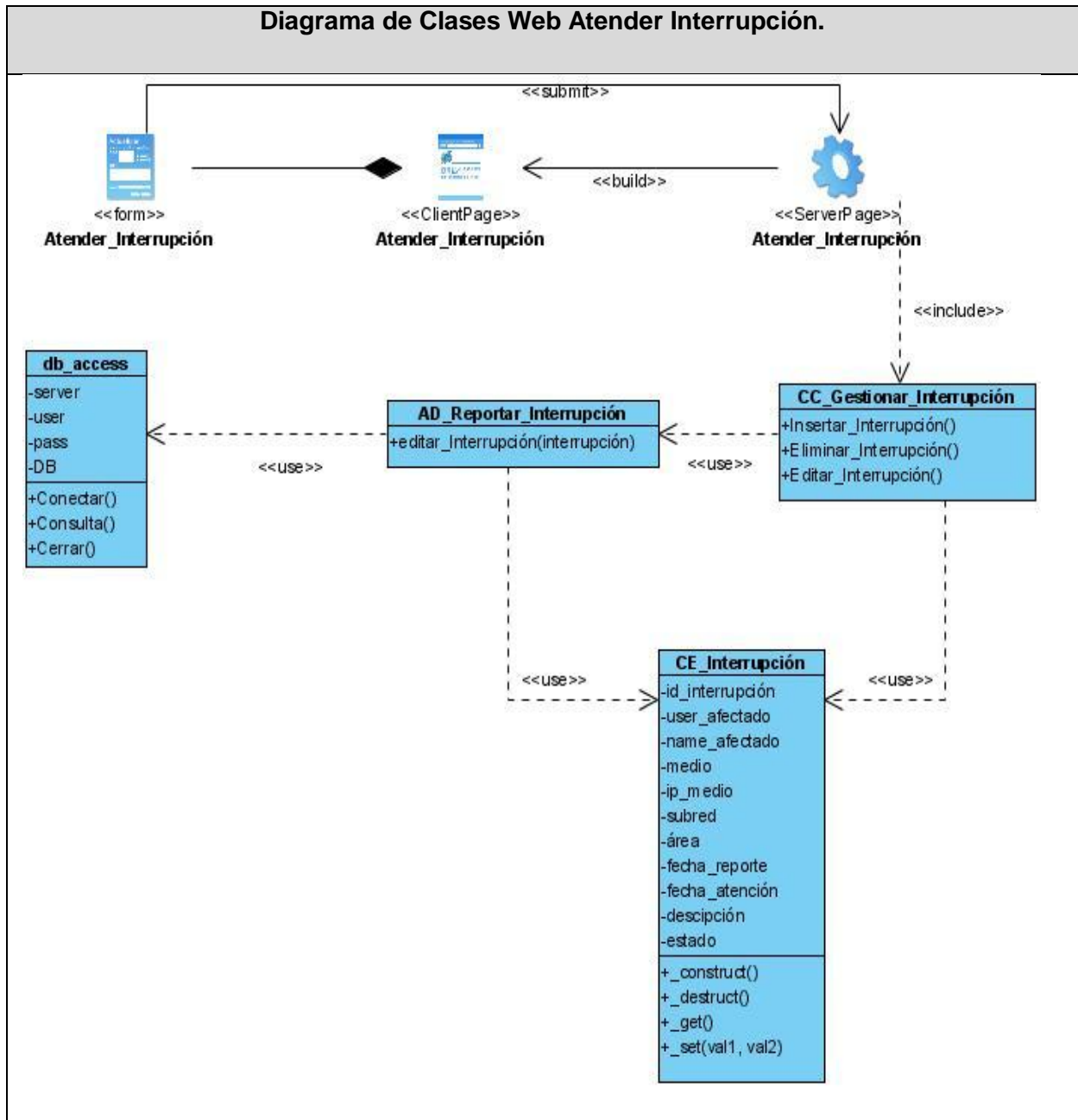


Figura 13: Diagrama de Clases Web del CUS Atender Interrupción.

3.3.3 Diagrama de Clases Web para el CUS Gestionar Interrupción.

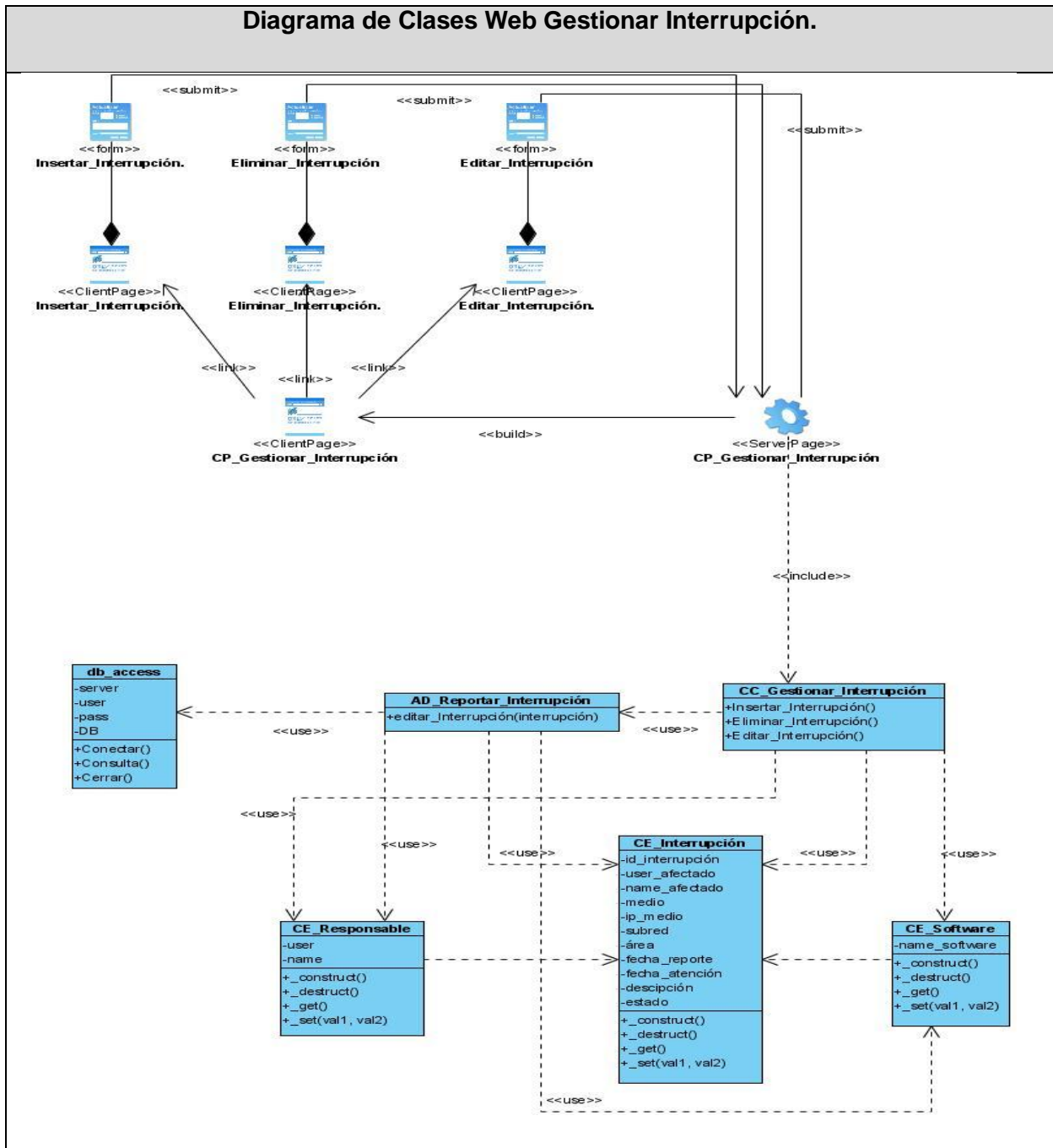


Figura 14: Diagrama de Clases Web del CUS Gestionar Interrupción.

3.3.4 Diagrama de Clases Web para el CUS Gestionar Responsable.

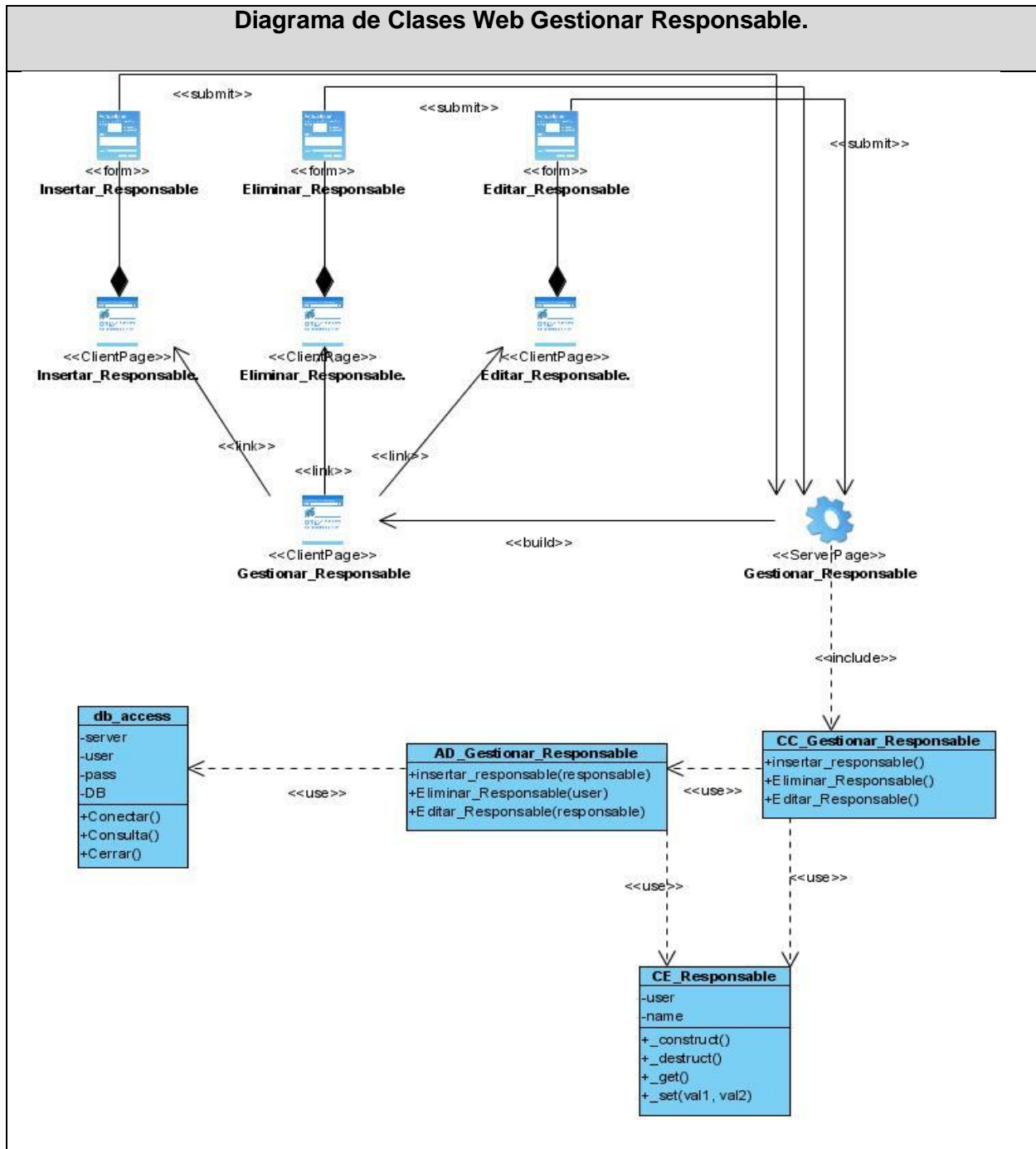


Figura 15: Diagrama de Clases Web del CUS Gestionar Responsable.

3.3.5 Diagrama de Clases Web para el CUS Asignar Responsable.

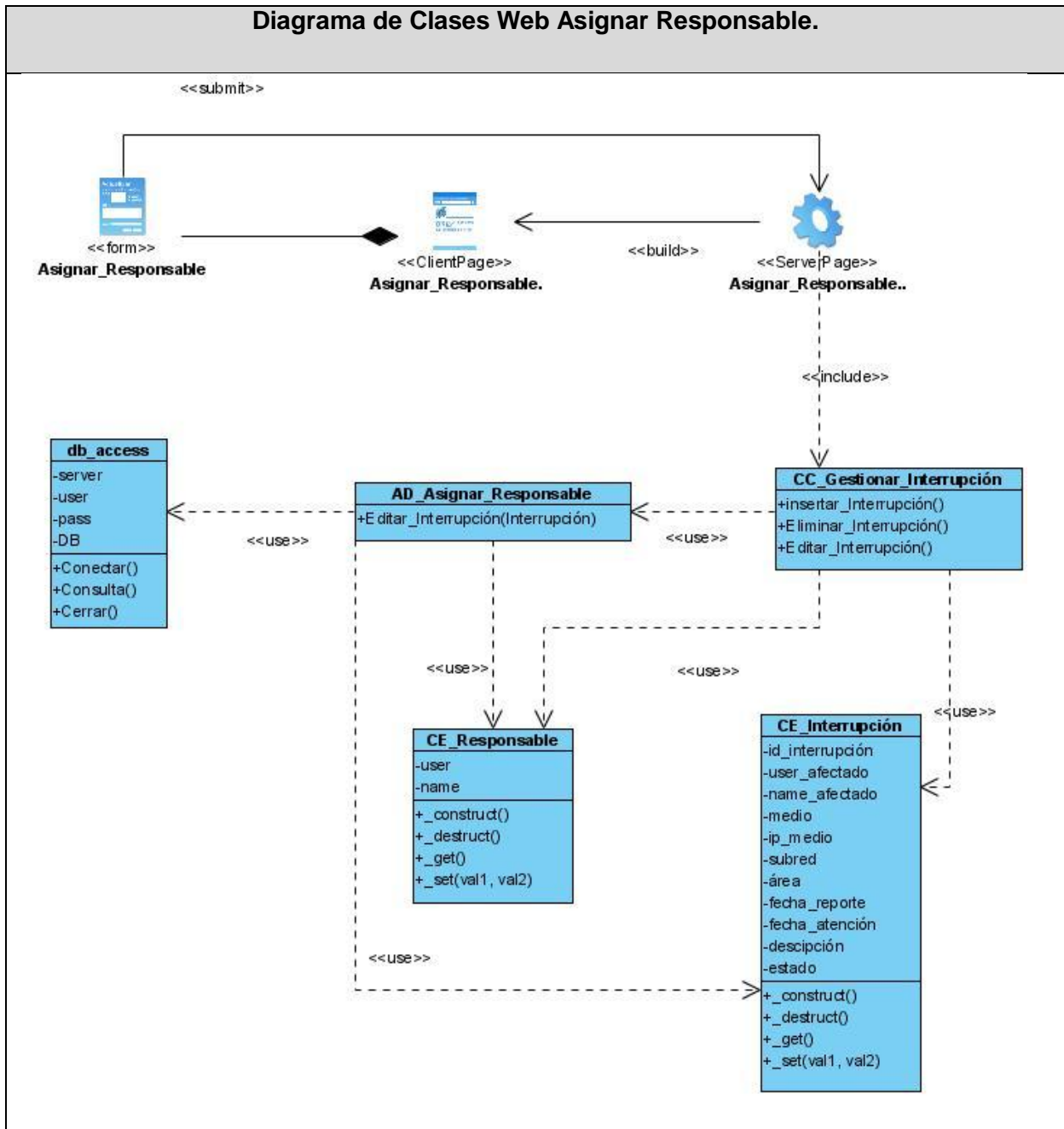


Figura 16: Diagrama de Clases Web del CUS Asignar Responsable.

3.3.6 Diagrama de Clases Web para el CUS Obtener Reporte.

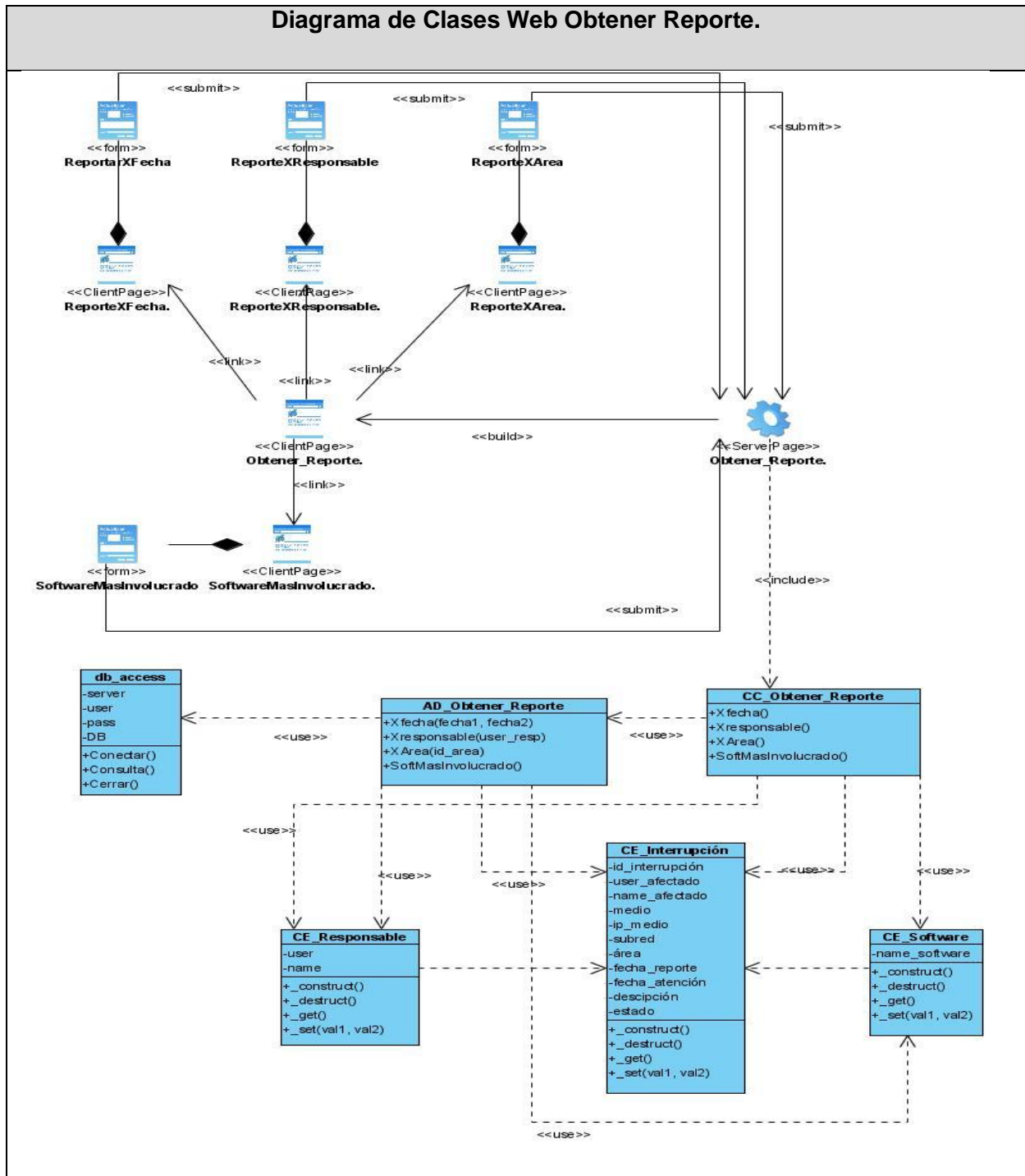


Figura 17: Diagrama de Clases Web del CUS Obtener Reporte.

3.3.7 Diagrama de Clases Web para el CUS Gestionar Software.

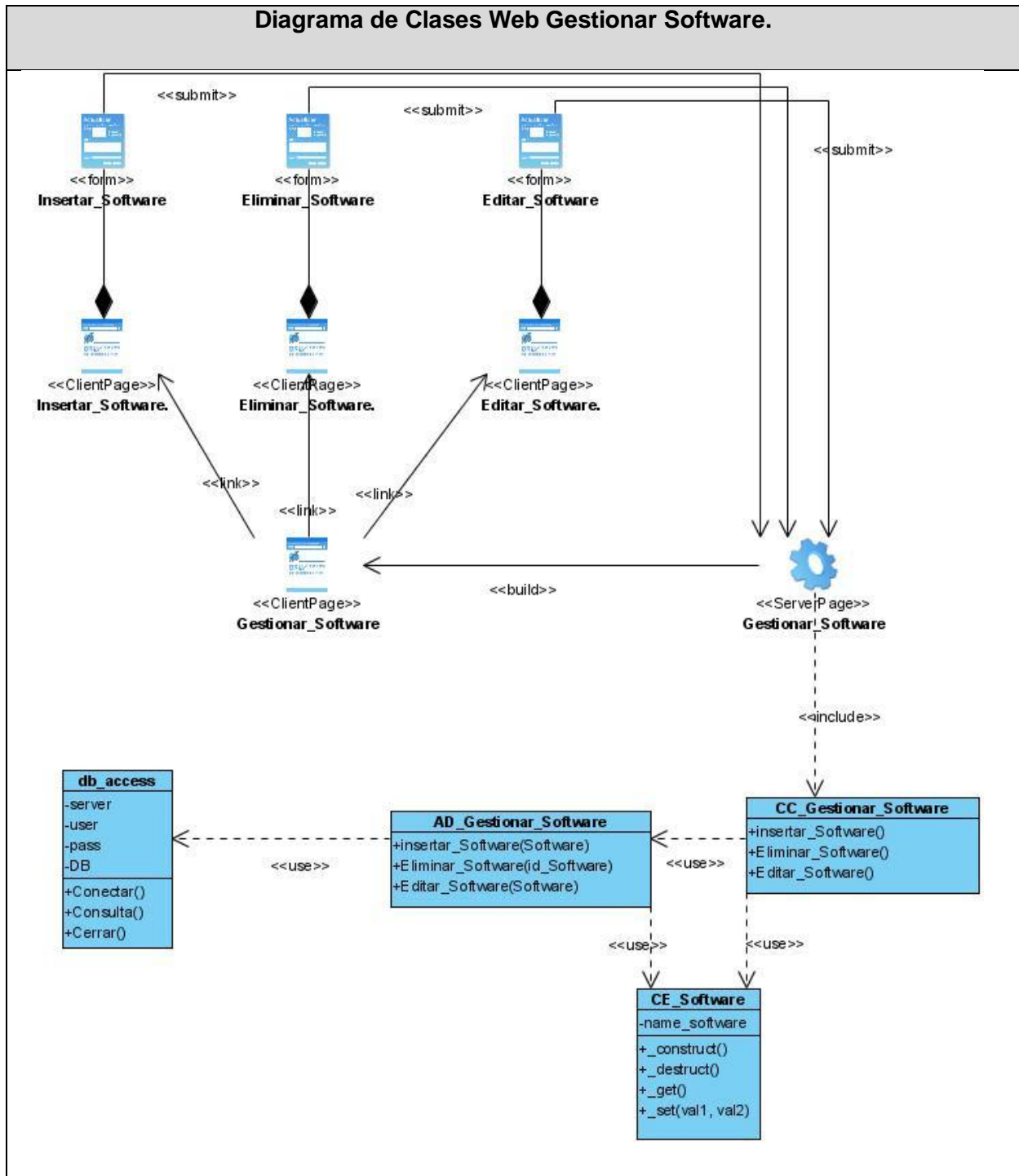


Figura 18: Diagrama de Clases Web del CUS Gestionar Software.

3.4 Diagramas de Interacción.

El modelo de diseño descrito para el desarrollo del sistema describe los diagramas de secuencia cada una de las realizaciones de los casos de uso, en esta sesión solo se representan cuatro diagramas, los cuales están entre los más importantes del sistema.

3.4.1 Diagrama de Secuencia del CUS Reportar Interrupción.

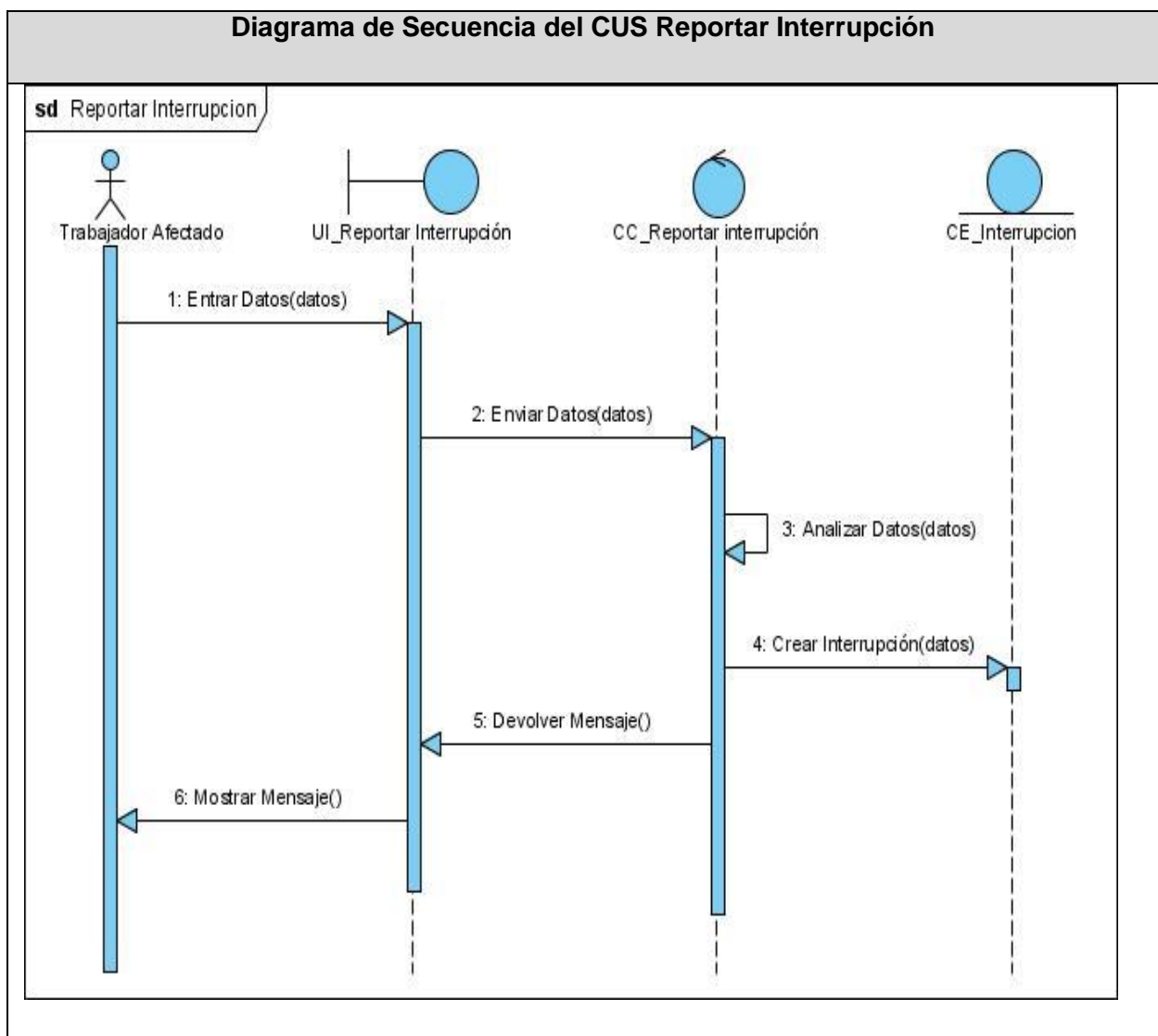


Figura 19: Diagrama de Secuencia del CUS Reportar Interrupción.

3.4.2 Diagrama de Secuencia del CUS Atender Interrupción.

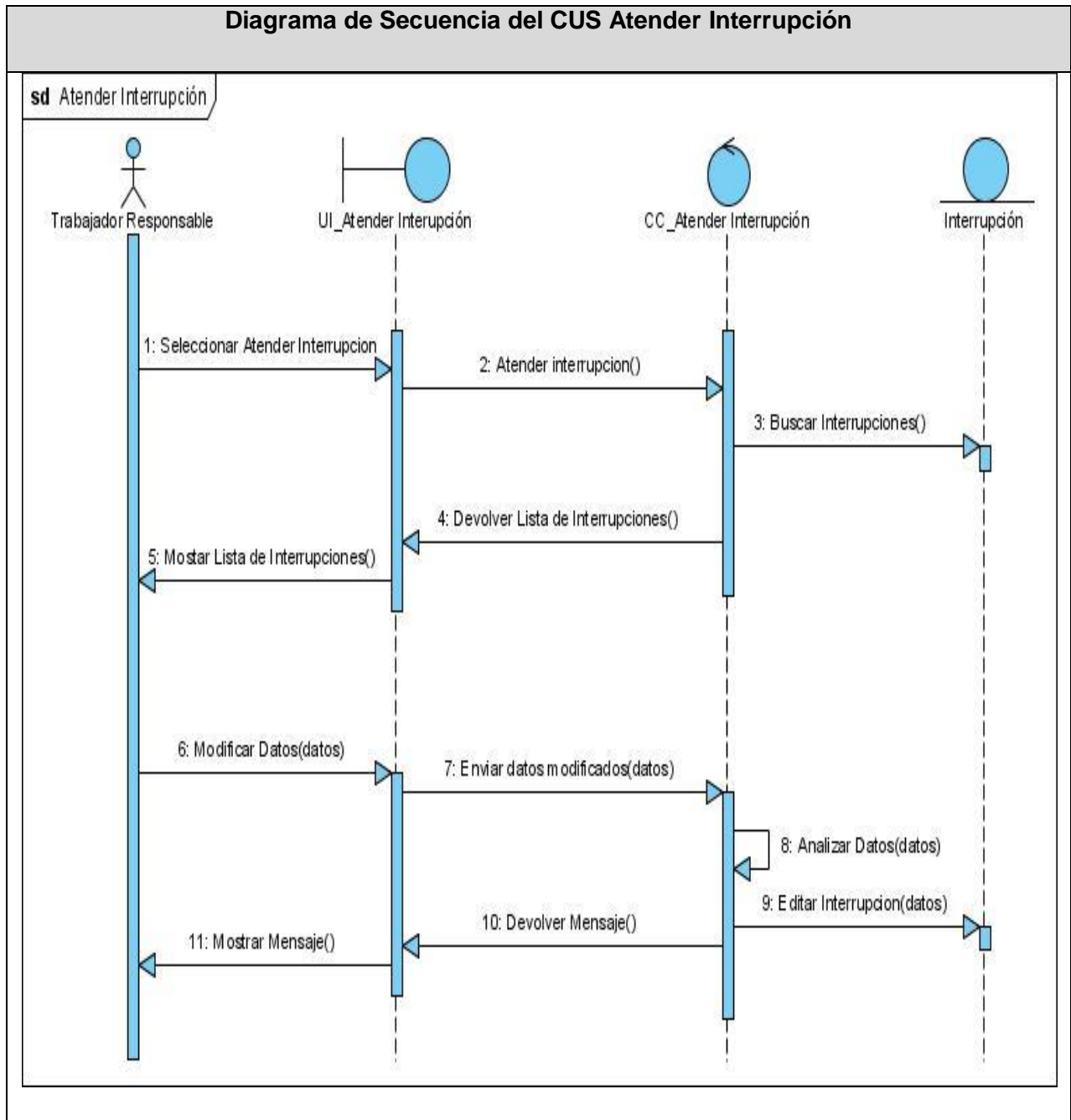


Figura 20: Diagrama de Secuencia del CUS Atender Interrupción.

3.4.3 Diagrama de Secuencia del CUS Asignar Responsable.

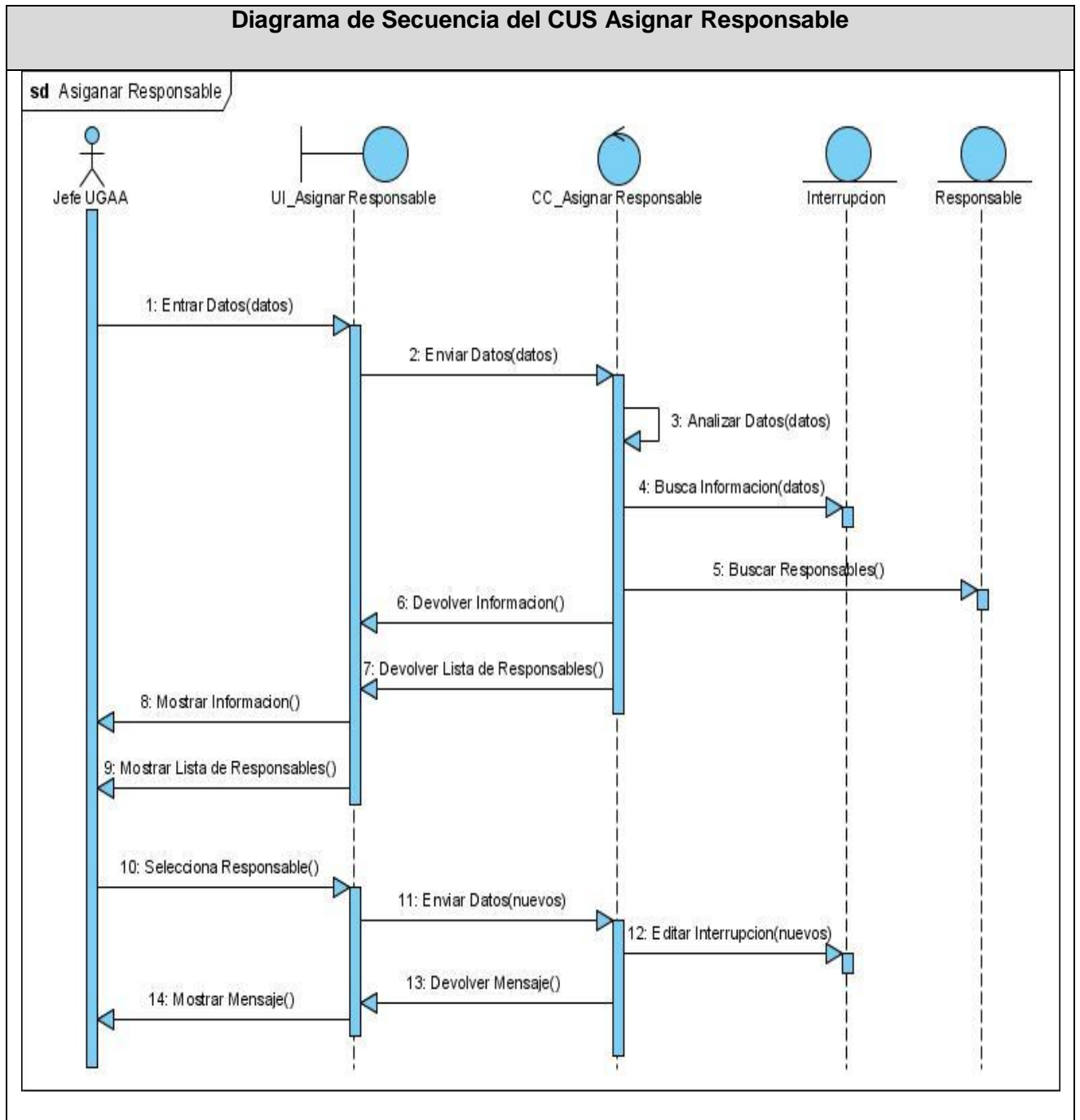


Figura 21: Diagrama de Secuencia del CUS Asignar Responsable.

3.4.4 Diagrama de Secuencia del CUS Obtener Reporte.

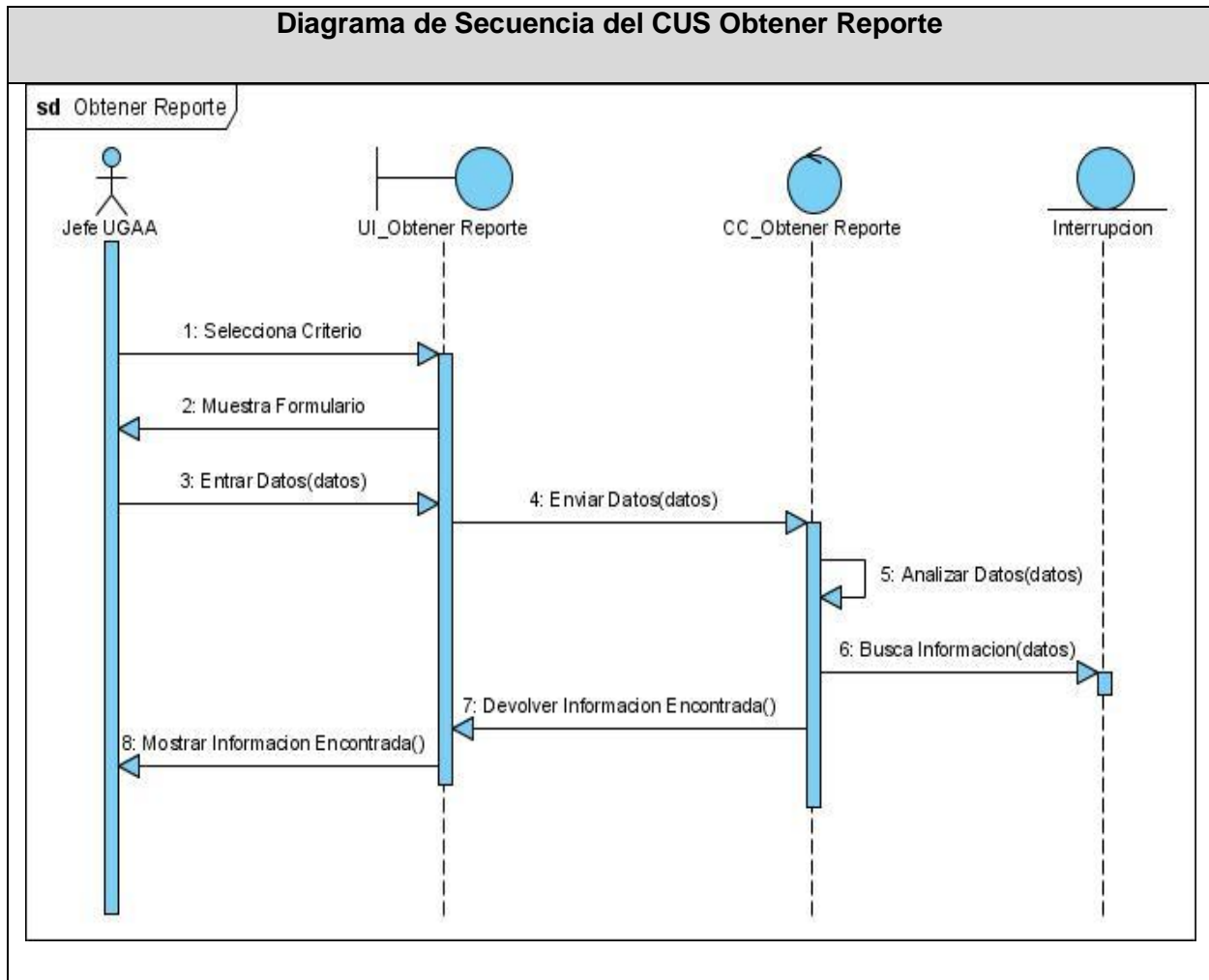


Figura 22: Diagrama de Secuencia del CUS Obtener Reporte.

3.5 Descripción Textual de las clases Web.

La descripción textual de las clases del diseño permiten comprender el alcance y la responsabilidad de cada una de estas dentro del sistema, con dicha descripción se puede conocer qué funcionalidad específica realiza cada clase, además de la información que cada una de esta maneja.

3.5.1 Clase Controladora CC_Gestionar_Interrupcion.

Nombre	CC_Gestionar_Interrupcion
Tipo de Clase	Controladora
Atributo	Tipo
Responsabilidad	
Insertar_Interrupcion()	
Eliminar_interrupcion()	
Editar_Interrupcion()	
Descripción	Es la clase encargada de gestionar todo la información necesaria de las interrupciones.

Tabla 12: Descripción de la Clase Controladora CC_Gestionar_Interrupcion.

3.5.3 Clase Controladora CC_Gestionar_Responsable.

Nombre	CC_Gestionar_Responsable
Tipo de Clase	Controladora
Atributo	Tipo
Responsabilidad	
Insertar_Responsable()	
Modificar_Responsable()	
Eliminar_Responsable()	
Seleccionar_Responsable()	
Descripción	Es la clase encargada de gestionar todo lo referente a los Responsable en el sistema.

Tabla 13: Descripción de la Clase Controladora CC_Gestionar_Responsable.

3.5.8 Clase de acceso a dato Acceso_Dato.

Nombre	Acceso_Datos
Tipo de Clase	DAO
Atributo	Tipo

Host	String
Usser	String
Password	String
Base_Datos	String
Conexion	String
Responsabilidad	
Ejecutar_Consulta()	
Cerrar_Conexion ()	
Fetch_Array()	
Descripción	Clase encargada de realizar la conexión a la base de datos y ejecutar las consultas.

Tabla 14: Descripción de la Clase de acceso a dato Accesos_Dato.

3.5.9 Clase de acceso a dato AD_Reportar_Interrupcion.

Nombre	AD_Reportar_Interrupcion.	
Tipo de Clase	DAO	
Atributo	Tipo	
Responsabilidad		
Insertar_Interrupcion(interrupción)		
Descripción	Clase encargada de controlar el acceso a los datos del caso de uso Reportar Interrupción.	

Tabla 15: Descripción de la Clase de acceso a dato AD_Reportar_Interrupcion.

3.5.10 Clase de acceso a dato AD_Atender_Interrupcion.

Nombre	AD_Atender_Interrupcion.	
Tipo de Clase	DAO	
Atributo	Tipo	

Responsabilidad	
Editar_Interrupcion(interrupción)	
Descripción	Clase encargada de controlar el acceso a los datos del caso de uso Atender Interrupción.

Tabla 16: Descripción de la Clase de acceso a dato AD_Atender_Interrupccion.

3.5.11 Clase de acceso a dato AD_Gestionar_Responsable.

Nombre	AD_Gestionar_Responsable.
Tipo de Clase	DAO
Atributo	Tipo
Responsabilidad	
Insertar_Responsable(responsable)	
Editar_Responsable(responsable)	
Eliminar_Responsable(responsable)	
Descripción	Clase encargada de controlar el acceso a los datos del caso de uso Gestionar Responsable.

Tabla 17: Descripción de la Clase de acceso a dato AD_Gestionar_Responsable

3.5.13 Clase de acceso a dato AD_Gestionar_Software.

Nombre	AD_Gestionar_Software.
Tipo de Clase	DAO
Atributo	Tipo
Responsabilidad	
Insertar_Software_BD(software)	
Editar_Software(software)	
Eliminar_Software(id_software)	

Descripción	Clase encargada de controlar el acceso a los datos del caso de uso Gestionar Software.
--------------------	--

Tabla 18: Descripción de la Clase de acceso a dato AD_Gestionar_Software.

3.5.15 Clase de acceso a dato AD_Asignar_Responsable.

Nombre	AD_Asignar_Responsable.	
Tipo de Clase	DAO	
Atributo	Tipo	
Responsabilidad		
Editar_Interrupcion(interrupción)		
Descripción	Clase encargada de controlar el acceso a los datos del caso de uso Asignar Responsable.	

Tabla 19: Descripción de la Clase de acceso a dato AD_Asignar_Responsable.

3.5.17 Clase entidad CE_Software.

Nombre	CE_Software
Tipo de Clase	Entidad
Atributo	Tipo
Nombre	string
Responsabilidad	
__construct(id_soft,nombre,id_area)	
__destrcut()	
__Set(\$valor1, \$valor2)	
__Get(valor)	
Descripción	Clase entidad que representa la tabla software en la BD.

Tabla 20: Descripción de la Clase entidad CE_Software.

3.5.17 Clase entidad CE_Responsable.

Nombre	CE_Responsable
Tipo de Clase	Entidad
Atributo	Tipo
Name	string
User	string
Responsabilidad	
__construct(id_soft,nombre,id_area)	
__destrcut()	
__Set(\$valor1, \$valor2)	
__Get(valor)	
Descripción	Clase entidad que representa la tabla software en la BD.

Tabla 21: Descripción de la Clase entidad CE_Responsable.

3.5.18 Clase entidad CE_Interrupcion.

Nombre	CE_Interrupcion
Tipo de Clase	Entidad
Atributo	Tipo
Id_interrupcion	String
User_afectado	String
Name_afectado	String
Medio	String
Ip_Medio	String
Subred	String
Área	String
Fecha_Reporte	date
Fecha_atencion	date
Software	Software
User_Responsable	String

Estado	String
Descripción	text
Responsabilidad	
__construct(Id_interrupcion, Fecha_Reporte, Fecha_atencion, Id_area, Medio, Mask_subRed, Id_Responsable, Estado, Descripción)	
__destrcut()	
__Set(\$valor1, \$valor2)	
__Get(valor)	
Descripción	Clase entidad que representa la tabla interrupción en la BD.

Tabla 22: Descripción de la Clase entidad CE_Interrupcion.

3.6 Diagrama de clases persistentes.

Las clases persistentes son capaces de guardar su estado en un medio permanente; lo cual está dado por el almacenamiento físico de la información de la clase, para la copia de seguridad en caso del fracaso del sistema, o para el intercambio de información.

A continuación se muestra el diagrama de clases persistentes del sistema:

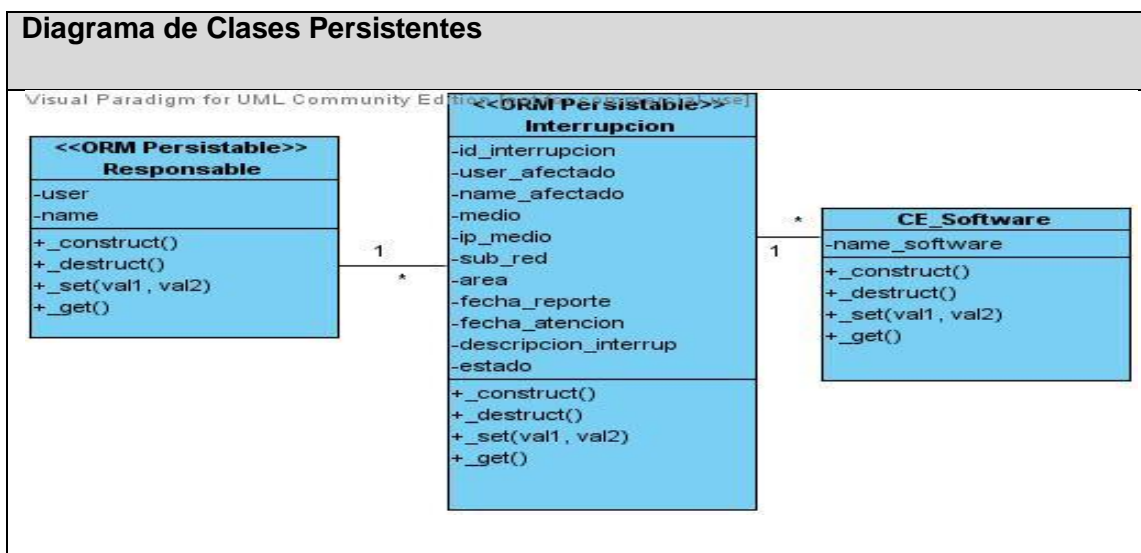


Figura 23: Diagrama de clases persistentes.

3.7 Patrones Utilizados

3.7.1 Patrones de Distribución de Responsabilidades

La aplicación de estos patrones al modelar la arquitectura representa una distribución efectiva de la carga de trabajo entre los objetos. El propósito de los mismos de forma general es originar componentes robustos, entendibles y fáciles de mantener y reutilizar, lo cual explica que su adecuada utilización sea la clave para un exitoso diseño.

Controlador

Se utiliza la misma clase de control para manejar todos los eventos del sistema pertenecientes a un mismo componente.

Explicación: Los subsistemas reciben eventos de entrada externa, los cuales generalmente incluyen una interfaz gráfica para el usuario operado por una persona. Otros medios de entrada son los mensajes externos o las señales procedentes de sensores como sucede en los sistemas de control de procesos. Incluso si no se recurre a un diseño orientado a objetos, es recomendable elegir los controladores que manejen esos eventos de entrada. Este patrón ofrece una guía para tomar decisiones apropiadas que generalmente se aceptan. La misma clase controlador debería utilizarse con todos los eventos sistémicos de un proceso, de modo que se pueda conservar la información referente al estado del caso.

Beneficios: Garantiza que la empresa o los procesos de dominio sean manejados por la capa de los objetos del dominio y no por la de la interfaz. Al delegar a un controlador la responsabilidad de la operación de un sistema entre las clases del dominio favorece la reutilización de la lógica para manejar los procesos afines del negocio en aplicaciones futuras.

3.7.2 Patrón Modelo Vista Controlador (MVC).

El principio más importante de la arquitectura MVC es la separación del código del programa en tres capas, dependiendo de su naturaleza. La lógica relacionada con los

datos se incluye en el modelo, el código de la presentación en la vista y la lógica de la aplicación en el controlador.

Modelo: Es la representación de la información que maneja la aplicación. El modelo en sí son los datos puros que puestos en contexto del sistema proveen de información al usuario y a la aplicación misma.

Vista: Es la representación del modelo en forma gráfica, disponible para la interacción con el usuario. En el caso de una aplicación Web, la "Vista" es una página HTML con contenido dinámico sobre el cual el usuario puede realizar operaciones.

Controlador: Es la capa encargada de manejar y responder las solicitudes del usuario, procesando la información necesaria y modificando el modelo en caso de ser necesario.

Ventajas y desventajas del MVC

Entre las principales ventajas se encuentran:

- La separación del Modelo de la vista, es decir, separar los datos de la representación visual de los mismos.
- Es mucho más sencillo agregar múltiples representaciones de los mismos datos.
- Facilita agregar nuevos tipos de datos según sea requerido por la aplicación ya que son independientes del funcionamiento.
- Facilita el mantenimiento en caso de errores.
- Ofrece maneras más sencillas para probar el correcto funcionamiento del sistema.

Como desventajas

- La separación de conceptos en capas agrega complejidad al sistema.
- La cantidad de archivos a mantener y desarrollar se incrementa considerablemente.
- El aprendizaje del patrón es más lento que otros modelos más sencillos.

3.8 Estándar de codificación

Actualmente se hallan estándares de codificación para la mayoría de los lenguajes de programación existentes. El uso de ellos partiendo de las convenciones definidas permite

una mejor comunicación entre los programadores creando condiciones para la reusabilidad y mantenimiento de los sistemas.

Para definir el estilo de codificación a seguir en la aplicación tomamos como base otros estándares para programación en PHP definiendo el siguiente:

Las etiquetas de apertura y cierre del lenguaje serán de la forma `<?php ?>`, ya que siempre están disponible en cualquier configuración.

Sangrías y Longitud de línea

Usar una sangría de 4 espacios sin usar la tecla “tab”. Esto ayuda a evitar problemas con diffs, patches, historia del CVS y anotaciones.

Estructuras de Control

Estas incluyen: if, for, while, switch, etc. Aquí está el ejemplo de la postura de “if”, porque es el más complicado de todas ellas. [\(Ver Anexo#4\)](#)

Los estatutos de control deben de tener un espacio entre la palabra clave de control y el inicio de paréntesis “(”, para distinguirlos de las llamadas de función.

Deben de utilizarse las llaves en cualquier caso “{ }”, incluso en situaciones donde son técnicamente opcionales. Su uso incrementa su capacidad de lectura y reduce la probabilidad de errores lógicos que son introducidos cuando líneas nuevas se agregan.

Estructuras switch [\(ver Anexo#5\)](#)

Llamadas de Funciones

Las funciones deberá de ser llamadas sin espacios entre el nombre de la función, el abrir paréntesis y el primer parámetro; espacios entre comas y cada parámetro, y sin espacios entre el ultimo parámetro, el cierra paréntesis y punto y coma. [\(Ver Anexo#6\)](#)

Definiciones de Función

En las declaraciones de función, el bracket (llave de apertura "{") de apertura empieza al final de la línea de definición después de un espacio. Y los argumentos con valores por defecto van al final de la lista de argumentos. Siempre tratando de regresar un valor significativo de una función, si una es la apropiada. ([Ver Anexo#7](#))

Inclusión de Código

En cualquier parte que estés incondicionalmente incluyendo un archivo de una clase, usa `required_once`. En cualquier parte donde estés condicionalmente incluyendo un archivo de clase (por ejemplo, métodos de fábrica) usa `include_once`. Cualquiera de esas se asegurará que la class file sea incluida únicamente una vez. Ellas comparten la misma lista de archivo (file list) así que no hay necesidad de preocuparse por mezclarlas, un archivo incluido con `required_once` no será nuevamente incluido por `include_once`.

Convención de Nombres de Variables, Funciones, Clases

Variables:

Estas deberán nombrarse con un prefijo de tres letras el cual define el tipo de dato de la misma, seguido de un guión bajo y el nombre descriptivo de la variable. ([Ver Anexo#8](#))

Funciones:

Estas deberán nombrarse con palabras minúsculas seguido de un guión bajo, para separar palabras en el nombre de la misma. ([Ver Anexo#9](#))

Clases:

Estas deberán nombrarse con la primera palabra iniciando con mayúscula y separando las demás palabras con letra mayúscula. ([Ver Anexo#10](#))

3.9 Conclusiones

En este capítulo fueron expuestos diferentes elementos que ilustran cómo está construido el sistema, en términos de clases del análisis y del diseño. Este último dio la posibilidad de comprender la lógica del sistema en general. Por otro lado, fueron detalladas textualmente cada una de las clases Web, lo cual trae consigo que se conozca la responsabilidad de cada una de estas y qué funcionalidad específica realizan, además de que fue presentado el diagrama de clases persistentes de la base de datos, que contiene la información física que se utilizó para la construcción de la aplicación.

CAPITULO 4: IMPLEMENTACIÓN

4.1 Introducción.

Para el desarrollo de este capítulo se tuvieron en cuenta algunos artefactos que fueron generados en el flujo de análisis y diseño. A partir de los cuales se desarrolló la implementación del sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ejecutables y similares.

4.2 Modelo de Implementación.

El modelo de implementación constituye la vista de Implementación de la arquitectura, y como tal guía las labores de construcción del sistema. Este contiene fundamentalmente los subsistemas de implementación, incluyendo las dependencias y otras informaciones necesarias para su utilización.

4.2.1 Diagrama de Componentes.

Para lograr una mejor comprensión de los componentes que forman el sistema, se presentarán los diagramas de componentes que describen los elementos físicos y lógicos del sistema y sus relaciones. Los diagramas que se presentan a continuación tienen como objetivo figurar la estructura general de la aplicación en desarrollo, en términos de componentes.

5.2.1.1 Diagrama de Componentes del CUS Reportar Interrupción.

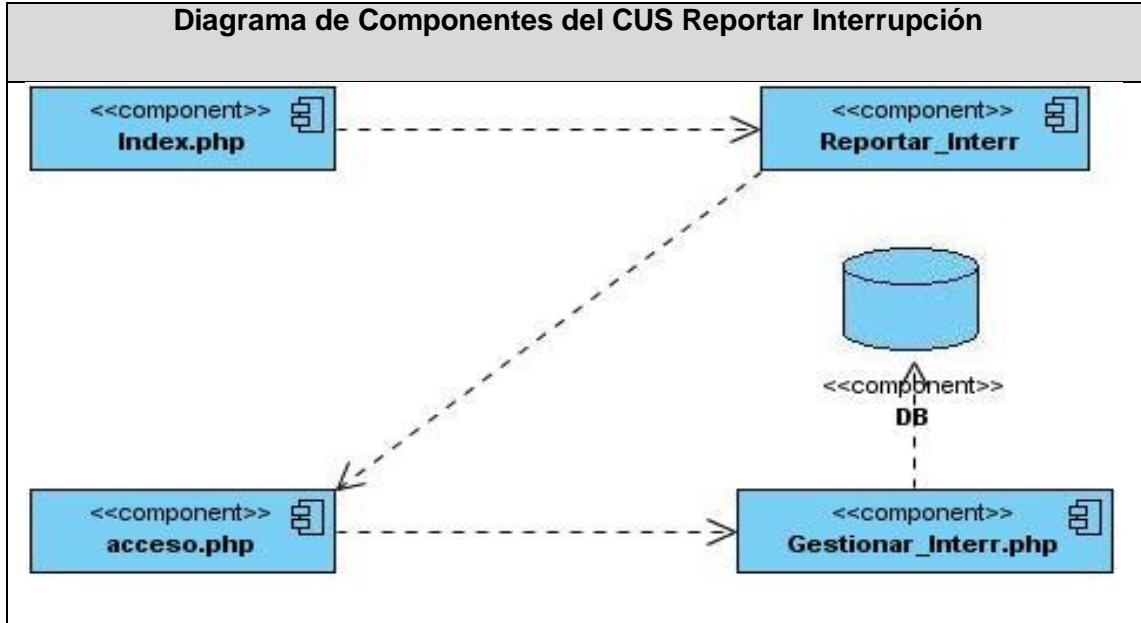


Figura 24: Diagrama de Componentes del CUS Reportar Interrupción.

5.2.1.2 Diagrama de Componentes del CUS Atender Interrupción.

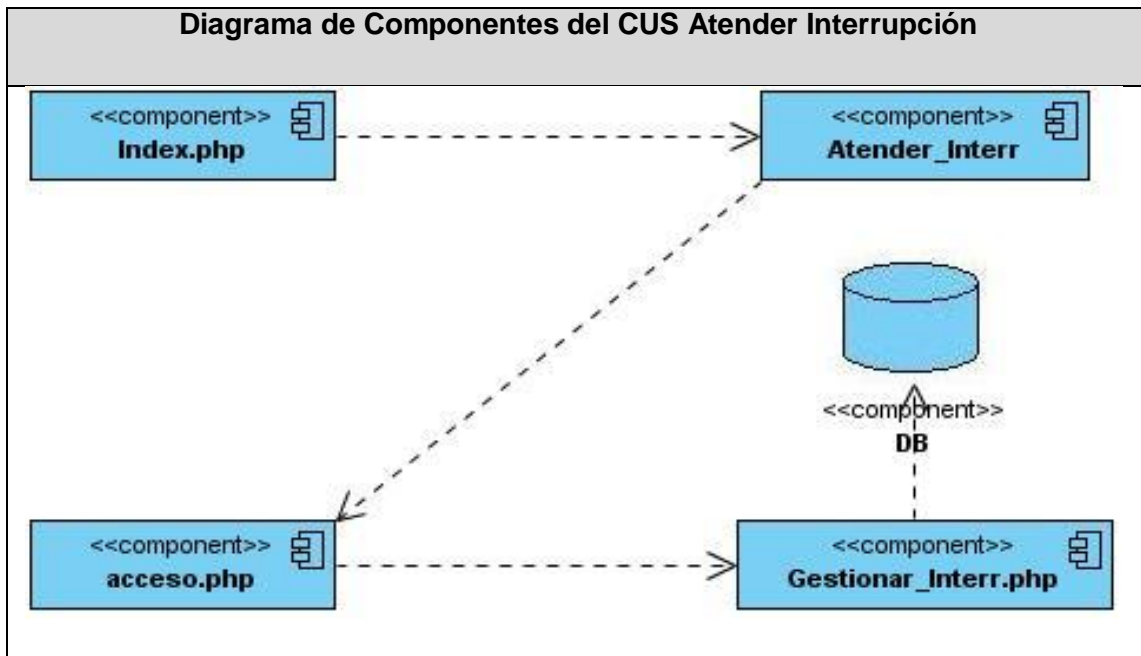


Figura 25: Diagrama de Componentes del CUS Atender Interrupción.

5.2.1.3 Diagrama de Componentes del CUS Asignar Responsable.

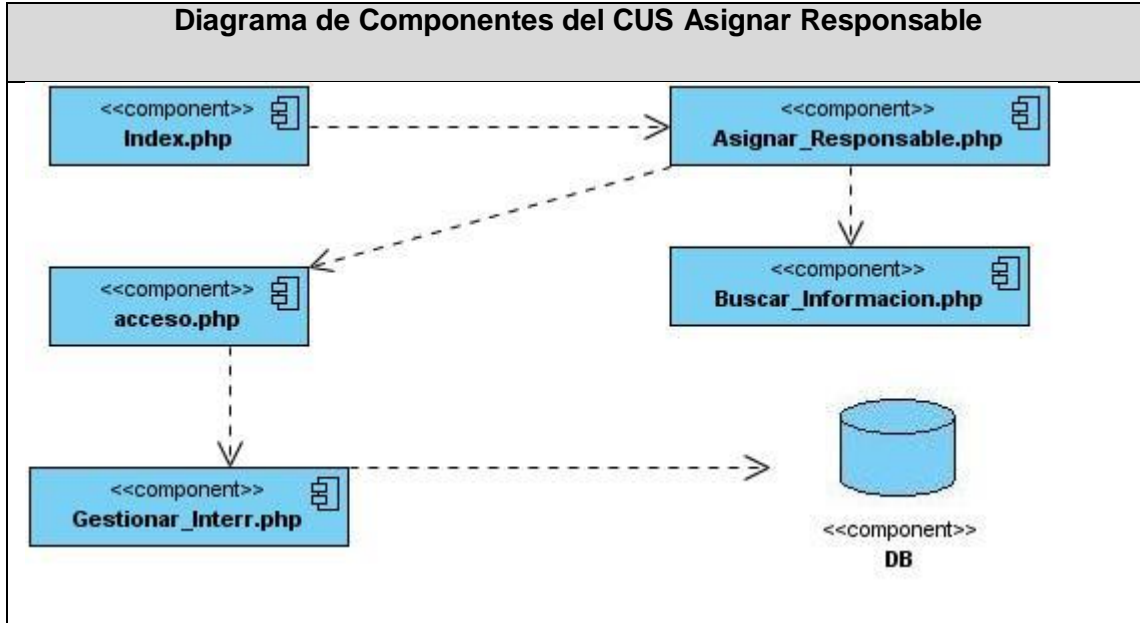


Figura 26: Diagrama de Componentes del CUS Asignar Responsable.

5.2.1.4 Diagrama de Componentes del CUS Obtener Reporte.

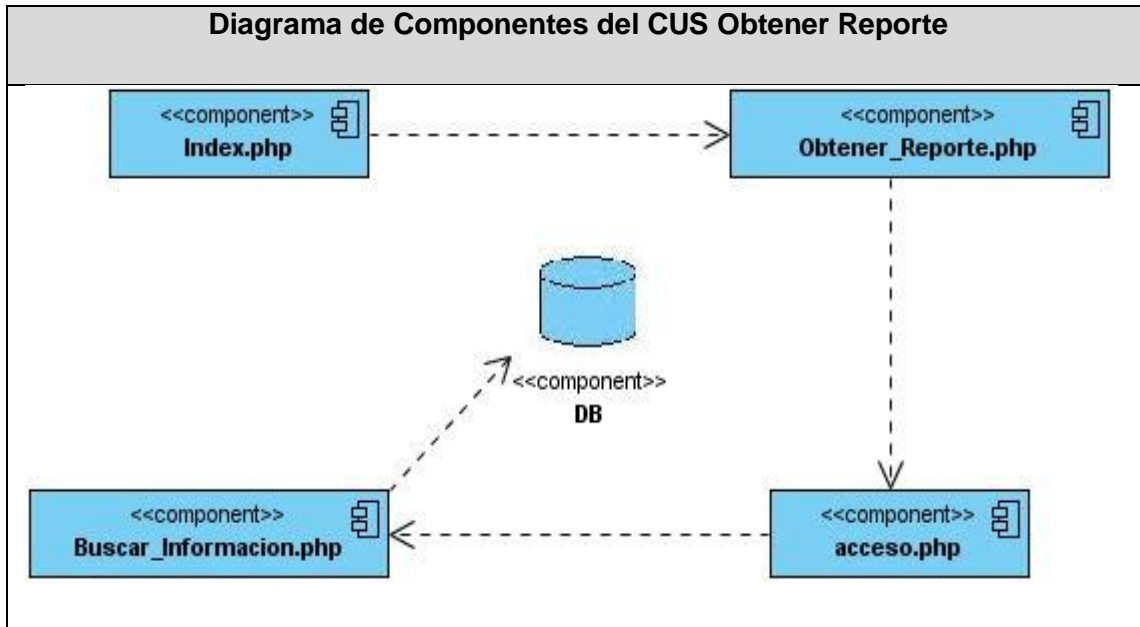


Figura 27: Diagrama de Componentes del CUS Obtener Reporte.

5.2.1.5 Diagrama de Componentes del CUS Gestionar Interrupción.

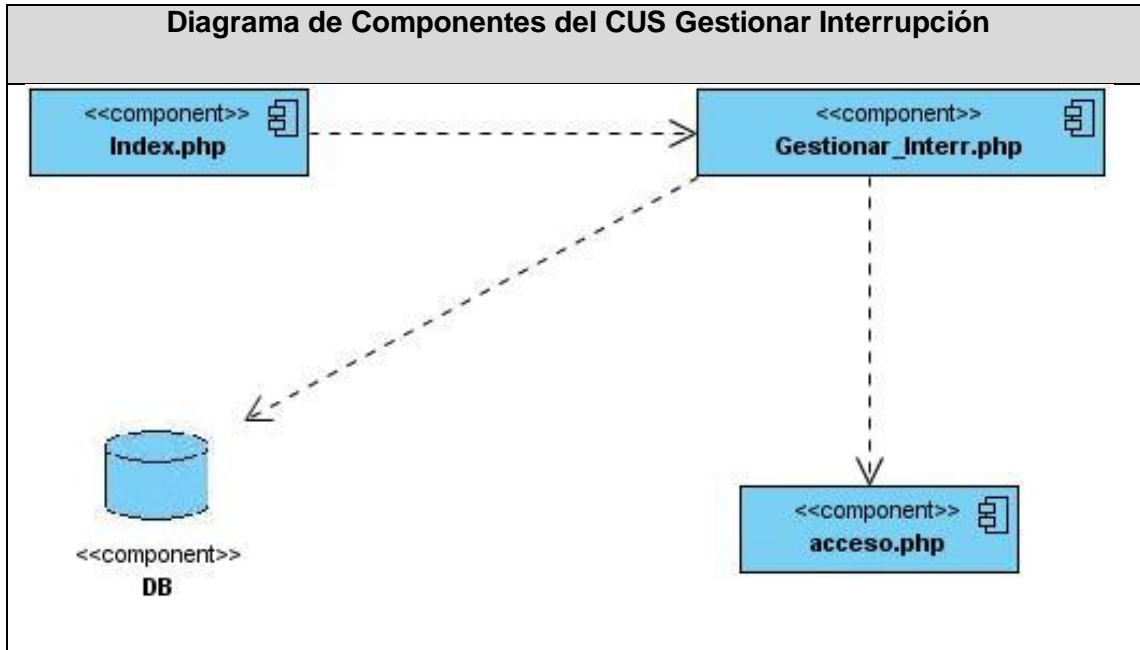


Figura 28: Diagrama de Componentes del CUS Gestionar Interrupción.

5.2.1.6 Diagrama de Componentes del CUS Gestionar Responsable.

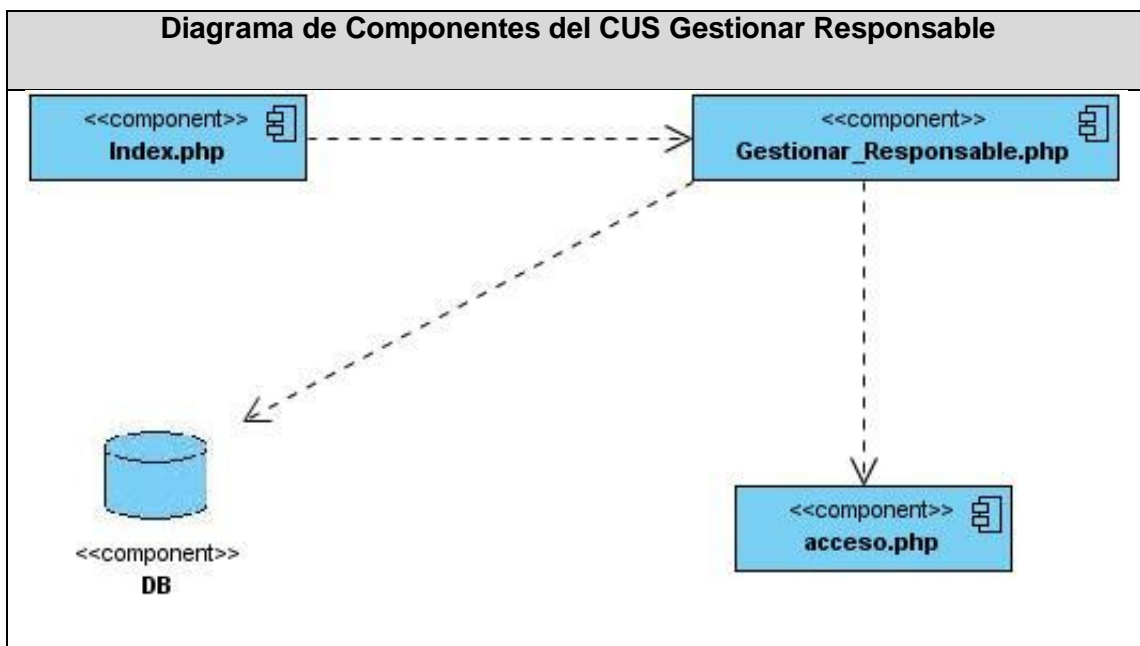


Figura 29: Diagrama de Componentes del CUS Gestionar Responsable.

5.2.1.7 Diagrama de Componentes del CUS Gestionar Software.

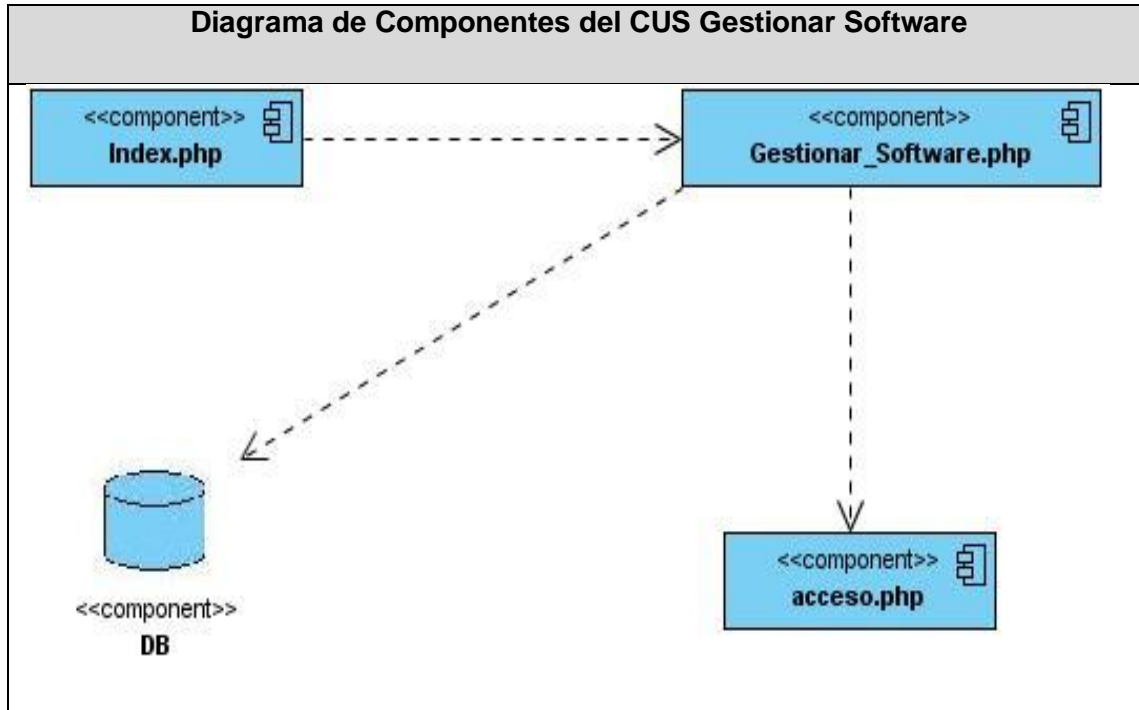


Figura 30: Diagrama de Componentes del CUS Gestionar Software.

4.3 Modelo de Despliegue.

El diagrama de despliegue representa la arquitectura de tiempo de ejecución de los procesadores, dispositivos y los componentes de software que se ejecutan en esa arquitectura. Es la última descripción física de la topología del sistema y describe la estructura de las unidades de hardware. Además, representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un procesador, un dispositivo o memoria. En los procesadores es donde se encuentran alojados los componentes.

4.3.1 Diagrama de Despliegue.

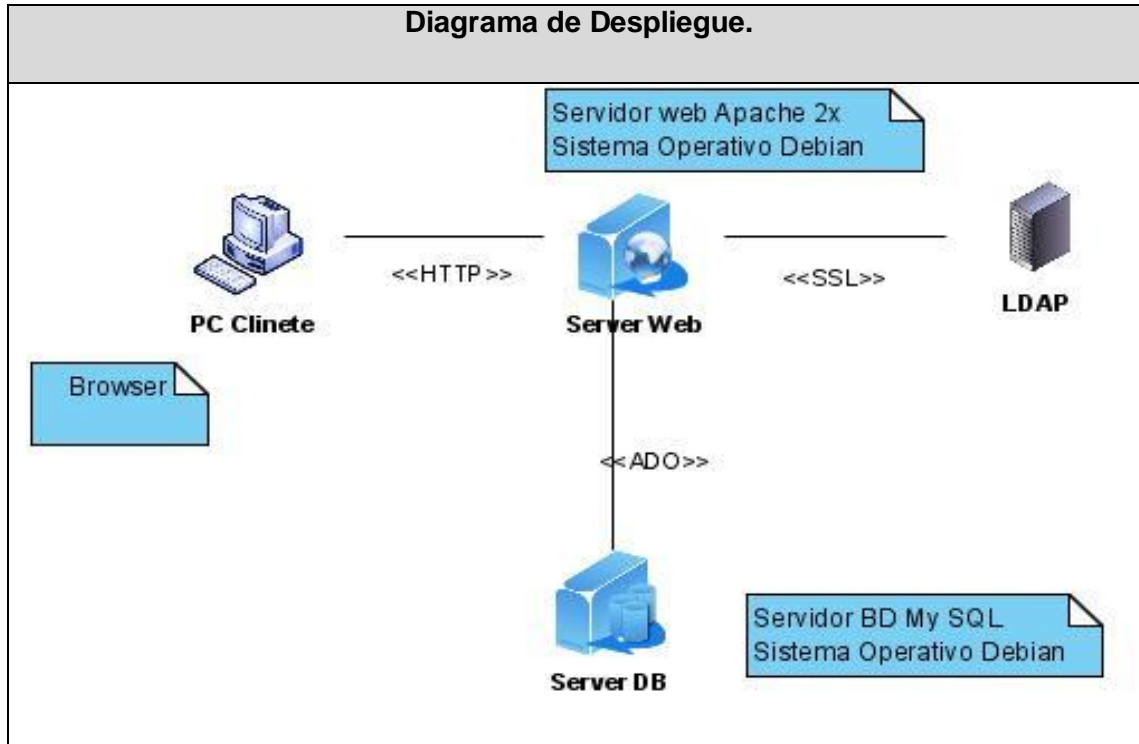


Figura 31: Diagrama de Despliegue.

4.4 Estudio de Factibilidad

Este estudio se realiza utilizando el método de análisis por Puntos de Casos de Uso para estimar el esfuerzo humano, el tiempo de desarrollo que se empleara así como su costo.

4.4.1 Planificación mediante el método de análisis por Puntos de Casos de Uso.

Fue propuesto originalmente por Gustav Kerner de Objector y posteriormente refinado por muchos otros autores. Se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores.

Cálculo de Puntos de Caso de Uso sin ajustar.

$$UUCP = UAW + UUCW$$

Donde:

UUCP: Puntos de Casos de Uso sin ajustar.

UAW: Factor de Peso de los Actores sin ajustar.

UUCW: Factor de Peso de los Casos de Uso sin ajustar.

Calculando UAW: Factor de Peso de los actores.

Este valor se calcula mediante un análisis de la cantidad de Actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Actores se establece teniendo en cuenta en primer lugar si se trata de una persona o de otro sistema, y en segundo lugar, la forma en la que el actor interactúa con el sistema.

Tipo de Actor	Descripción	Peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, Application Programming Interface).	1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3

Tabla 23: Factor de peso de los actores.

Actores de tipo complejo en el sistema desarrollado:

Trabajador Afectado, Trabajador Responsable y Jede de la UGAA.

Actores de tipo medio:

Servidor LDAP.

$$UAW = \sum \text{Peso de los actores.}$$

$$UAW = 3*3+2$$

$$UAW= 11$$

Calculando UUCW: Factor de Peso de los Casos de Uso sin ajustar.

Este valor se calcula mediante un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Casos de Uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo. Una transacción está representada por uno o más pasos del flujo de eventos principal del Caso de Uso, pudiendo existir más de una transacción dentro del mismo Caso de Uso.

Tipo de Caso de Uso	Descripción	Peso
Simple	El Caso de Uso contiene de 1 a 3 transacciones.	5
Medio	El Caso de Uso contiene de 4 a 7 transacciones.	10
Complejo	El Caso de Uso contiene más de 8 transacciones.	15

Tabla 24: Factor de Peso de los Casos de Uso sin ajustar.

Caso de Uso	Transacciones	Peso
Autenticarse	2	5
Reportar Interrupción	3	5
Atender Interrupción	3	5
Gestionar Interrupción	4	10
Gestionar Responsable	4	10
Gestionar Software	4	10
Asignar Responsable	3	5
Obtener Reporte	4	10

Tabla 25: Transacciones y peso de los casos de uso.

$$\text{UUCW} = \sum \text{Peso de Caso de Uso sin ajustar}$$

$$\text{UUCW} = 5 + 5 + 5 + 10 + 10 + 10 + 5 + 10$$

$$\text{UUCW} = 60$$

Teniendo el resultado de **UAW** y de **UUCW** procedemos al **Cálculo de Puntos de Caso de Uso sin ajustar.**

$$UUCP = UAW + UUCW$$

$$UUCP = 11 + 60$$

$$UUCP = 71$$

Calculando lo Puntos de Caso de Uso ajustados.

Los puntos de Casos de Uso ajustados se calculan con la siguiente fórmula:

$$UCP = UUCP * TCF * EF$$

Donde:

UCP: Puntos de Casos de Uso ajustados

UUCP: Puntos de Casos de Uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor ambiente.

Calculando TCF: Factor de complejidad técnica.

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores (**T**) que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante. En la siguiente tabla se muestran los factores, el peso de cada uno de ellos y el valor asignado:

Factor	Descripción	Peso	Valor asignado
T1	Sistema distribuido	2	0
T2	Tiempo de respuesta	1	3
T3	Eficiencia del usuario final	1	3
T4	Procesamiento interno complejo	1	2
T5	El código debe ser reutilizable	1	1
T6	Facilidad de instalación	0.5	1
T7	Facilidad de uso	0.5	4
T8	Portabilidad	2	2
T9	Facilidad de cambio	1	3

T10	Concurrencia	1	4
T11	Incluye objetivos especiales de seguridad	1	3
T12	Provee acceso directo a terceras partes	1	0
T13	Se requieren facilidades especiales de entrenamiento a usuarios	1	3

Tabla 26: Factor de complejidad técnica.

$$TCF = 0.6 + 0.01 * \Sigma (\text{Peso} * \text{Valor Asignado})$$

$$TCF = 0.6 + 0.01 * 28.5$$

$$TCF = 0.885$$

Calculando EF: Factor de ambiente.

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores (**E**) son los que se contemplan en el cálculo del Factor de ambiente. El cálculo del mismo es similar al cálculo del Factor de complejidad técnica, es decir, se trata de un conjunto de factores que se cuantifican con valores de 0 a 5.

Factor	Descripción	Peso	Valor asignado
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	3
E2	Experiencia en la aplicación.	0.5	3
E2	Experiencia en orientación a objetos	1	4
E4	Capacidad del analista líder.	0.5	4
E5	Motivación.	1	5
E6	Estabilidad de los requerimientos.	2	4
E7	Personal part-time.	-1	4
E8	Dificultad del lenguaje de programación.	-1	2

Tabla 27: Factor de ambiente.

$$EF = 1.4 - 0.03 * \Sigma (\text{Peso} * \text{Valor Asignado})$$

$$EF = 1.4 - 0.03 * 31$$

$$EF = 0.47$$

Calculando los Puntos de Caso de Uso ajustados

$$\text{UCP} = \text{UUCP} * \text{TCF} * \text{EF}$$

$$\text{UCP} = 71 * 0.885 * 0.47$$

$$\text{UCP} = 29.53$$

De los Puntos de Casos de Uso a la estimación del esfuerzo.

El esfuerzo en horas-hombre viene dado por:

$$\text{E} = \text{UCP} * \text{CF}$$

Donde:

E: Esfuerzo estimado en horas-hombre.

UCP: Puntos de Casos de Uso ajustados.

CF: Factor de conversión.

Para calcular el factor de conversión es necesario tener como guía el siguiente criterio:

- Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por debajo del valor medio (3), para los factores E1 a E6.
- Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por encima del valor medio (3), para los factores E7 y E8.
- Si el total es 2 o menos, se utiliza el factor de conversión 20 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 20 horas-hombre.
- Si el total es 3 o 4, se utiliza el factor de conversión 28 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 28 horas-hombre.
- Si el total es mayor o igual que 5, se recomienda efectuar cambios en el proyecto, ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

Resumiendo:

CF = 20 horas-hombre (si Total EF ≤ 2)

CF = 28 horas-hombre (si Total EF = 3 ó Total EF = 4)

CF = abandonar o cambiar proyecto (si Total EF \geq 5)

En el presente caso no existen factores por debajo de 3 desde E1 hasta E6 y solo un factor de E7 y E8 está por encima de 3, por lo que el total de factores a tener en cuenta es 1 y el factor conversión sería de 20 horas-hombre.

Calculando el esfuerzo del flujo de trabajo:

$$E = UCP * CF$$

$$E = 29.53 * 20$$

$$E = 591 \text{ horas-hombre.}$$

Se debe tener en cuenta que éste método proporciona una estimación del esfuerzo en horas-hombre contemplando sólo el desarrollo de la funcionalidad especificada en los casos de uso. Finalmente, para una estimación más completa de la duración total del producto, hay que agregar a la estimación del esfuerzo obtenida por los Puntos de Casos de Uso, las estimaciones de esfuerzo de las demás actividades relacionadas con el desarrollo de software. Además se considera que este esfuerzo representa un porcentaje del esfuerzo total del proyecto, de acuerdo a los valores porcentuales:

Actividad	Porcentaje	Horas/Hombre
Análisis	12.7 %	157.8
Diseño	30 %	374.4
Implementación	47.3 %	591
Sobrecarga (otras actividades)	10 %	124.8
Total	100 %	1248

Tabla 28: Distribución por ciento-horas-hombre.

Convirtiendo a Horas/Hombre.

Con una jornada laboral de 8 horas como lo establece la institución y en un mes se trabajan 24 días aproximadamente, por ende una persona ajustada a este horario en 1 mes trabaja 192 horas, por tanto:

$E_t = \text{Total (horas-hombre)} / 192 \text{ horas-mes}$, Obteniéndose un $E_t = 1248 \text{ horas-hombre} / 192 \text{ horas-mes}$.

Quedando $E_t = 6.5 \text{ mes-hombre}$

Por tanto como la fuerza de trabaja recae sobre 1 hombre el tiempo de desarrollo será de 6 meses y 2 semanas aproximadamente.

Salario

Para determinar el salario mensual se tiene en cuenta que los desarrolladores del sistema pueden ser ingenieros recién graduados, por lo que se toma como salario mensual: \$225.

4.4.2 Costo.

Como se definió anteriormente el salario promedio de un ingeniero, y es una persona el desarrollador del proyecto, entonces para hallar el costo total sería:

$C_t = \text{Salario mensual} * \text{Cantidad de hombres} * \text{Tiempo de desarrollo}$

$C_t = \$225 * 1 * 6.5$

$C_t = \$1462.5$

4.4.2.1 Tangibles.

Teniendo en cuenta que la aplicación en cuestión no es un producto desarrollado para la comercialización, ya que surge con la idea de apoyar a la **Dirección Territorial de ETECSA en Granma**, específicamente, a la **Unidad de Gestión de Administración de Aplicaciones (UGAA)** no es válido mencionar beneficios económicos. Se puede decir que el costo por desarrollar la aplicación es de \$1462.5 MN (moneda nacional) ó \$60.93 CUC (convertible), el cual es perfectamente reparable si en un futuro se comercializara.

4.4.2.2 Intangibles.

Como beneficios intangibles de la aplicación se señalan los siguientes:

- Se agiliza el proceso de reporte de interrupciones.

- Se evita la congestión de las líneas telefónicas.
- Permite garantizar una mayor centralización de la información que se maneja en todo este proceso.
- Se optimiza el tiempo de atención a las interrupciones.
- Se minimizan los costos de los recursos implicados en este proceso.
- Se realizan de forma automatizada todas las actividades que normalmente los trabajadores afectados, responsables y el Jefe de la UGAA realizan de forma manual.

4.4.3 Análisis costo-beneficio.

Dadas las características de la aplicación se puede decir que se requiere de un tiempo aceptable para el desarrollo del software, además no necesita de grandes gastos de recursos. El ambiente del software se ve favorecido con un diseño que permite al usuario desarrollar una perfecta navegabilidad sin motivos de perderse dentro del mismo. Es factible desarrollar una aplicación para agilizar y organizar todo el proceso de gestión y control de las interrupciones de software.

4.4 Conclusiones.

En este capítulo se mostró como a través de la implementación, se produjo un refinamiento de la vista de la arquitectura del modelo de despliegue, donde los componentes ejecutables fueron asignados a nodos. Se utilizaron diagramas de componentes para representar a través de un grafo los componentes de software unidos por medio de relaciones de dependencia; con los cuales se modeló la vista estática de un sistema. Además sirvieron para mostrar la organización y las dependencias lógicas entre un conjunto de componentes software. Además se realizó un estudio de la factibilidad del producto, se puntualizan los costos a incurrir, los recursos humanos implicados, el tiempo de desarrollo y los beneficios que aporta la terminación del producto "Sistema de Gestión y Control de Interrupciones de Software en la Dirección Territorial de ETECSA.

CONCLUSIONES

- Se detectó la inexistencia de un sistema automatizado para controlar y gestionar las interrupciones de software que ocurren en la Dirección Territorial de ETECSA en Granma.
- Se obtiene la propuesta de una aplicación que informatizará el procesamiento manual de la gestión y control de interrupciones de software que realiza la Dirección Territorial de ETECSA.
- Se realizó el análisis y diseño del sistema.
- Se demostró la factibilidad de desarrollar la aplicación para agilizar y organizar todo el proceso de gestión y control de las interrupciones de software que se producen en el territorio.
- Se desarrolló un sistema que permite gestionar y controlar las interrupciones de software en la Dirección Territorial de ETECSA en Granma.

RECOMENDACIONES

Los objetivos de este trabajo han sido logrados, teniendo en cuenta que se cumplieron todos los requisitos. No obstante se recomienda lo siguiente:

- 1- Implantar la aplicación en las demás Direcciones Territoriales del país.
- 2- Migrar la base de datos a PostgreSQL pues actualmente se estudia para futura migración de todas las base de datos de la institución.
- 3- Implementar una funcionalidad capaz de calcular el indicador de pago.
- 4- Agregar otras funcionalidades como la gestión de las interrupciones de hardware.

BIBLIOGRAFIA

1. **Manual de usuario.** [En línea] <http://www.osiptel.gob.pe>
2. **Ficha Técnica AvilaQuid.** [En línea]
<http://www.desoft.cu/Portals/0/FichaTecnicaAvilaQuid.pdf>
3. **AvilaQuid.** [En línea]
www.desoft.cu/Productos1/AvilaQuid/tabid/432/Default.aspx?PageContentID=7
4. **AvilaQuid.** [En línea]
www.informaticahabana.com/evento_virtual/files/GOB035.doc
5. **SisMon.** [En línea]
www.bibliociencias.cu/gsd/collect/eventos/index/assoc/HASH4e09.dir/doc.pdf
6. **Aplicación Web.** [En línea] http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web
7. **Cliente - Servidor.** [En línea] <http://es.wikipedia.org/wiki/Cliente-servidor>
8. **Lenguajes del lado servidor o cliente.** [En línea]
http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html
9. **El Lenguaje HTML.** [En línea] <http://www.webestilo.com/html/cap2a.phtml>
10. **Rodríguez, Daniel y Bravo, Joaquín. “Tutorial de HTML”.** [En línea]
<http://html.programacion.net>.
11. **Introducción a XML para Documentalistas.** [En línea]
<http://www.hipertext.net/web/pag100.htm>
12. **Hojas de Estilo en Cascada, nivel 1.** [En línea]
<http://html.conclase.net/w3c/css1-es.html>
13. **Tecnologías.** [En línea] <http://www.alexandria.com.mx/tecnologias.php>
14. **PHP en castellano.** [En línea] <http://www.programacion.com/php/direcciones>
15. **Aulbach, Alexander “Manual de PHP”.** [En línea]
http://www1.us.es/pautadatos/publico/personal/pas/2166/1498/php_manual.pdf
16. **PHP 5 Fast & Easy Web Development.** [En línea] <http://www.amazon.com/PHP-Fast-Easy-Web-Development/dp/1592004733#reader>
17. **Miguel Angel Alvarez Manual de PHP 5.** [En línea]
<http://www.desarrolloweb.com/manuales/58>

18. **Joel González Estrada Desarrollo Web con PHP y MySQL.** [En línea]
http://grupos.emagister.com/documento/manual_de_php/1080-10581
19. **Python instantáneo.** [En línea] <http://www.arraakis.es/~rpto/AprendaPython.html>
1999.
20. **Python.** [En línea] <http://es.wikipedia.org/wiki/Python>
21. **Perl.** [En línea] <http://es.wikipedia.org/wiki/Perl>
22. **Sistema de gestión de Bases de Datos.** [En línea]
<http://es.wikipedia.org/wiki/DBMS>
23. **Sistemas de bases de datos y SGBD.** [En línea]
<http://www.tramullas.com/nautica/documatica/2-2.html>.
24. Date, C.J., “**Introducción a los Sistemas de Bases de Datos**”, 7ª Edición,
Addison Wesley, 2001.
25. **Historia de los sistemas de bases de datos.** [En línea]
<http://www3.uji.es/~mmarques/f47/apun/node6.html>
26. **Postgre.** [En línea] <http://es.wikipedia.org/wiki/Postgre>
27. **MySQL.** [En línea] <http://es.wikipedia.org/wiki/Mysql>
28. **Servidor HTTP Apache.** [En línea]
http://es.wikipedia.org/wiki/Servidor_HTTP_Apache.
29. **HyperText Transfer Protocol.** [En línea] <http://es.wikipedia.org/wiki/HTTP>
30. **HyperText Transfer Protocol Secure.** [En línea]
<http://es.wikipedia.org/wiki/HTTPS>
31. **Internet Information Server.** [En línea] <http://es.wikipedia.org/wiki/IIS>
32. **Metodologías de Desarrollo de Software (MDS).** [En línea]
<http://es.geocities.com/gustsucc/Archivos/metodologiaRUP.pdf>
33. **Molpeceres, Alberto.** Artículo: “Procesos de desarrollo: RUP, XP y FDD”, 2002.
[En línea] <http://www.javahispano.org/articles.article.action?id=76>
34. **El Proceso Unificado Racional.** [En línea] <http://es.wikipedia.org/wiki/RUP>
35. **Herramientas Case.** [En línea]
<http://www.monografias.com/trabajos14/herramicase/herramicase.shtml>

36. **Visual Paradigm for UML (ME) - (Paradigma Visual para UML (ME)) (Visual Paradigm for UML (ME)) 6.0.** [En línea]
[http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/)
37. **Vizcaíno, Aurora; García, Félix Oscar; Del Caballero, Ismael.** Prácticas Ingeniería del Software, Una Herramienta CASE para ADOO: Visual Paradigm Análisis y Diseño Orientado a Objetos.2008.
38. **“Tutorial de UML”.** [En línea] <http://www.dcc.uchile.cl/~psalinas/uml>
39. **Lenguaje Unificado de Modelado.** [En línea] <http://es.wikipedia.org/wiki/UML>
40. **Active Komodo.** [En línea]
<http://www.blinklist.com/tag/activestate%20komodo%20ide%205.1.0.27487>
41. **Zend Studio for Eclipse, desarrollo profesional en PHP.** [En línea]
<http://www.techtear.com/2008/01/22/zend-studio-for-eclipse-desarrollo-profesional-en-php/>
42. **Los mejores IDEs para PHP.** [En línea] <http://www.buenmaster.com/?a=1203>
43. **Sistemas de Gestión de Contenidos (CMS).** [En línea]
<http://www.capsoft.com.ar/cms.asp>
44. **Joomla Foro “Joomla en Español”.** [En línea]
www.cmsenespanol.com/modules/news/index.php?storytopic=0&start=0
45. **Sistema de Gestión de Contenidos (Libres), Plone.** [En línea]
http://nibbler.es/_oneclick_uploads/2009/02/cms-iiijornadas.pdf
46. **Características de Drupal.** [En línea] <http://drupal.org/es/caracteristicas>
47. **Jacobson, Ivar; Booch, Grady; Rumbaugh, James.** “El proceso unificado del desarrollo de software (Tomo I)”. Editorial “Félix Varela”, 2004. No. Páginas 437.
48. **Jacobson, Booch, Rumbaugh** "El Proceso Unificado de Desarrollo de Software". España. Editorial: Addison Wesley.
49. **Pressman, Roger S.** Ingeniería de Software, un enfoque práctico. Parte 1. La Habana, Cuba. Editorial Félix Varela. 2005.
50. **Pressman, Roger S.** Ingeniería de Software, un enfoque práctico. Parte 2. La Habana, Cuba. Editorial Félix Varela. 2005.

ANEXOS

Anexo #1: Ventajas y desventajas de los lenguajes de programación.

Lenguaje de programación	Ventajas	Desventajas
Phyton	<ol style="list-style-type: none"> 1. Multipropósito (no solo para web). 2. Programación Orientada a Objetos. 3. Multiplataforma. 	<ol style="list-style-type: none"> 1. Más lento
Perl	<ol style="list-style-type: none"> 1. Soporta diferentes paradigmas. 2. Elaboración de programas de manera sencillas. 3. Lenguaje extensible. 4. Multiplataforma. 	<ol style="list-style-type: none"> 1. Lentitud. 2. Programa ilegible.
PHP	<ol style="list-style-type: none"> 1. Programación Orientada a Objetos. 2. Sencillo de aprender. 3. Es libre. 4. Capacidad de conexión. 5. Multiplataforma. 6. Suficientemente versátil y potente. 	<ol style="list-style-type: none"> 1. Todo el trabajo lo realiza el servidor. 2. Legibilidad del código.

Anexo #2: Ventajas y desventajas de los gestores de base de datos.

Gestores de BD	Ventajas	Desventajas
PostgreSQL	<ol style="list-style-type: none"> 1. Soporta transacciones y claves ajenas. 2. Soporte para triggers y procedimientos en el servidor. 3. Soporta un subconjunto de SQL 92. 4. Es una herramienta para la construcción de BD potentes. 	<ol style="list-style-type: none"> 5. Consume más recursos y carga más al sistema. 6. Más lento. 7. Menos funciones en PHP.
MySQL	<ol style="list-style-type: none"> 1. Mayor rendimiento y mayor velocidad. 2. Mejor utilidades de administración. 3. Seguridad de la información. 4. Mejor control de acceso. 5. No tiene límite en el tamaño de los registros 	<ol style="list-style-type: none"> 6. No soporta transacciones ni subselect. 7. No considera las claves ajenas. 8. Ignora la integridad referencial.

Anexo #3: Ventajas y desventajas de las Metodologías de desarrollo de software.

Metodologías	Ventajas	Desventajas
Extreme Programming (XP)	<ol style="list-style-type: none"> 1. Constituye una metodología ágil. 2. Estimaciones más acertadas. 3. Es flexible. 4. No requiere documento formal. 	<ol style="list-style-type: none"> 1. Exigen documentación. 2. No tiene etapa de arquitectura. 3. Presencia de cliente On-Site. 4. Dependencias entre componentes.
RUP	<ol style="list-style-type: none"> 1. Más utilizada junto con el lenguaje UML. 2. Procesos organizados. 3. Gestión de requisitos y desarrollos interactivos. 4. Uso de arquitectura basadas en componentes. 5. Calidad de software. 	<ol style="list-style-type: none"> 1. Duración extendida. 2. Se genera documentación sin valor.

Anexo #4: Estructuras de Control.

PHP:

```
1. <?php
2. if ((condición1) || (condición2)) {
3.     acción1;
4. } elseif ((condición 3) && (condición 4)) {
5.     acción2;
6. } else {
7.     acciónXdefecto;
8. }
9. ?>
```

Anexo #5: Estructura Switch.

PHP:

```
1. <?php
2. switch (condición) {
3.     case 1:
4.         acción1;
5.         break;
6.     case 2:
7.         acción2;
8.         break;
9.     default:
10.         acciónXdefecto;
11.         break;
12. }
13. ?>
```

Anexo #6: Llamadas de Funciones.

PHP:

```
1. <?php
2. $var = función($bar, $baz, $quux);
3. ?>
```

Anexo #7: Definición de Funciones.

PHP:

```
1. <?php
2. function connect(&$dsn, $persistent = false) {
3.     if (is_array($dsn)) {
4.         $dsninfo = &$dsn;
5.     } else {
6.         $dsninfo = DB::parseDSN($dsn);
7.     }
8.     if (!$dsninfo || !$dsninfo['phptype']) {
9.         return $this->raiseError();
10.    }
11.    return true;
12. }
13. ?>
```

Anexo #8: Convención de Nombres de Variables

PHP:

1. `$txt_descripcion`: Campo tipo text
2. `$chr_nombres`: Campo tipo char o varchar
3. `$int_cantidad`: Campo tipo integer
4. `$dtm_fecha`: Campo tipo fecha
5. `$bol_logueado`: Campo tipo boolean

Anexo #9: Convención de Nombres de Funciones.

PHP:

```
1. <?php
2. function conectar_db($arg1, $arg2 = 'x') {
3.     if (condición) {
4.         algo;
5.     }
6.     return $val;
7. }
8. ?>
```

Anexo #10: Convención de Nombres de Clases.

PHP:

```
1. <?php
2. class ClaseEstandar {
3.     function función($arg1) {
4.         algo;
5.     }
6. }
7. ?>
```

GLOSARIO DE TERMINOS

Apache: Servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP y la noción de sitio virtual.

Aplicación WEB: Es un sistema informático que los usuarios utilizan accediendo a un servidor Web a través de internet o de una intranet.

Browser: Es un navegador web o explorador web (del inglés, *navigator* o *browser*) es una aplicación software que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores web de todo el mundo a través de Internet.

CASE: Las Herramientas CASE (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

CMS: Content Management System (Sistema de Gestión de Contenidos), son sistemas usados para la construcción de aplicaciones que gestionan contenido.

CSS: Del inglés Cascade Style Sheet (Hoja de Estilo en Cascada) Dentro del diseño de páginas de Internet se presenta como la vanguardia en cuanto a definición de estilos dentro de plantillas de diseño.

CUN: Caso de uso del negocio.

CUS: Caso de uso del Sistema.

ETECSA: Empresa de Telecomunicaciones de Cuba.

HTML: Del inglés de HyperText Markup Language (lenguaje de marcas hipertextuales), lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

Internet: Sistema de redes de computación ligadas entre sí, con alcance mundial, que facilita servicios de comunicación de datos como registro remoto, transferencia de archivos, correo electrónico y grupos de noticias. Internet es una forma de conectar las redes de computación existentes que amplía en gran medida el alcance de cada sistema participante.

Intranet: Es una red de computadoras dentro de una red de área local (LAN) privada, empresarial o educativa que proporciona herramientas de internet.

Interrupción: Errores que provocan mal funcionamiento de las tareas.

LDAP: Del inglés, *Lightweight Directory Access Protocol*, es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar información en un entorno de red. LDAP también es considerado una base de datos al que pueden realizarse consultas.

Oracle: Sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), fabricado por Oracle Corporation.

Perl: Lenguaje Práctico para la Extracción e Informe. Es un lenguaje de programación que toma características del C, del lenguaje interpretado shell (sh), AWK, sed, Lisp y, en un grado inferior, muchos otros lenguajes de programación.

PostgreSQL: Es un motor de base de datos, es servidor de base de datos relacional libre, liberado bajo la licencia BSD.

Python: Python es un lenguaje interpretado, lo que ahorra un tiempo considerable en el desarrollo del programa, pues no es necesario compilar ni enlazar. Es considerado como la "oposición leal" a Perl, lenguaje con el cual mantiene una rivalidad amistosa. Los usuarios de Python consideran a éste mucho más limpio y elegante para programar.

RUP: *Rational Unified Process* (Proceso Unificado de desarrollo). Metodología para el

desarrollo de Software.

Scripts: Se conoce como lenguaje interpretado a un lenguaje de programación que fue diseñado para ser ejecutado por medio de un intérprete, en contraste con los lenguajes compilados.

Servidor Web: Es un programa que implementa el protocolo HTTP (hypertext transfer protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas.

SGBD: *Sistema de Gestión de Bases de Datos.* Es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez.

Software: Se refiere al equipamiento lógico o soporte lógico de un computador digital, comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de una tarea específica.

Trabajador Afectado: Trabajador que se le imposibilita realizar sus tareas por interrupciones que presente.

Trabajador Responsable: Persona calificada y preparada con permisos necesarios para darle solución a las interrupciones.

TCP/IP: Es un conjunto de protocolos de red que implementa la pila de protocolos en la que se basa Internet y que permiten la transmisión de datos entre redes de computadoras. En ocasiones se la denomina *conjunto de protocolos TCP/IP*, en referencia a los dos protocolos más importantes que la componen: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP), que fueron los dos primeros en definirse, y que son los más utilizados de la familia.

TIC: Tecnologías de la Informática y las Comunicaciones.

UCI: Universidad de las Ciencias Informáticas.

UGAA: Iniciales de Unidad de Gestión de Administración de Aplicaciones.

UML: Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software.

Web: World Wide Web (o la Web) es un sistema de documentos de hipertexto enlazados y accesibles a través de Internet.

WWW: Del inglés, World Wide Web.

XML: *Extensible Markup Language*. Es un lenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium. Orientado principalmente al almacenamiento, procesamiento y transmisión de mensajes.

Zend: Compañía líder de infraestructuras para web; está reconocida internacionalmente como la autoridad actualmente en PHP. Sus fundadores son los diseñadores del PHP v.4 en adelante, actualmente es una compañía líder dentro de la comunidad Open Source.