Facultad 2 Redes y Seguridad Informática

SISTEMA DE GESTIÓN DE INCIDENTES INFORMÁTICOS.

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas.

Autor(es): Adilén Sánchez Ramirez.
Yadira Nuñez Arteaga.

Tutor: Ing. Eduard Palomo Gené.

Ciudad de la Habana, julio de 2007
Año 49 de la Revolución.

“La clave del éxito depende sólo de lo que podamos hacer de la mejor manera posible.”

Henry Wadsonrth

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) para que haga el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los __ días del mes de julio del 2007.

_____	_____	_____
Firma del Autor	Firma del Autor	Firma del Tutor
Adilén Sánchez Ramirez	Yadira Nuñez Arteaga	Ing. Eduard Palomo Gené

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado “Sistema de Gestión de Reportes de Incidentes Informáticos.”, fue realizado en la Universidad de las Ciencias Informáticas (UCI) de la provincia de Ciudad Habana. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface

1. Totalmente.
2. Parcialmente en un ____ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes:

Y para que así conste, se firma la presente a los ___ días del mes de julio del 2007

_____ _____
Representante de la entidad Cargo

_____ _____
Firma Cuño

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: Sistema de Gestión de Reportes de Incidentes Informáticos.

Autores: Adilén Sánchez Ramirez y Yadira Nuñez Arteaga

El tutor del presente Trabajo de Diploma considera que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan.

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero en Ciencias Informáticas; y propongo que se le otorgue al Trabajo de Diploma la calificación de____.

_____ ____ de junio del 2007

Firma

Agradecimientos.

*A **Yohandri Ril,** por ayudarnos a romper el hielo con la investigación.*

*A **Roberto J. Martinez Mañalich,** y **Anabel Parra** por su contribución en el desarrollo de este trabajo.*

*A **Ronny, Yasser, Deivis y Giorbis** por toda su ayuda con la programación.*

*A **Eduard**, nuestro tutor, por tener siempre un tiempito a pesar de la carga de trabajo.*

*A todos los **profes** que pusieron su granito de arena a lo largo de la carrera para formarnos como profesionales.*

*A la **Revolución Cubana** por permitir que muchos jóvenes como nosotros puedan realizar sus sueños.*

*A **Fidel** por ser nuestro guía y por mostrarnos que se puede hacer mucho en tan poco tiempo.*

A todos aquellos que se detuvieron al menos un instante en su camino para regalarnos una sonrisa, una frase de apoyo o simplemente preguntarnos ¿Y la Tesis?

Muchas Gracias!!

Dedicatoria.

*A **mis padres**, por el amor, apoyo, comprensión y ejemplo que me han dado siempre, por brindarme seguridad y confianza en mis estudios y en cada una de mis acciones. Por anhelar tanto como yo que se realizara este sueño.*

*A mami **Nilda** y papi **Juan** por malcriarme, cuidarme y quererme tanto.*

*A **Osmani** por ser mi alma gemela, por su comprensión y Amor.*

*A mi **hermano**, y a mi **cuñi** por sus consejos, y por estar siempre pendientes de mí.*

*A mis **abuelitos** que donde quiera que estén sé que estarán orgullosos de su nieta.*

*A todos mis **amigos** por los maravillosos momentos que vivimos juntos.*

Yadira Nuñez Arteaga

*A mi **mamá** por todo su amor, comprensión y sacrificio.*

*A mi **papá** por haberme iniciado en el maravilloso mundo de la informática, por sus consejos, su guía y su amor.*

*A mi **hermanito** con el deseo de que siga mis pasos.*

*A mis **abuelos** que no pudieron estar hoy conmigo.*

*A mi **familia** en general por todo su cariño, su apoyo, por siempre confiar en mí.*

*A mis **amigos** de estos cinco años de bellos momentos que de seguro serán los mejores de toda mi vida.*

Adilén Sánchez Ramirez

Para quienes son y serán por siempre nuestra fuente de inspiración

...nuestros Padres.

Resumen.

Los sistemas actuales de las tecnologías de la información y las comunicaciones se caracterizan por su complejidad, interconexión, interdependencia y movilidad. Hoy en día estamos en presencia de un desarrollo de las tecnologías cada vez más vertiginoso y acelerado, directamente proporcional a él han crecido también las vulnerabilidades y los ataques sobre estos.

Actualmente son muchos los incidentes de seguridad reportados ya sean por parte de empresas o de usuarios domésticos en todo el mundo y muchos de estos recurren a sistemas automatizados de gestión donde registran sus quejas para que se les pueda dar solución.

En este trabajo de diploma se propone desarrollar un Sistema de Gestión de Reportes de Incidentes Informáticos en la Universidad de las Ciencias Informáticas (UCI), donde no existe ninguno de este tipo, para así facilitar el trabajo a los investigadores que son los encargados de resolver los incidentes informáticos. El mismo permite almacenar los reportes de incidentes realizados por los usuarios de la UCI, clasificarlos de acuerdo a su tipo e importancia, distribuirlos por investigador y darle un adecuado seguimiento y solución a los mismos.

Para el desarrollo de esta herramienta se utiliza RUP como metodología, UML como lenguaje representativo y Rational Rose como herramienta Case, además de PHP para la implementación y MySQL como gestor de base de datos.

Índice:

Introducción. 1
Capítulo 1. Fundamentación Teórica. 5
1.1. Introducción. 5
1.2. Seguridad Informática. 5
1.2.1. Necesidad de la Seguridad Informática. 5
1.3. Delitos informáticos. 6
1.3.1. Clasificación de delitos informáticos. 6
1.4. Definición de Incidente de Seguridad Informática. 7
1.4.1. Clasificación de Incidente de Seguridad Informática. 7
1.5. ¿Qué es un Equipo de Respuesta a Incidentes de Seguridad? 9
1.6. ¿Qué es el manejo de un incidente? 9
1.7. Estudio de sistemas existentes. 10
1.8. Tendencias y tecnologías actuales. 12
1.8.1. Aplicaciones Web. 12
1.8.2. Modelo Cliente - Servidor. 12
1.8.3. Servidor Web Apache. 13
1.8.4. Patrón Arquitectónico: Arquitectura en Capas. 14
1.8.5. Sistemas Web Modulares. 15
1.8.6. Lenguaje de programación. 16
1.8.7. Gestor de bases de datos. 19
1.9. Metodologías y herramientas a utilizar para el desarrollo del sistema. 21
1.9.1. Metodología RUP. 21
1.9.2. Lenguaje de modelado UML. 23
1.9.3. Herramientas Case. 24
1.9.4. Macromedia Dreamweaver 8.0 26
1.9.5. Zend Studio 26
1.10. Conclusiones. 27
Capítulo 2. Características del Sistema. 28
2.1 Introducción 28
2.2 Objeto de Estudio. 28
2.2.1 Situación Problémica. 28
2.2.2 Información que se maneja. 29
2.2.3 Propuesta de sistema. 29
2.2.4 Objeto de automatización. 29
2.3 Modelo del Negocio. 29
2.3.1 Proceso de negocio: Gestión de Incidentes. 29
2.3.2 Reglas del Negocio. 30
2.3.3 Actores del negocio 30
2.3.4 Trabajadores del negocio. 30
2.3.5 Diagrama de casos de uso del negocio. 31
2.3.6 Descripción de los Casos de Uso del Negocio. 31
2.3.7 Diagramas de Actividad del Negocio. 32
2.3.8 Diagrama de Clases del Modelo de objetos del Negocio. 32

2.4 Especificación de los requisitos de software. 32
2.4.1 Requerimientos Funcionales. 32
2.4.2 Requerimientos No Funcionales 33
2.5 Modelo del Sistema. 35
2.5.1 Actores. 35
2.5.2 Casos de Usos. 35
2.5.3 Diagrama de Casos de Uso del Sistema. 36
2.5.4 Descripción de Casos de uso. 39
2.6 Conclusiones: 52
Capítulo 3. Análisis y diseño del sistema. 53
3.1 Introducción. 53
3.2 Modelo del Análisis. 53
3.2.1 Diagramas de clases de análisis. 54
3.3 Modelo del Diseño. 58
3.3.1 Diagramas de clases y de interacción del diseño. 58
3.3.2 Diseño de la Base de Datos. 68
3.4 Definiciones de diseño. 71
3.4.1 Tratamiento de errores. 71
3.4.2 Seguridad. 71
3.4.3 Interfaz. 71
3.5 Conclusiones. 72
Capítulo 4. Implementación. 73
4.1 Introducción. 73
4.2 Modelo de Implementación. 73
4.2.1 Diagrama de Despliegue. 73
4.2.2 Diagramas de Componentes. 73
4.2.2.1 Diagrama de componentes Base de Datos. 74
4.2.2.2 Diagrama de componentes Código Fuente. 74
4.2.2.3 Diagrama de componentes Código Ejecutable. 75
4.3 Conclusiones. 76
Capítulo 5. Análisis de factibilidad. 77
5.1 Introducción. 77
5.2 Planificación basada en casos de uso. 77
5.3 Beneficios tangibles e intangibles. 82
5.4 Análisis de costos y beneficios. 83
5.5 Conclusiones. 83
Conclusiones. 84
Recomendaciones. 86
Referencias Bibliográficas. 87
Bibliografía. 88
Glosario de términos y siglas. 90
Anexos. 93
Anexo 1. Modelo del Negocio. 93
Anexo 2. Diagramas de Interacción del diseño. 95

Índice de Figuras.

Figure 1: Arquitectura en tres capas. 15
Figura 2: RUP en Dos Dimensiones. 22
Figura 3: El vocabulario UML. 24
Figura 4: Diagrama de casos de uso del negocio. 31
Figura 5: Diagrama de CU General por paquetes. 36
Figura 6: Diagrama de CU Paquete Seguridad. 37
Figure 7: Diagrama de CU Paquete Entrada de Datos. 38
Figura 8: Diagrama de CU Paquete Salida de Datos. 38
Figura 9: Diagrama de Clases del Análisis: CU Gestionar Incidente. 54
Figura 10: Diagrama de Clases del Análisis: CU Gestionar Solución Incidente. 54
Figura 11: Diagrama de Clases del Análisis: CU Reportar Incidente. 55
Figura 12: Diagrama de Clases del Análisis: CU Generar Reportes. 55
Figura 13: Diagrama de Clases del Análisis: CU Mostrar Solución Incidentes. 56
Figura 14: Diagrama de Clases del Análisis: CU Autenticar usuario. 56
Figura 15: Diagrama de Clases del Análisis: CU Asignar Incidente. 57
Figura 16: Diagrama de Clases del Análisis: CU Gestionar investigador. 57
Figura 17: Diagrama de Clases del Análisis: CU Insertar Noticias. 58
Figura 18: Diagrama de Clases del Diseño CU Gestionar Incidente. 59
Figura 19: Diagrama de Clases del Diseño CU Reportar Incidente. 60
Figura 20: Diagrama de Clases del Diseño CU Gestionar Solución Incidentes. 61
Figura 21: Diagrama de Clases del Diseño CU Generar Reportes. 62
Figura 22: Diagrama de Clases del Diseño CU Mostrar Solución Incidentes. 63
Figura 23: Diagrama de Clases del Diseño CU Asignar Incidentes. 64
Figura 24: Diagrama de Clases del Diseño CU Autenticar Usuario. 65
Figura 25: Diagrama de Clases del Diseño CU Gestionar Investigador. 66
Figura 26: Diagrama de Clases del Diseño CU Insertar Noticias. 67
Figura 27: Diagrama de clases persistentes. 68
Figura 28: Diagrama Entidad Relación. 69
Figura 29: Modelo de Datos. 71
Figure 30: Diagrama de Despliegue. 73
Figura 31: Diagrama de componentes Base de Datos. 74
Figura 32: Diagrama de componentes Código Fuente. 74
Figura 33: Diagrama de componentes Web o Código Ejecutable I. 75
Figura 34: Diagrama de componentes Web o Código Ejecutable II. 76
Figura 35: Diagrama de actividad CU Gestionar Incidente. 93
Figura 36: Diagrama de Clases del Modelo de objetos del Negocio. 94
Figura 37: Diagrama de Interacción del Diseño. Escenario Eliminar Incidente. 95
Figura 38: Diagrama de Interacción del Diseño. Escenario Modificar Incidente. 96
Figura 39: Diagrama de Interacción del Diseño. Escenario Mostrar Listado Incidentes. 97
Figura 40: Diagrama de Interacción del Diseño CU Reportar Incidente. 98
Figura 41: Diagrama de Interacción del Diseño CU Gestionar Solución Incidente. 99

Figura 42: Diagrama de Interacción del Diseño CU Generar Reportes. 100
Figura 43: Diagrama de Interacción del Diseño CU Mostrar Solución Incidentes. 101
Figura 44: Diagrama de Interacción del Diseño CU Asignar Incidentes. 102
Figura 45: Diagrama de Interacción del Diseño CU Autenticar Usuario. 103
Figura 46: Diagrama de Interacción del Diseño CU Gestionar Investigador. 104
Figura 47: Diagrama de Interacción del Diseño CU Insertar Noticias. 105

Índice de Tablas.

Tabla 1: Actores del negocio. 30
Tabla 2: Trabajadores del negocio. 31
Tabla 3: Descripción del CU del Negocio: Gestionar Incidente. 32
Tabla 4: Actores del sistema. 35
Tabla 5: Casos de Usos del Sistema. 36
Tabla 6: Descripción del CU: Autenticarse. 40
Tabla 7: Descripción del CU: Gestionar Investigador. 42
Tabla 8: Descripción del CU: Asignar Incidente. 43
Tabla 9: Descripción del CU: Insertar Noticia. 44
Tabla 10: Descripción del CU: Gestionar Incidente. 45
Tabla 11: Descripción del CU: Reportar Incidente 47
Tabla 12: Descripción del CU: Gestionar Solución de Incidentes. 49
Tabla 13: Descripción del CU: Generar Reportes 51
Tabla 14: Descripción del CU: Mostrar Solución de Incidentes. 51

Introducción.

El 2 de noviembre de 1988, un programa dañino empezó a introducirse en los ordenadores del sistema Arpanet, la red militar estadounidense, aprovechando varios errores en programas UNIX. Miles de máquinas colapsadas y el pánico general fueron el resultado de este gusano, al que llamaron "gusano de Morris" por su creador, Robert Morris Jr.

A pesar del desconcierto, el "gusano de Morris" tuvo una consecuencia positiva: abrir los ojos sobre la necesidad de montar algún tipo de respuesta organizada a futuros casos como ése. El ataque había cogido en falso a la naciente comunidad de Internet, formada mayoritariamente por una tropa de ciencia, que respondió de forma descoordinada y espontánea. Así nació el primer CERT (Computer Emergency Response Team), en la Universidad de Carnegie Mellon, llamado CERT/CC.

El CERT tiene como objetivos trabajar junto a la comunidad de Internet para facilitar su respuesta a problemas de seguridad informática que afecten a los sistemas centrales de Internet, dar pasos proactivos para elevar la conciencia colectiva sobre temas de seguridad informática, y llevar a cabo tareas de investigación que tengan como finalidad mejorar la seguridad de los sistemas existentes.

Los productos y servicios del CERT incluyen asistencia técnica 24 horas del día para responder a incidencias sobre seguridad informática, asistencia sobre vulnerabilidad de productos, documentos técnicos y cursos de formación.

En la actualidad existen cientos de CERTs en todas las partes del mundo todos con la misma misión: ser un punto central de ayuda en ataques, coordinar y formar para una mejor seguridad informática. [1]

En nuestra Universidad debido al desarrollo tecnológico asociado a sistemas computarizados, redes y aplicaciones, se hace difícil controlar las actividades de monitoreo y respuestas a posibles incidentes de seguridad debido a la ausencia de un sistema que permita a investigadores y administradores la gestión de los reportes sobre los incidentes informáticos realizados por los usuarios de la red.

Actualmente los usuarios reportan los incidentes de varias maneras ya sea por el sitio de Seguridad Informática, por la dirección de sus facultades etcétera, los investigadores revisan estos incidentes de manera esporádica, lo que trae consigo una demora a la respuesta del mismo.

No existe una forma rápida de clasificar los incidentes por tipo, localización, fecha, etcétera. Tampoco se tiene una manera de darle seguimiento a los incidentes ya que una vez resueltos no queda constancia de las medidas tomadas ni de otros datos que pudieran servir para futuras averiguaciones. No hay como

cuantificar el rendimiento de los investigadores. Todo esto trae consigo que los investigadores no posean una estandarización de los procedimientos a seguir en su trabajo.

Tomando en consideración lo anteriormente expuesto, se plantea el siguiente problema:

¿Cómo automatizar el proceso de gestión de reportes de incidentes en la UCI?, entonces llegamos a la Hipótesis de que con el desarrollo del software con la calidad requerida, se automatizarán los servicios de reporte de incidencias en la UCI haciendo que estos sean analizados de una manera más rápida y eficiente.

De acuerdo al problema planteado anteriormente se propone como objetivo general:

Desarrollar un Sistema que sea capaz de realizar la Gestión de los Reportes de Incidentes Informáticos en la UCI.

De este objetivo general se derivan los siguientes objetivos específicos:

- Obtener el diseño de una base de datos capaz de almacenar de manera organizada la información que se manipula.
- Desarrollar el análisis y diseño del sistema.
- Implementar el sistema con las características definidas en los procesos de análisis y diseño.

El Campo de Acción lo constituyen los Procesos Automatizados de Gestión de Reportes de Incidencias sobre delitos informáticos en la UCI.

Se espera con la realización de este trabajo obtener los siguientes Resultados:

♦ Se desarrolla en nuestra Universidad por primera vez un sistema automatizado que gestione los incidentes de seguridad reportados por los usuarios.

♦ Los reportes quedan distribuidos por investigador, almacenados, clasificados, y se les puede dar seguimiento por lo que se agiliza y se perfecciona el trabajo de los investigadores.

♦ Se pueden hacer informes y obtener información sobre los incidentes reportados donde los directivos de Seguridad en la UCI pueden llevar un mejor control respecto al tema.

Este trabajo de tesis consta de 5 capítulos los cuales se estructuran como se muestra a continuación:

Capítulo 1 Fundamentación Teórica.

Este primer capítulo tiene como objetivo exponer los fundamentos teóricos generales que sirven de punto de partida a la solución del problema. Se definen algunos conceptos que serán de utilidad en la

comprensión del mismo y de la propuesta de solución, también se hace alusión a algunos de los software que existen en el mundo relacionados con esta temática. Se describen además los procesos que serán objeto de automatización, así como la metodología a seguir y la tecnología a utilizar.

Capítulo 2 Características del Sistema.

Se realiza el estudio del problema, se plantean las reglas del negocio a considerar en la aplicación, así como su modelo de negocio donde se describen los actores y trabajadores que están involucrados con el negocio. También en este capítulo se describe la solución propuesta, exponiendo elementos imprescindibles para una solución exitosa: como lo son los requerimientos funcionales y no funcionales. De forma general se describen los actores del sistema y se realiza el modelos de casos de uso con sus correspondientes descripciones.

Capítulo 3 Análisis y Diseño.

En este capítulo se realiza el análisis y diseño del sistema. Se muestra la expansión de los casos de uso del sistema, los diagramas de clases y de interacción para cada uno de estos casos de usos, así como la descripción detallada de las clases. Se definen las clases persistentes para desarrollar el modelo de datos.

Capítulo 4 Implementación.

En este capítulo se realizan los diagramas de componentes y de implementación. Se implementa el sistema en términos de componentes organizando a estos de acuerdo a los nodos específicos en el modelo de despliegue.

Capítulo 5. Estudio de factibilidad.

En este capítulo se realiza un estudio de factibilidad económica realizado para este proyecto específico, en el que se determina si es factible o no el desarrollo del software propuesto, analizando los diferentes criterios que influyen en el cálculo del esfuerzo, tiempo de desarrollo y costo del proyecto.

Capítulo 1. Fundamentación Teórica.

1.1. Introducción.

El presente capítulo tiene como objetivo exponer los fundamentos teóricos generales que sirven de punto de partida a la solución del problema antes mencionado. En él se definen algunos conceptos que serán de utilidad en la comprensión del mismo y de la propuesta de solución, también se hace alusión a algunos de los software que existen en el mundo relacionados con esta temática. Se describen además los procesos que serán objeto de automatización, también la metodología a seguir y la tecnología a utilizar.

1.2. Seguridad Informática.

En términos generales la seguridad informática puede entenderse como aquellas reglas, técnicas y/o actividades destinadas a prevenir, proteger y resguardar los equipos informáticos individuales y conectados en una red frente a daños accidentales o intencionados, así como lo que es susceptible de robo, pérdida o daño ya sea de manera personal, grupal o empresarial. Entre los daños que se pueden producir se encuentran el mal funcionamiento del hardware, la pérdida física de datos y el acceso a zonas restringidas por parte de personas no autorizadas.

1.2.1. Necesidad de la Seguridad Informática.

En la medida que crece y se diversifica el uso de sistemas informáticos, se incrementan también los riesgos de que los equipos de cómputo y dispositivos electrónicos, conectados o no a Internet, sean vulnerables a ataques e incidentes que ponen en peligro la integridad de la información que en ellos se procesa, almacena o transfiere; de ahí la importancia fundamental de contar con programas preventivos, estrategias correctivas, planes de emergencia y respuestas inmediatas para proteger los equipos y sistemas, así como salvaguardar información y datos. En una sociedad que, cada día, basa más sus dinámicas y procesos en sistemas de cómputo y redes, la seguridad debe pasar de ser una responsabilidad importante a ser una prioridad para los gobiernos, las instituciones y empresas, así como para las personas, es decir, se requiere de una política clara e integral de seguridad en cómputo.

En el área de la informática existen varios riesgos tales como: ataque de virus, códigos maliciosos, gusanos, caballos de Troya y hackers; no obstante, con la adopción de Internet como instrumento de comunicación y colaboración, los riesgos han evolucionado y, ahora, las empresas deben enfrentar ataques de negación de servicio y amenazas combinadas; es decir, la integración de herramientas

automáticas de "hackeo", accesos no autorizados a los sistemas y capacidad de identificar y explotar las vulnerabilidades de los sistemas operativos o aplicaciones para dañar los recursos informáticos. Específicamente, en los ataques de negación de servicio, el equipo de cómputo ya no es un blanco, es el medio a través del cual es posible afectar todo el entorno de red; es decir, anular los servicios de la red, saturar el ancho de banda o alterar el Sitio Web de cualquier institución.

Es por la existencia de un número importante de amenazas y riesgos, que la infraestructura de red y recursos informáticos de una organización deben estar protegidos bajo un esquema de seguridad que reduzca los niveles de vulnerabilidad y permita una eficiente administración del riesgo.

1.3. Delitos informáticos.

Hoy en día nadie escapa de la enorme influencia que ha alcanzado la informática en la vida diaria de las personas y organizaciones, y la importancia que tiene su progreso para el desarrollo de un país. Las transacciones comerciales, la comunicación, los procesos industriales, las investigaciones, la seguridad, la sanidad, etc. son todos aspectos que dependen cada día más de un adecuado desarrollo de la tecnología informática. Junto al avance de la tecnología informática y su influencia en casi todas las áreas de la vida social, han surgido una serie de comportamientos ilícitos denominados, de manera genérica: delitos informáticos; el cual se define como "cualquier comportamiento antijurídico no ético o no autorizado, relacionado con el procesado automático de datos y/o trasmisiones de datos" y este implica actividades criminales que los países han tratado de encuadrar en figuras típicas de carácter tradicional, tales como robos, hurtos, fraudes, falsificaciones, perjuicios, estafas, sabotajes.

1.3.1. Clasificación de delitos informáticos.

La Organización Nacional de Naciones Unidas reconoce los siguientes tipos de delitos informáticos:

1. Fraudes cometidos mediante manipulación de computadoras:

♦ Manipulación de los datos de entrada: este tipo es el más común (fácil de cometer y difícil de descubrir)

♦ La manipulación de programas: Es muy difícil de descubrir y a menudo pasa inadvertido.

♦ Manipulación de los datos de salida: Ejemplo más común del que se hace objeto a los cajeros automáticos.

♦ Fraude efectuado por manipulación informática: aprovecha las repeticiones automáticas de los procesos de cómputo. Técnica especial que se denomina "Técnica del salchichón" en la que "rodajas muy

finas" apenas perceptibles de transacciones financieras, se van sacando repetidamente a otra. Se basa en el principio de que 10,66 es igual a 10,65 pasando 0,01 centavos a la cuenta del ladrón n veces.

2. Manipulación de los datos de entrada:

♦ Como objeto: cuando se alteran datos de los documentos almacenados en forma computarizada.

♦ Como instrumento: las computadoras pueden utilizarse también para efectuar falsificaciones de documentos de uso comercial.

3. Daños o modificaciones de programas o datos computarizados.

♦ Sabotaje informático.

♦ Acceso no autorizado a servicios y sistemas informáticos.

♦ Reproducción no autorizada de programas informáticos de protección legal. [2]

Adicionalmente a estos tipos de delitos reconocidos, el XV Congreso Internacional de Derecho ha propuesto todas las formas de conducta lesivas de la que puede ser objeto la información, ellas son:

- Fraude en el campo de la informática.
- Falsificación en materia informática.
- Sabotaje informático y daños a datos computarizados o programas informáticos.
- Acceso no autorizado.
- Reproducción no autorizada de un programa informático protegido.
- Espionaje informático.
- Uso no autorizado de una computadora.
- Tráfico de claves informáticas obtenidas por medio ilícito.
- Distribución de virus o programas delictivos. [3]

1.4. Definición de Incidente de Seguridad Informática.

Los Incidentes de Seguridad Informática son todos aquellos sucesos resultados de acciones malintencionadas y que comprometen leve o gravemente la confidencialidad, disponibilidad y/o integridad de la información. Son los eventos adversos, reales o sospechados en relación con la seguridad, de sistemas o redes de computación. Es el acto de violar una política de seguridad explícita o implicada. [4]

1.4.1. Clasificación de Incidente de Seguridad Informática.

♦ Acceso no autorizado.

- ✓Accesos no autorizados exitosos, sin perjuicios visibles a componentes tecnológicos.
- ✓Robo de información
- ✓Borrado de información
- ✓Alteración de la información
- ✓Intentos no recurrentes sin éxito de acceso no autorizado
- ✓Intentos recurrentes sin éxito de acceso no autorizado
- ✓Abuso y/o Mal uso de los servicios informáticos internos o externos que requieren autenticación.

♦ Código malicioso.

- ✓Virus
- ✓Troyanos
- ✓Worms
- ✓ Root Quits

♦ Denegación del Servicio.

- ✓Tráfico muy alto, ancho de banda totalmente ocupado, sin razones aparentes.
- ✓Servicio(s) interno(s) inaccesibles, sin razones aparentes.
- ✓Servicio(s) Externo(s) inaccesibles, sin razones aparentes.

♦ Escaneo, pruebas o intentos de Denegación del Servicio.

- ✓Tráfico muy alto, consumo de ancho de banda, sin razones aparentes.
- ✓Escáneres de Vulnerabilidades.
- ✓Huellas de herramientas para probar denegación de servicio.

♦ Escaneo, pruebas o intentos de obtención de información de la red o servidor.

- ✓Sniffers.
- ✓Escáneres de Red.
- ✓Escáneres de Vulnerabilidades.

♦ Mal uso de recursos corporativos.

- ✓Mal uso y/o Abuso de servicios corporativos internos o externos.
- ✓Utilización de recursos corporativos con fines personales.
- ✓Violación de las normas corporativas de acceso a Internet.
- ✓Mal uso y/o abuso del correo electrónico corporativo.

- Correo entrante.
- Correo saliente.
    - ✓Espionaje interno.
    - ✓Violación de las Políticas, Normas y Procedimientos de Seguridad Informática. [5]

1.5. ¿Qué es un Equipo de Respuesta a Incidentes de Seguridad?

Un Equipo de Respuesta a Incidentes de Seguridad (CSIRT) es una organización que es responsable de recibir, revisar y responder a informes y actividad sobre incidentes de seguridad. Sus servicios son generalmente prestados para un área de cobertura definida que podría ser una entidad relacionada u organización de la cual dependen, una corporación, una organización de gobierno o educativa; una región o país, una red de investigación; o un servicio pago para un cliente. [6]

1.6. ¿Qué es el manejo de un incidente?

El manejo de un incidente incluye tres funciones: información sobre el incidente, análisis del incidente, y respuesta al incidente. La función de información sobre el incidente le permite al equipo servir como el punto de contacto central para informar los problemas locales. Esto es reunir todos los informes y actividad sobre incidentes en un lugar donde la información puede ser revisada y correlacionada a través de la organización de la que depende o del área de cobertura. Esta información puede luego ser utilizada para determinar las tendencias y patrones de la actividad de los intrusos y recomendar las estrategias de prevención correspondientes para toda el área de cobertura. Ésta es una parte de la función del análisis del incidente. La otra parte del análisis del incidente es mirar con profundidad un informe de incidente o una actividad de incidente para determinar el alcance, la prioridad y la amenaza del mismo y también llevar a cabo la investigación de una posible respuesta y estrategia de mitigación.

Las funciones de respuesta a un incidente pueden tener varias formas. Se pueden enviar recomendaciones para recuperación, contención y prevención al área de cobertura o a los administradores de los sistemas y de la red de los sitios, quienes, por sí mismos, luego llevan a cabo los pasos indicados para la respuesta. Otra opción es realizar directamente estos pasos sobre los sistemas afectados. La respuesta también puede implicar compartir información y lecciones aprendidas con otros equipos de respuesta y otras organizaciones y sitios apropiados. [6]

1.7. Estudio de sistemas existentes.

Realizando un estudio sobre los equipos de respuesta a emergencias informáticas existentes, así como de las tecnologías más modernas que utilizan los sistemas automatizados de dichos equipos, se recopiló una gran cantidad de estos repartidos por todo el mundo, de los más importantes se hablará a continuación:

1. **IRIS-CERT:** Equipo de Respuesta a Emergencias Informáticas en España surgido en 1995. Es un servicio de seguridad que tiene como finalidad la detección de problemas que afecten a la seguridad de las redes de centros de RedIRIS, así como la actuación coordinada con dichos centros para poner solución a estos problemas. Esta RedIRIS brinda solo sus servicios a universidades y a otros centros de investigación, proveedores y usuarios de Internet en España, así como otros servicios de seguridad nacionales e internacionales, además actúa como punto de contacto y de coordinación de incidentes para otros servicios de seguridad. También se realiza una labor preventiva, avisando con tiempo de problemas potenciales, ofreciendo asesoramiento a los centros, organizando actividades de acuerdo con los mismos, y ofreciendo servicios complementarios. Los productos y servicios IRIS-CERT incluyen asistencia técnica 24 horas al día para responder a problemas o incidentes informáticos, asistencia sobre vulnerabilidad de productos, documentos técnicos y cursos de formación.

La herramienta de gestión de reportes de incidentes que utiliza IRIS-CERT ha evolucionado considerablemente trayendo consigo un mayor rendimiento y factibilidad a la hora de resolver un problema de seguridad. A continuación se muestran algunos puntos que demuestran dicha evolución:

Anteriormente

- Se utilizaba un mismo código para todo el intercambio de información relativo a un incidente.
- A nivel de operación, se procuraba que un incidente solamente involucrara a una organización.
- Varios tipos de correos sin incidente: Copyright, preguntas, notificaciones de ISP (menos 10%). Envío de los mensajes directamente desde agente de correo.
- Control manual de fechas.

Actualmente

- Se utilizan códigos distintos (incidentes, investigaciones y filtros).
- Posibilidad de controlar en un mismo incidente problemas en varios equipos de distintas organizaciones.

- Agrupación en algunas categorías de uso interno de los eventos que corresponden a asuntos de copyright, preguntas, etc.
- Envío de mensajes desde la herramienta (no personalización) y control automático de fechas.[7]

2. UNAM-CERT: Es un equipo de profesionales en seguridad en cómputo. Surge en el año 2000 como un organismo universitario y sin fines de lucro. Está localizado en el Departamento de Seguridad en Cómputo de la Dirección General de Servicios de Cómputo Académico (DGSCA), de la UNAM, en México. Atiende a instituciones de cualquier tipo, que han sido víctimas de algún ataque tanto en sus sistemas de cómputo como en sus sitios de Internet; publica periódicamente información actualizada sobre alertas y vulnerabilidades, implantación de políticas, elabora análisis de riesgos y realiza investigación dentro de esta área para contribuir a hacer, cada día, más seguros los sistemas y las redes.
El mismo se encarga de proveer el servicio de respuesta a incidentes de seguridad en cómputo a sitios que han sido víctimas de algún "ataque", así como de publicar información respecto a vulnerabilidades de seguridad, alertas de la misma índole y realizar investigaciones de la amplia área del cómputo y así ayudar a mejorar la seguridad de los sitios. Actualmente, UNAM-CERT atiende un promedio de 400 incidentes mayores al año en coordinación con los mejores equipos de respuesta a incidentes de todo en el mundo, y entre sus principales funciones también se encuentran informar, catalogar, clasificar y analizar oportunamente los problemas relacionados con la seguridad en cómputo; así como regular, controlar estándares y facilitar la difusión de normas para que los laboratorios, centros de investigación, instituciones bancarias y financieras, empresas y organizaciones las adopten en su beneficio; también trabaja en línea y ofrece un foro abierto en donde concurren diferentes instituciones, escuelas y universidades para obtener información, asesorías y herramientas especializadas. [8]

3. CU-CERT: Es un equipo de Respuestas a Incidentes Computacionales, integrado por un grupo de profesionales pertenecientes a la Oficina de Seguridad para las Redes Informáticas (OSRI), adscripta al Ministerio de Informática y las Comunicaciones (MIC), y su misión es la de prevenir y responder a los incidentes computacionales en Cuba.
CuCERT tiene en su membresía a 4 profesionales Certificados en Creación y Administración de Equipos de Respuesta ante Incidentes de Seguridad Computacional, CSIRT. Certificado que emite la empresa NeoSecure de Chile al impartir el curso oficial del CERT®/CC.

Los principales objetivos de este equipo son:

- Responder a los incidentes computacionales que se presenten en el país.
- Informar sobre vulnerabilidades y amenazas de seguridad informática.
- Detección temprana de incidentes computacionales.
- Gestionar las reclamaciones internacionales.
- Educar a la comunidad en general, sobre temas de seguridad informática. [9]

1.8. Tendencias y tecnologías actuales.

1.8.1. Aplicaciones Web.

Una aplicación Web es un sistema informático que los usuarios utilizan accediendo a un servidor Web a través de Internet.

La creciente popularidad de las aplicaciones Web se debe a sus múltiples ventajas, entre las cuales se pueden citar:

1. Multiplataforma: Con un solo programa, un único ejecutable, las aplicaciones pueden ser utilizadas a través de múltiples plataformas, tanto de hardware como de software.
2. Actualización instantánea: Debido que todos los usuarios de la aplicación hacen uso de un sólo programa que radica en el servidor, los usuarios siempre utilizarán la versión más actualizada del sistema.
3. Suave curva de aprendizaje: Los usuarios, como utilizan la aplicación a través de un navegador, hacen uso del sistema tal como si estuvieran navegando por Internet, por lo cual su acceso es más intuitivo.
4. Fácil de integrar con otros sistemas: Debido a que se basa en protocolos estándares, la información manejada por el sistema puede ser accedida con mayor facilidad por otros sistemas.
5. Acceso móvil: El usuario puede acceder a la aplicación con la única restricción de que cuente con un acceso a la red privada de la organización o a Internet, dependiendo de las políticas de dicha organización; puede hacerlo desde una computadora de escritorio, una laptop o desde una agenda electrónica; desde su oficina, hogar u otra parte del mundo.

1.8.2. Modelo Cliente - Servidor.

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar

información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes.

En esta arquitectura la computadora de cada uno de los usuarios, llamada cliente, produce una demanda de información a cualquiera de las computadoras que proporcionan información, conocidas como servidores estos últimos responden a la demanda del cliente que la produjo.

Los clientes y los servidores pueden estar conectados a una red local o una red amplia, como la que se puede implementar en una empresa o a una red mundial como lo es la Internet.

Bajo este modelo cada usuario tiene la libertad de obtener la información que requiera en un momento dado proveniente de una o varias fuentes locales o distantes y de procesarla como según le convenga. Los distintos servidores también pueden intercambiar información dentro de esta arquitectura. [10]

Ventajas de la arquitectura cliente-servidor:

- Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema.
- Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado.

1.8.3. Servidor Web Apache.

Un servidor es una computadora que entrega a otras computadoras (los clientes), una información que ellos requieren bajo un lenguaje común, denominado protocolo. Por lo tanto al ver una página Web es porque el servidor les entrega una página HTML vía protocolo HTTP o protocolo para la transmisión de hipertexto, a través de una conexión TCP/IP por el puerto 80.

Por lo tanto en el Servidor Web es donde se almacena la información estática accedida y/o las aplicaciones que la generan.

Apache está diseñado para ser un servidor Web potente y flexible que pueda funcionar en la más amplia variedad de plataformas y entornos. Apache se ha adaptado siempre a una gran variedad de entornos a través de su diseño modular. Este diseño permite a los administradores de sitios Web elegir que características van a ser incluidas en el servidor seleccionando que módulos se van a cargar, ya sea al compilar o al ejecutar el servidor.

Hoy en día es el servidor Web más utilizado del mundo, encontrándose muy por encima de sus competidores, tanto gratuitos como comerciales. Es un software de libre distribución que publica su código fuente, lo que permite que cualquiera pueda modificarlo y colaborar así a su desarrollo.

Apache ha sido desde abril de 1996 el servidor Web más popular. En noviembre de 2005 la Netcraft Web Server Survey encontró que más del 70% de los sitios Web en Internet usan a Apache como servidor Web, haciéndolo más ampliamente usado que los otros servidores Web de forma combinada. [*]

1.8.4. Patrón Arquitectónico: Arquitectura en Capas.

Un patrón de arquitectura de software describe un problema particular y recurrente del diseño, que surge en un contexto específico, y presenta un esquema genérico y probado de su solución.

Este patrón define cómo organizar el modelo de diseño en capas, que pueden estar físicamente distribuidas, lo cual quiere decir que los componentes de una capa sólo pueden hacer referencia a componentes en capas inmediatamente inferiores. Este patrón es importante porque simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores. Además, nos ayuda a identificar qué puede reutilizarse, y proporciona una estructura que nos ayuda a tomar decisiones sobre qué partes comprar y qué partes construir.

Principales estilos de arquitecturas estratificadas de las aplicaciones distribuidas contemporáneas:

- Arquitecturas de dos niveles
- Arquitecturas de tres niveles
- Arquitecturas de n niveles

Arquitectura de tres niveles.

Para enfrentarse a estos temas, la comunidad de software desarrolló la noción de una arquitectura de tres niveles. La aplicación se divide en tres capas lógicas distintas, cada una de ellas con un grupo de interfaces perfectamente definido.

Estas tres capas son:

- *La capa de la Presentación.*

Esta capa reúne todos los aspectos del software que tiene que ver con las interfaces y la interacción con los diferentes tipos de usuarios humanos. Estos aspectos típicamente incluyen el manejo y aspecto de las ventanas, el formato de los reportes, menús, gráficos y elementos multimedia en general.

- *La capa de Negocio (Capa Lógica).*

Esta capa reúne todos los aspectos del software que automatizan o apoyan los procesos de negocio que llevan a cabo los usuarios. Estos aspectos típicamente incluyen las tareas que forman parte de los procesos, las reglas y restricciones que aplican.

- *La capa de Datos.*

Esta capa reúne todos los aspectos del software que tienen que ver con el manejo de los datos persistentes, por lo que también se le denomina la capa de las Bases de Datos.

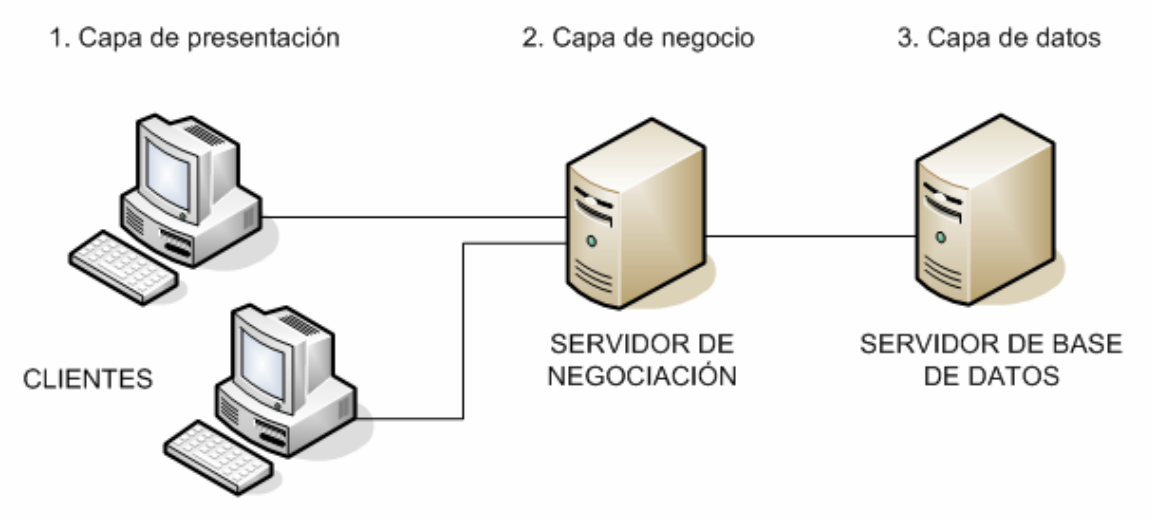

Figure 1: Arquitectura en tres capas.

1.8.5. Sistemas Web Modulares.

Los Sistemas Web Modulares por definición, son Sitios Web completos que poseen todas las herramientas necesarias para administrar variados tipos de información de manera eficiente y profesional, con diversas áreas de exposición. Es importante destacar el grado de interactividad o comunicación que se logra con los usuarios o clientes debido a las formas y estructuras que poseen los Sistemas Web Modulares.

No importa el volumen de información que contenga el Sistemas Web Modular, la estructura, diseño y plataforma sobre el cual se construye, permite mantener una respuesta óptima a gran velocidad, aún cuando muchos usuarios solicitan la información al mismo tiempo.

En este tipo de Sitios Web la seguridad juega un papel importante, la lectura de la información se realiza de acuerdo a lo que el usuario desea ver, sin arriesgar otra información saturando la pagina actual.

Los Sistemas Web Modulares se crean para crecer e ir incorporando módulos de acuerdo a las necesidades de los creadores. Esto constituye una de sus principales facilidades. La reutilización del código evitando su innecesaria repetición constituye otra de sus mayores ventajas.

1.8.6. Lenguaje de programación.

En la actualidad existen varios lenguajes de programación que son usados para el desarrollo de páginas Web. A continuación se muestran algunos de estos lenguajes donde se realiza un estudio comparando fortalezas y debilidades de los mismos.

ASP.

ASP significa Active Server Pages y es un componente de Microsoft Internet Information Server. Este lenguaje reemplaza la forma tradicional de intercambiar información entre usuarios, además permite que el código sea incrustado en un documento HTML y que corra en el servidor. ASP es seguro y fácil de programar, se ejecuta sobre el servidor y envía datos de regreso al cliente (a través del buscador).

Existen varios lenguajes que se pueden usar para hacer ASP. El más comúnmente utilizado es VBScript y es por eso mucho más fácil, ya que es nativo de Microsoft.

La contrapartida es que ASP dispone de funcionalidades muy limitadas en modo nativo, por lo que necesita de objetos externos para las funciones más básicas (acceso a datos, correo electrónico, acceso a archivos, etc.).

Las funciones complejas tales como imágenes dinámicas, IMAP, SNMP, flash dinámico, pdf, acceso nativo (no ODBC) a bases de datos tales como Oracle, Ovrimos, Postgres, Sybase, mysql, MSSQL, bases de datos de Ingres, Interbase y de Informix, LDAP, y zócalos, están disponibles para cualquier instalación de PHP, pero no están disponibles con el ASP.

PERL.

Perl es un lenguaje interpretado optimizado para la lectura y extracción de información de archivos de texto, generando reportes basados en la información proporcionada por ellos. Es también un lenguaje bastante utilizado para muchos sistemas manipuladores de tareas como lenguaje de contenido dinámico.

Perl (Practical Extraction and Report Language) es lo que se conoce como un lenguaje “script", es decir, uno en el que no hace falta compilar el programa escrito o “script". En la práctica se compila parcialmente al comienzo de su ejecución.

Entre las ventajas que tiene este lenguaje es que se pueden juntar varios programas de una forma sencilla para alcanzar una meta determinada. Es relativamente rápido para un lenguaje tipo “script". Está disponible en múltiples plataformas y sistemas operativos y funciona bajo diferentes sabores de UNIX, Linux y todo tipo de Windows. Un programa que se escriba teniendo en cuenta la compatibilidad puede ser escrito en una plataforma y ejecutado en otra. El desarrollo de aplicaciones es muy rápido. Hay una colección enorme de módulos que pueden ser incorporados a cualquier “script" de Perl. En particular existe una extensión para cálculo numérico denominada PDL. Perl es gratuito. Brinda facilidades a los programadores para desarrollar los programas como deseen.

Perl tiene como contrapartida que es lento para algunas aplicaciones, como programación a bajo nivel, escribiendo un “driver" para una aplicación o corriendo modelos numéricos de cálculo intensivo. La libertad que se le otorga al programador puede significar que el resultado sea un programa ilegible. Si no se escribe con cuidado puede llegar a ser difícil de leer. No se pueden compilar programas Perl. Aunque actualmente se está desarrollando un compilador que realice esta tarea y hay uno comercial disponible para la plataforma Windows. Los programas Perl no correrán mucho más rápidos cuando se compilen, la única ventaja es que está en la desaparición de la fase inicial de compilación al correr la aplicación. Utiliza muchos recursos de la máquina. Esto significa que no es tan ligero como un programa en C, pero en la práctica es ligero comparado con la potencia de computación de los ordenadores actuales. Este no fue diseñado con la finalidad de crear aplicaciones Web.

PHP es más seguro que PERL ya que los scripts de este último tienden a tener más agujeros de seguridad. Además PHP es más fácil de aprender en comparación con Perl, el estilo de programación de Perl es único y no es aplicable universalmente o desde otros lenguajes de programación.

¿Por qué se usó PHP?

PHP es un lenguaje de programación usado generalmente para la creación de contenido para sitios Web habitualmente en combinación con el motor de base datos MySQL, aunque cuenta con soporte nativo

para otros motores, incluyendo el estándar ODBC, lo que amplía en gran medida sus posibilidades de conexión.

PHP es un acrónimo recurrente que significa "Hypertext Pre-processor", y se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web. Se utiliza también para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica usando la biblioteca GTK+. Este se utiliza también para Programación en consola, al estilo de Perl o Shell scripting y para la creación de aplicaciones gráficas independientes del navegador, por medio de la combinación de PHP y GTK (GIMP Tool Kit), lo que permite desarrollar aplicaciones de escritorio en los sistemas operativos en los que está soportado.

PHP es la gran opción de desarrollo de aplicaciones Web porque el código de PHP se ejecuta sin cambios en una gran variedad de sistemas, cosas que no pueden decir otras herramientas como ASP de Microsoft y en distintos servidores. Este lenguaje tiene muchas virtudes que lo han convertido en la opción de muchos programadores debido a que:

- Brinda facilidad de aprendizaje del lenguaje.
- Gran cantidad de funciones desarrolladas (PHP incorpora mas de 1000 funciones).
- Tiene una amplia disponibilidad de secuencia de comandos regeneradas en PHP.
- Tiene la capacidad de incrustar código PHP en las páginas HTML, además se vincula fácilmente con las principales bases de datos.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Es multiplataforma y Multisistema Operativo.
- Posee una amplia documentación, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Miles de ejemplos y código fuente disponible.
- Perfecta integración del Apache-PHP-MySQL.
- No depende de un único proveedor de servicios.

- El código fuente es abierto y gratuito.
- Existen gran cantidad de scripts en PHP ya programados y totalmente gratuitos, que permiten fácilmente añadir todo tipo de funcionalidad a nuestra Web (foros, Chat, encuestas, tiendas de comercio electrónico, etc.).
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

1.8.7. Gestor de bases de datos.

Un Sistema Gestor de base de datos (SGBD) es un conjunto de programas que permiten crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad. Actualmente el gestor de base de datos juega un rol central en la informática, como única utilidad, o como parte de otra aplicación.

Microsoft SQL Server

Es un sistema de gestión de bases de datos relacionales (SGBD) basada en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea Así de tener unas ventajas que a continuación se pueden describir.

Entre sus características figuran:

- Soporte de transacciones.
- Gran estabilidad.
- Gran seguridad.
- Escalabilidad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos

Desventajas.

No es multiplataforma, solo puede ser utilizado con sistema operativo Windows que está patrocinado por la compañía Microsoft.

PostgreSQL

Postgres intenta ser un sistema de bases de datos de mayor nivel que MySQL, a la altura de Oracle, Sybase o Interbase. Entre sus ventajas tenemos que por su arquitectura de diseño, escala muy bien al aumentar el número de CPUs y la cantidad de RAM. Soporta transacciones y desde la versión 7.0, claves ajenas (con comprobaciones de integridad referencial). Tiene mejor soporte para triggers y procedimientos en el servidor. Soporta un subconjunto de SQL92 MAYOR que el que soporta MySQL. Además, tiene ciertas características orientadas a objetos.

Entre sus inconvenientes, está principalmente que consume bastantes recursos y carga más el sistema. Ocupa mucho espacio: límite del tamaño de cada fila de las tablas a 8k (se puede ampliar a 32k recompilando, pero con un coste añadido en el rendimiento). Todo esto trae consigo que sea de 2 a 3 veces más lenta que MySQL. También se tiene que posee menos funciones en PHP.

MySQL

MySQL es un gestor de bases de datos SQL (Structured Query Language). Es una implementación Cliente-Servidor que consta de un servidor y diferentes clientes (programas/librerías). Se pueden agregar, acceder, y procesar datos grabados en una base de datos.

Es un Sistema de Gestión de Base de Datos Relacional. El modelo relacional se caracteriza a muy grandes rasgos por disponer que toda la información debe estar contenida en tablas, y las relaciones entre datos deben ser representadas explícitamente en esos mismos datos. Esto añade velocidad y flexibilidad.

MySQL es un software de código abierto esto quiere decir que es accesible para cualquiera, para usarlo o modificarlo. Podemos descargar MySQL desde Internet y usarlo sin pagar nada, de esta manera cualquiera puede inclinarse a estudiar el código fuente y cambiarlo para adecuarlo a sus necesidades. MySQL usa el GPL (GNU Licencia Publica General) para definir que se puede y no se puede hacer con el software en diferentes situaciones.

¿Por qué se usó MySQL?

Trabajar con MySQL es factible ya que es muy rápido, tiene buenas utilidades de administración, es confiable, robusto y fácil de usar tanto para volúmenes de datos grandes como pequeños, sin límites en los tamaños de registros. Además tiene un conjunto muy práctico de características desarrolladas en cooperación muy cercana con los usuarios. MySQL hoy en día ofrece un rico y muy útil conjunto de

funciones. La conectividad, velocidad y seguridad ya que tiene mejor control de acceso a usuarios, hace de MySQL altamente conveniente para acceder a bases de datos en Internet.
Además el lenguaje que se utiliza para el desarrollo de este trabajo que es PHP tiene una integración perfecta con MySQL ambas resultan muy útiles para diseñar de forma rápida y eficaz aplicaciones Web dirigidas a bases de datos.

1.9. Metodologías y herramientas a utilizar para el desarrollo del sistema.

Para el desarrollo del sistema se realiza un estudio de las posibles metodologías y herramientas a utilizar en su construcción, teniéndose en cuenta las tendencias y tecnologías actuales y las novedades de cada una de ellas.

1.9.1. Metodología RUP.

Es necesario al enfrentarse a la construcción de un software seguir una serie de pasos que lleven a la realización de un producto robusto y que cumpla con todos los requerimientos que a este se le imponen. Para lograr esto se hace necesario el uso de una metodología.
La metodología que se emplea para el desarrollo de este trabajo de diploma es RUP (Rational Unified Process, Proceso Unificado de Desarrollo de Software).
El Proceso Unificado es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Sin embargo el Proceso unificado es mas que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto. [11]
Esta metodología hace énfasis en la adopción de las mejores prácticas del desarrollo de software, como una manera de reducir los riesgos inherentes en el desarrollo de una nueva aplicación de software, de esta manera se logran resultados más predecibles, con procesos comunes que mejoran la comunicación y crean un entendimiento de todas las tareas y responsabilidades. Podemos decir además que es un proceso muy organizativo, orientado a objetos, el cual se basa en roles.
En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales divididos en 4 fases. Los 6 primeros flujos son conocidos como flujos de ingeniería y los tres últimos como de apoyo. En la Figura1: RUP en Dos Dimensiones se representa el proceso en el que se grafican los flujos de trabajo y las fases y muestra la dinámica expresada en iteraciones y puntos de control.

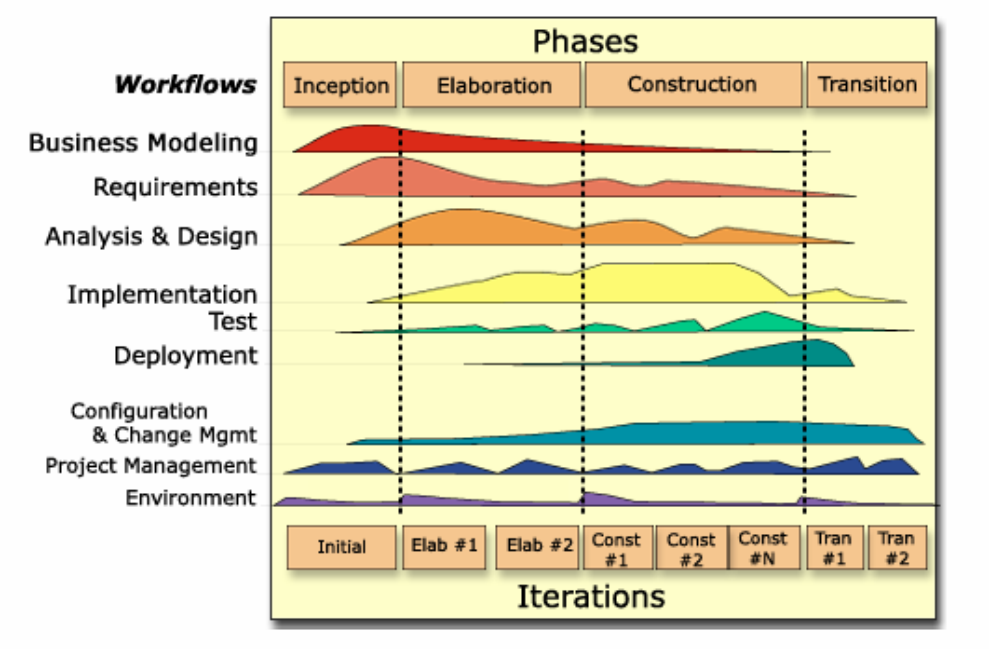

Figura 2: RUP en Dos Dimensiones.

Entre sus principales características de RUP tenemos las siguientes:

- ♦ Iterativo e Incremental.
- ♦ Dirigido por los Casos de Uso.
- ♦ Centrado en la Arquitectura. [11]

Iterativo e Incremental:

Pequeños proyectos que incorporan incrementalmente nueva funcionalidad y cuyo desarrollo es una iteración.

- Obtiene un Sistema Robusto
- Reduce el Riesgo de tener un mal producto
- Reduce el Riesgo de no obtener el producto en el tiempo previsto
- Permite atacar problemas con requisitos incompletos.

Dirigido por los Casos de Uso:

- Servicios que un actor requiere del sistema y le proporcionan un resultado.
- Proporcionan los Requisitos Funcionales del Sistema.
- Describen toda la funcionalidad del Sistema.
- Cambios en Requisitos de un Caso de Uso fácil detectar las clases y componentes que afectan.

Centrado en la Arquitectura:

- Casos de Uso describen la *Funcionalidad* del Sistema
- Arquitectura define la *Forma* del Sistema
- Se describe mediante *Vistas* que incorporan el 5 -10% de los casos de uso más relevantes.

1.9.2. Lenguaje de modelado UML.

El Lenguaje unificado de Modelado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar los artefactos de un sistema de software [12]. Está formado por diagramas que contienen elementos y sus relaciones. Figura 2.

UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar.

Además tiene las siguientes características:

- Permite modelar sistemas utilizando técnicas orientadas a objetos (OO).
- Permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos.
- Puede conectarse con lenguajes de programación (Ingeniería directa e inversa).
- Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.).
- Cubre las cuestiones relacionadas con el tamaño propio de los sistemas complejos y críticos.
- Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.
- UML es independiente del proceso, aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.

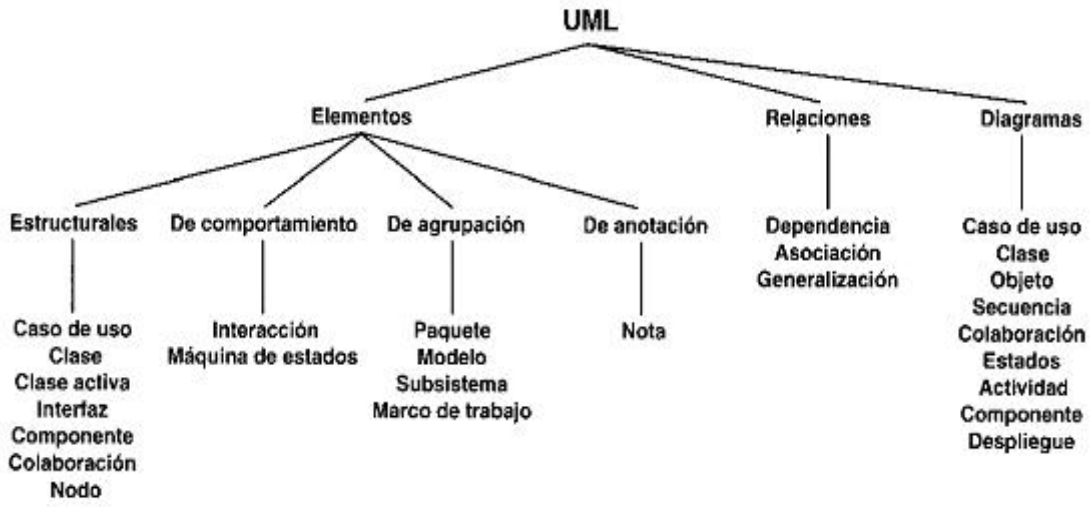

Figura 3: El vocabulario UML.

1.9.3. Herramientas Case.

1.9.3.1. Rational Rose.

Es la herramienta Case desarrollada por los creadores de UML que cubren todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes y certificación de las distintas fases. Permite una trazabilidad real entre modelo (análisis y diseño) y el código ejecutable.

Rational Rose domina el mercado de herramientas para el análisis, modelamiento, diseño y construcción orientada a objetos, tiene todas las características que los desarrolladores, analistas, y arquitectos exigen soporte UML incomparable, desarrollo basado en componentes con soporte para arquitecturas líderes en la industria y modelos de componentes, facilidad de uso e integración optimizada.

La corporación Rational Rose ofrece el Proceso Unificado de Racional (RUP), que unifica las mejores prácticas de muchas disciplinas en un consistente y completo proceso del ciclo de vida, que permite al equipo de desarrollo disminuir los tiempos de liberación, además de hacer más predecible el software que ellos producen. Este proceso está basado en el Lenguaje Unificado de Modelación (UML – estándar de la industria) y únicamente integrado a herramientas líderes en el desarrollo de software de Racional, el Proceso Unificado de Racional apoya el equipo completo de desarrollo de software con guías detalladas e información crítica aplicable a la mayoría de las aplicaciones de la industria.

1.9.3.2. Embarcadero ERStudio 6.6

Embarcadero ERStudio es una de las herramientas CASE de diseño de bases de datos que ayuda a generar, mantener alta calidad y gran rendimiento en las aplicaciones de la base de datos desde un modelo lógico de los requerimientos de información y las reglas de negocio que definen la base de datos al modelo físico optimizado por las características especificas de esta. Permite visualizar la estructura, elementos clave y optimizar el diseño de las bases de datos, genera tablas u otras especificaciones en dependencia de la plataforma seleccionada.

Características:

- Arquitectura de diseño en capas: Erwin proporciona la flexibilidad necesaria para generar el modelo de datos que mejor satisface las necesidades de las entidades. Ofrece soporte para modelos lógicos y físicos independientes, así como para el modelo lógico/físico combinado tradicional.
- Tecnología de transformación: El diseño físico de una base de datos coincide raras veces con el diseño lógico inicial de los datos. Las limitaciones de las entidades imponen la necesidad de modificar tablas para cumplir los requisitos de rendimiento de las aplicaciones actuales. La tecnología de transformación permite implementar este tipo de cambios y a la vez mantener la integridad del diseño original.
- Definición de estándares: La herramienta ofrece soporte para la definición y el mantenimiento de estándares mediante el Diccionario de Dominios y el Editor de Estándares de tipos de datos, permite que el usuario defina los estándares para la asignación de tipos definidos por el usuario y predeterminadas a tipos de datos específicos de cada sistema gestor de base de datos.
- Gestión de modelos de gran tamaño: Proporciona una visión específica para usuarios individuales y dividen los modelos en subconjuntos más pequeños y manipulables.

Tiene como ventajas:

- ♦ Facilidades de diseño de diagramas Entidad-Relación y Entidad-Relación extendido y transformación de este al modelo relacional (en tercera forma normal, preservando las dependencias funcionales y sin pérdidas de información).
- ♦ Comparación comprensiva entre el modelo de datos y la base de datos.
- ♦ Soporta la separación del modelo lógico y del físico.

1.9.4. Macromedia Dreamweaver 8.0

Macromedia Dreamweaver es un editor de páginas web, creado por Macromedia. Es el programa de este tipo más utilizado en el sector del diseño y la programación web, por sus funcionalidades y su integración con otras herramientas.

Dreamweaver permite al usuario utilizar la mayoría de los navegadores Web instalados en su ordenador para previsualizar las páginas web. También dispone de herramientas de administración de sitios dirigidas a principiantes como, por ejemplo, la habilidad de encontrar y reemplazar líneas de texto y código por cualquier tipo de parámetro especificado, hasta el sitio web completo. El panel de comportamientos también permite crear JavaScript básico sin conocimientos de código.

Además de sus capacidades, tiene las funciones típicas de un editor de código fuente para la web:

- Un administrador de sitios, para agrupar los archivos según el proyecto al que pertenezcan.
- Un cliente FTP integrado, que permite subir los archivos editados inmediatamente al sitio en Internet.
- Función de autocompletar y resaltado de la sintaxis para instrucciones en HTML y lenguajes de programación como PHP, JSP o ASP.
- Soporta gran cantidad de tecnologías, hojas de estilo y capas, Javascript para crear efectos e interactividades, Inserción de archivos multimedia.

1.9.5. Zend Studio

Zend Studio es uno de los ambientes de desarrollo integrado o Integrated Development Environment (IDE) disponible para desarrolladores profesionales que agrupa todos los componentes de desarrollo necesarios para ciclo de desarrollo de aplicaciones PHP. Se trata de un programa de la casa Zend, impulsores de la tecnología de servidor PHP, orientada a desarrollar aplicaciones web, en lenguaje PHP. El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código. A través de un comprensivo conjunto de herramientas de edición, depurado, análisis, optimización y bases de datos, Zend Studio acelera los ciclos de desarrollo y simplifica los proyectos complejos.

1.10. Conclusiones.

En este capítulo se muestran las condiciones y problemas que rodean el objeto de estudio; a través de los conceptos y definiciones planteadas. Para desarrollar el sistema se hace uso de la tecnología para la programación de páginas dinámicas el lenguaje PHP y con soporte de base de datos en MySQL. El proceso de desarrollo es RUP, el cual está basado en la orientación a objetos y el modelamiento visual usando UML, lo cual permite incorporar al proceso de desarrollo de software un mejor control de los requerimientos y cambios.

Capítulo 2. Características del Sistema.

2.1 Introducción

En el presente capítulo se hace la descripción de la propuesta de sistema, se describe el objeto de estudio y los procesos del negocio para un mayor entendimiento del mismo. Se especifican los detalles de la construcción de la herramienta, se realiza la propuesta del sistema, se analizan los requerimientos funcionales y no funcionales, y los casos de uso del sistema.

2.2 Objeto de Estudio.

Debido a la reciente creación de la Universidad de las Ciencias Informáticas no existe un sistema automatizado que gestione el proceso de gestión de reportes de incidentes de seguridad informática de manera eficiente.

2.2.1 Situación Problémica.

Actualmente en la Universidad de las Ciencias Informáticas debido al desarrollo tecnológico asociado a sistemas computarizados, redes y aplicaciones, se hace difícil controlar las actividades de monitoreo y respuestas a posibles incidentes de seguridad debido a la ausencia de un sistema que permita a investigadores y administradores la gestión de los reportes sobre los delitos informáticos realizados por los usuarios de la red.

Actualmente los usuarios reportan los incidentes de varias maneras ya sea por el sitio de Seguridad Informática, por la dirección de sus facultades etcétera, los investigadores revisan estos incidentes de manera esporádica, lo que trae consigo una demora a la respuesta del incidente.

No existe una forma rápida de clasificar los incidentes por importancia, localización, fecha, etcétera. Tampoco se tiene una manera de darle seguimiento a los incidentes ya que una vez resueltos no queda constancia de las medidas tomadas ni de otros datos que pudieran servir para futuras averiguaciones. No hay como cuantificar el rendimiento de los investigadores. Todo esto trae consigo que los investigadores no posean una estandarización de los procedimientos a seguir en su trabajo.

2.2.2 Información que se maneja.

En la aplicación se maneja toda la información referente a los Incidentes de Seguridad Informática.

Un Incidente de Seguridad Informática se clasifica como todos aquellos sucesos resultados de acciones malintencionadas y que comprometen leve o gravemente la confidencialidad, disponibilidad y/o integridad

de la información. Son los eventos adversos, reales o sospechados en relación con la seguridad, de sistemas o redes de computación. Es el acto de violar una política de seguridad explícita o implicada.

2.2.3 Propuesta de sistema.

Con el objetivo de mejorar la calidad del trabajo y darle solución a los problemas antes mencionados se ha decidido desarrollar un sistema de gestión que se centra dentro de este proceso tan importante para la universidad.

2.2.4 Objeto de automatización.

Los procesos que son objeto de automatización de dicho sistema son los siguientes:

Reportar Incidentes.

Gestionar Incidente.

2.3 Modelo del Negocio.

2.3.1 Proceso de negocio: Gestión de Incidentes.

La gestión de incidentes tiene como objetivo controlar y resolver todos los incidentes de seguridad que han sido reportados por los usuarios en la Universidad de las Ciencias Informáticas.

El negocio presenta un actor representado por el Reportador el cual es el encargado reportar los incidentes de seguridad.

El investigador es el encargado de resolver el incidente de seguridad y para ello lo primero que realiza es un análisis del mismo para identificar el tipo de incidente y asignarle la prioridad requerida.

La prioridad que se le otorga al incidente es de Normal, Alta, o Baja en dependencia de la decisión del investigador. Si al incidente se le da prioridad entonces se pasa a resolver el mismo, en caso de que no existan todas las pruebas suficientes para darle solución se intercambia información con el reportador del incidente para lograr así una mayor precisión de datos.

En caso de que se le dé solución satisfactoriamente al incidente se procede a archivarlo y el investigador toma las medidas pertinentes luego de resolver el incidente de seguridad.

En caso de que no existan pruebas suficientes para darle solución al incidente el investigador le informa al reportador de que no es posible resolver el incidente por falta de pruebas y el mismo pasa a ser descartado.

En caso de que el incidente no tenga la prioridad requerida se archiva y se espera a que le dé la prioridad.

2.3.2 Reglas del Negocio.

Las reglas de negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio.

- El investigador es la única persona encargada de manejar la información de los incidentes y de darle solución a los mismos.
- Cuando no existen pruebas suficientes para resolver el incidente este pasa a ser descartado.

2.3.3 Actores del negocio

Un actor del negocio es cualquier individuo, grupo, organización, máquina o sistema de información externo que interactúa con el negocio.

Actores del negocio	Justificación
Reportador	El reportador es el que inicia las acciones que dan lugar a la Gestión de Reportes de Incidentes de Seguridad Informática, y al mismo tiempo es el principal beneficiado con el resultado de dicho proceso.

Tabla 1: Actores del negocio.

2.3.4 Trabajadores del negocio.

Un trabajador representa a personas, o sistemas (software) dentro del negocio que son las que realizan las actividades que están comprendidas dentro de un caso de uso.

Trabajadores del negocio	Justificación
Investigador	Es el encargado de realizar la Gestión de los Incidentes de Seguridad Informática y de brindarle al Personal Autorizado las informaciones que estos soliciten. No se beneficia en ningún momento de las acciones realizadas en los procesos de negocio, sino que se limita a ejecutar dichas acciones.

Tabla 2: Trabajadores del negocio.

2.3.5 Diagrama de casos de uso del negocio.

El diagrama de casos de uso del negocio representa gráficamente los procesos del negocio y su interacción con los actores del negocio.

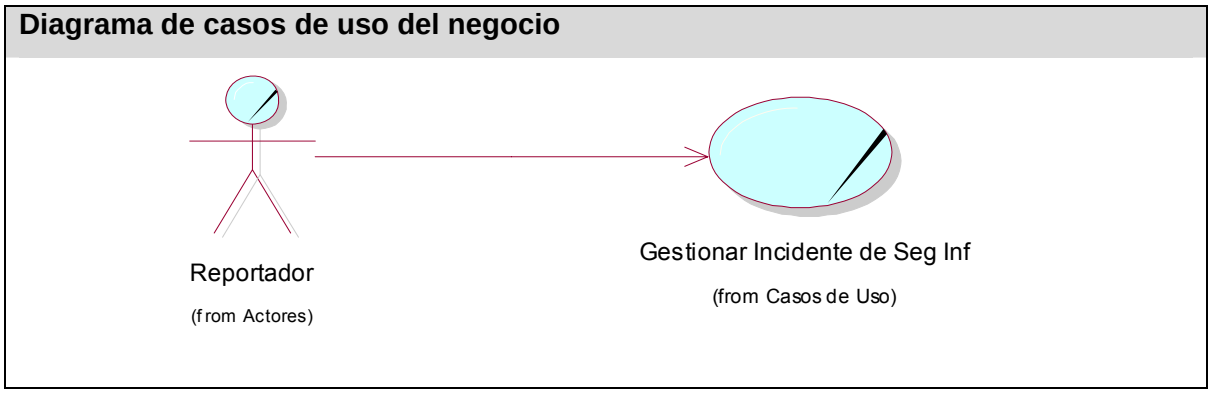

Figura 4: Diagrama de casos de uso del negocio.

2.3.6 Descripción de los Casos de Uso del Negocio.

Nombre del Caso de Uso	**Gestionar Incidente de Seguridad Informática**
Actores	Reportador (inicia).
Propósito	Permitir al reportador iniciar las acciones de la Gestión de incidentes de Seguridad Informática.
Resumen	El caso de uso se inicia cuando el reportador reporta un incidente de seguridad informática, este incidente es analizado por el investigador el cual será el encargado de darle respuesta.
Curso Normal de los eventos	
Acciones del Actor	**Respuesta del proceso de negocio**
1. El reportador reporta el Incidente.	
	2. El investigador analiza el Incidente, lo clasifica, le da la prioridad. Si esta es alta lo comienza a resolver.
	3. Si se resuelve el incidente lo archivan y toman las medidas pertinentes.
	4. Le informa al reportador de la respuesta al incidente.
5. El reportador recibe la información.	
Curso alternativo de los eventos	

Acción 2	De no ser la prioridad alta el incidente debe ser archivado y debe esperar por que su prioridad cambie.
Acción 3	Si el incidente no se resuelve es necesario solicitar nuevos datos al reportador.
Prioridad	Crítico.
Mejoras	
Otros	

Tabla 3: Descripción del CU del Negocio: Gestionar Incidente.

2.3.7 Diagramas de Actividad del Negocio.

Los Diagramas de Actividades ayudan a describir detalladamente que es lo que pasa dentro del negocio. Ellos detallan los roles específicos que juegan las personas (trabajadores del negocio) y las actividades que realizan. Estos diagramas ayudan a identificar que funciones deberá asumir el producto de software, y quiénes serán los actores del futuro sistema. (Ver Figura 35 Anexo 1.)

2.3.8 Diagrama de Clases del Modelo de objetos del Negocio.

El diagrama de clases, como artefacto que se construye para describir el modelo de objetos del negocio, muestra la participación de los trabajadores y entidades del negocio y la relación entre ellos. (Ver Figura 36 Anexo 1)

2.4 Especificación de los requisitos de software.

2.4.1 Requerimientos Funcionales.

De acuerdo con los objetivos planteados el sistema debe ser capaz de:

R1 Gestionar Incidente

R1.1 Reportar Incidente.

R1.2 Modificar Incidente

R1.3 Eliminar Incidente

R2 Gestionar Investigador

R2.1 Insertar Investigador

R2.2 Modificar Investigador

R2.3 Eliminar Investigador

R3 Autenticar Usuario

R4 Generar Reportes

R5 Asignar incidentes

R6 Gestionar Noticias

R6.1 Insertar Noticia

R7 Gestionar Solución a Incidentes

R7.1 Mostrar solución de Incidentes

R7.2 Insertar solución de Incidentes

R7.3 Eliminar solución de Incidentes

2.4.2 Requerimientos No Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener:

Apariencia o interfaz externa.

La interfaz del sistema será a través de una aplicación Web. Diseño sencillo, permitiendo que no sea necesario mucho entrenamiento para utilizar el sistema. Debe ser además un diseño formal y serio teniendo en cuenta el fin con el que se desarrolla la aplicación.

Usabilidad.

El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente Web en sentido general.

Rendimiento.

Debido a la importancia de la información que procesamos en nuestra aplicación, el sistema deberá de ser lo más estable y confiable posible. Además se garantizará que la respuesta a solicitudes de los usuarios del sistema sea en un período de tiempo breve para evitar la acumulación de trabajo por parte de los investigadores.

Soporte.

Servidor:

Se requiere de cualquier servidor con las siguientes características:

Apache (Servidor Web) Versión 2 o superior

PHP (Lenguaje de programación) Versión 5 o superior

MySQL (Gestor de Bases de Datos) Versión 5 o superior

Cliente:

Por parte del cliente se requiere un navegador capaz de interpretar JavaScript.

Portabilidad.

El producto podrá ser usado bajo los sistemas operativos de Windows y Linux. Corre sobre una plataforma Web, codificada en PHP y sus sistemas de bases de datos en MySQL.

Seguridad.

- Existencia de distintos roles que establezcan las acciones que pueden realizar los usuarios.
- Identificar al usuario antes de que pueda realizar cualquier acción sobre el sistema.
- Verificación sobre acciones irreversibles (por ejemplo las eliminaciones).
- El sistema constará de varios niveles de acceso. Un primer nivel o nivel básico donde están las funciones asociadas al usuario general o común, que requieren poca responsabilidad. El segundo nivel esta compuesto por funciones de mayor complejidad y que pueden destruir información relacionada a las entidades del sistema, a este nivel pertenecen los investigadores. El tercer nivel esta conformado con las funciones administrativas del sitio y del sistema.
- Se usan mecanismos de encriptación de los datos que por cuestiones de seguridad no deben viajar al servidor en texto claro, como es el caso de las contraseñas.

Políticos-Culturales y Legales.

El sistema cumple con todas las normas y leyes establecidas en nuestro país.

Confiabilidad.

- Garantía de un tratamiento adecuado de las excepciones y validación de las entradas del usuario para evitar entradas inadecuadas.
- El sistema debe tener soporte para recuperación ante fallos y errores.

2.5 Modelo del Sistema.

2.5.1 Actores.

Los actores del sistema pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado, son parte del sistema, y pueden intercambiar información con él o ser recipientes pasivos de información.

Actores	Justificación
Usuario	Cualquier persona que necesite reportar un incidente de seguridad en la UCI. Puede ser incluso, un investigador, un administrador o cualquier

	persona que quiera entrar al sistema.
Investigador	Persona encargada de gestionar los incidentes de seguridad.
Administrador	Persona encargada de gestionar a los recursos del sistema.

Tabla 4: Actores del sistema.

2.5.2 Casos de Usos.

Cod.	Nombre de caso de uso	Paquete	Justificación de la selección.
1.	Autenticarse	Seguridad	Este caso de uso es de vital importancia para la seguridad del sistema.
2.	Gestionar Investigador	Seguridad	Se gestionan toda la información de investigadores.
3.	Asignar Incidentes	Seguridad	Se asignan los incidentes a los investigadores.
4.	Insertar Noticias	Seguridad	Se insertan noticias relacionadas con la seguridad informática.
5.	Gestionar Incidente	Entrada de Datos	Se gestiona la información de los incidentes ya sea modificar o eliminar.
6.	Reportar Incidente	Entrada de datos	Se toma la información que brinda el reportador y se almacena en la base de datos para poder analizarla y darle una respuesta.
7.	Gestionar Solución a Incidentes	Entrada de datos	Se almacena la solución a los incidentes de seguridad luego de estar resueltos.
8.	Generar reportes	Salida de datos	El sistema muestra los reportes solicitados por el investigador.
9.	Mostrar Solución a Incidentes	Salida de datos	El sistema muestra la solución de los incidentes resueltos.

Tabla 5: Casos de Usos del Sistema.

2.5.3 Diagrama de Casos de Uso del Sistema.

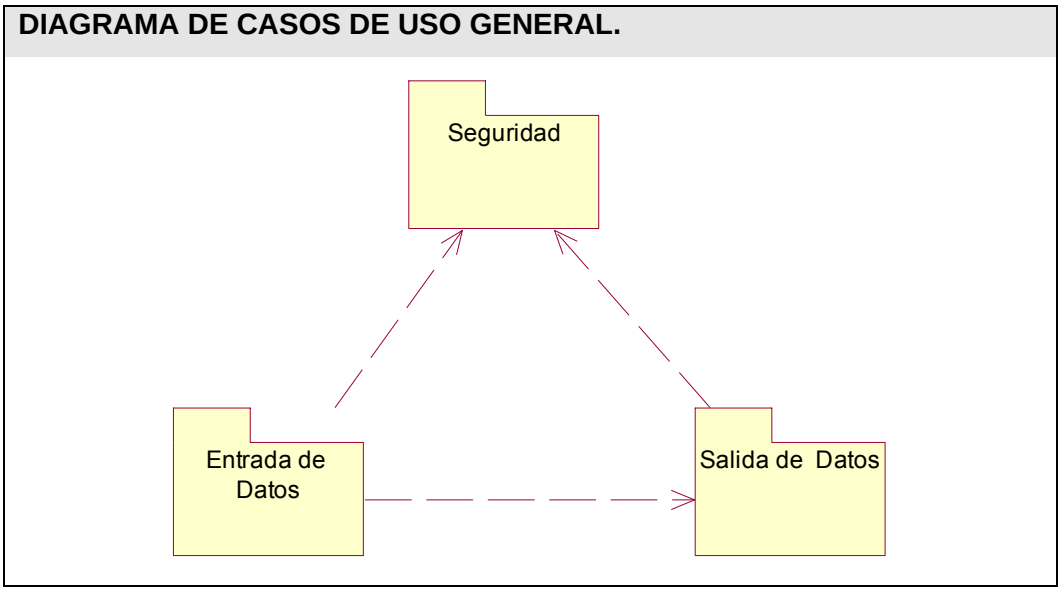

Figura 5: Diagrama de CU General por paquetes.

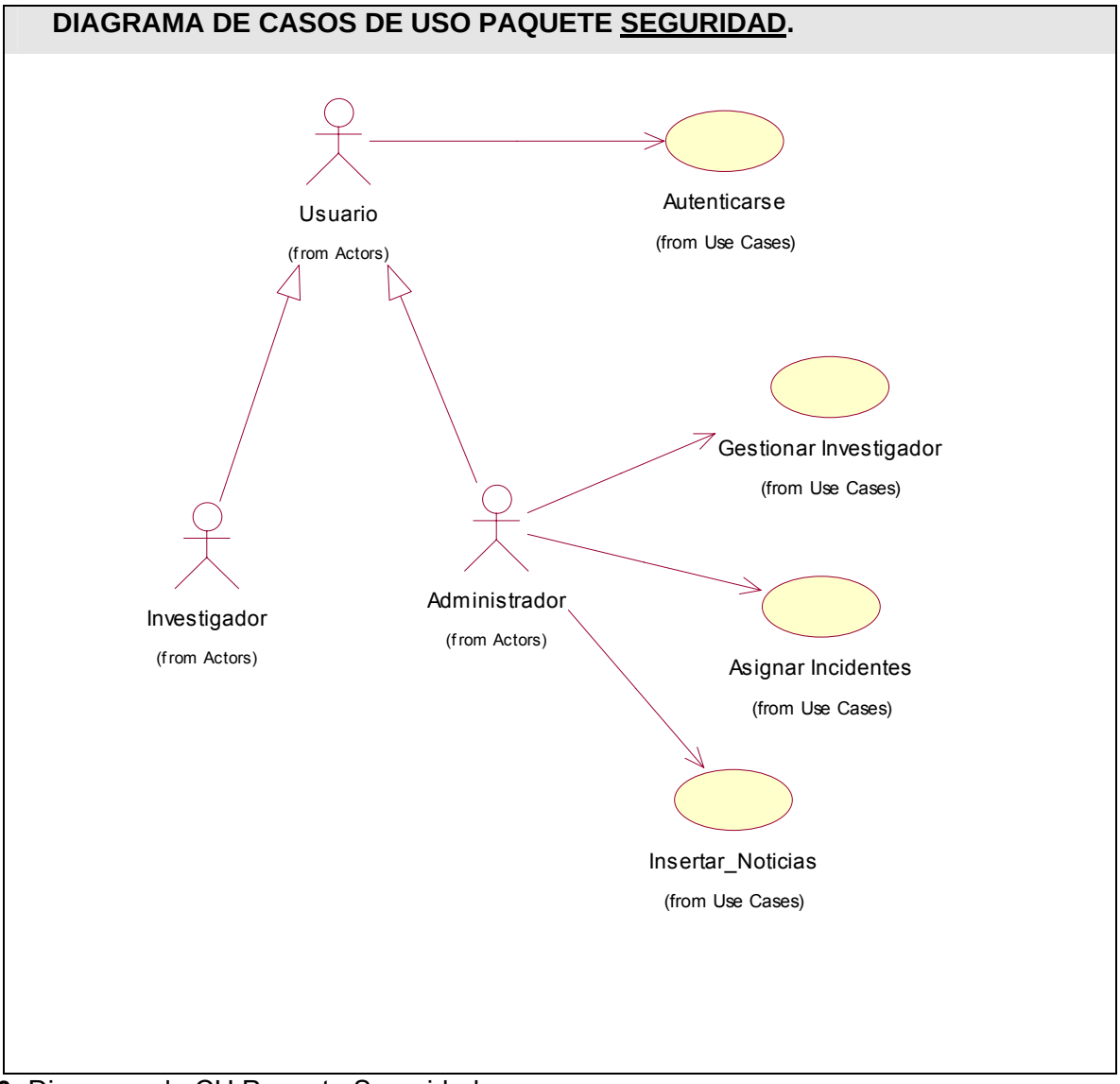

Figura 6: Diagrama de CU Paquete Seguridad.

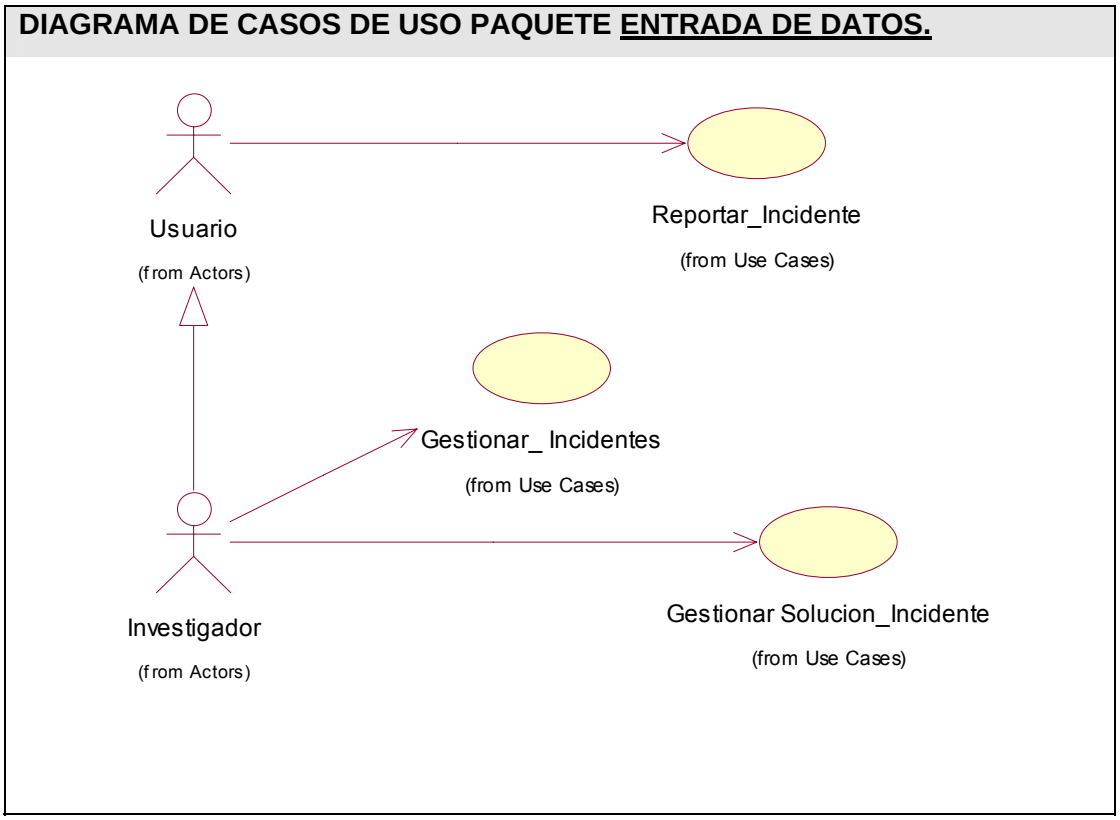

Figure 7: Diagrama de CU Paquete Entrada de Datos.

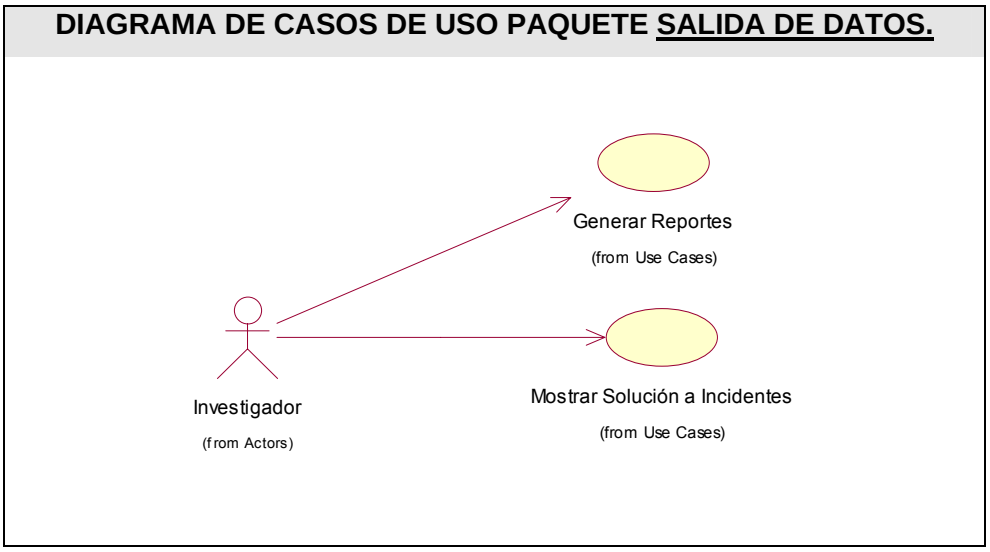

Figura 8: Diagrama de CU Paquete Salida de Datos.

2.5.4 Descripción de Casos de uso.

Paquete Seguridad.

Nombre del CU	**Autenticarse**
Actores	Usuario (inicia)
Objetivo	Brindar la sesión requerida en dependencia del usuario que sea.
Resumen	El CU se inicia cuando el usuario introduce sus datos del dominio uci en el sistema para autenticarse; y en este, según los datos introducidos le son asignados al usuario sus privilegios y en caso de que no sean correctos se le niega el acceso mostrando un mensaje de error.
Referencias	**R3.**
Precondiciones	Debe ser un usuario del dominio UCI.
Poscondiciones	El sistema otorga los permisos correspondientes a cada usuario.
Curso Normal de los Eventos	
Acciones del Actor	**Respuesta del Sistema**
	1. El sistema muestra la interfaz para autenticarse.
2. El usuario escribe usuario y contraseña del dominio uci.	2.1 El sistema verifica que los datos entrados por el usuario sean correctos. 2.2 El sistema muestra la información según los privilegios del usuario.
Curso Alterno de los Eventos	
Acción 2.1	Si los datos no son correctos el sistema muestra un mensaje donde lo informa y se retorna paso 2.
Prioridad	Crítico

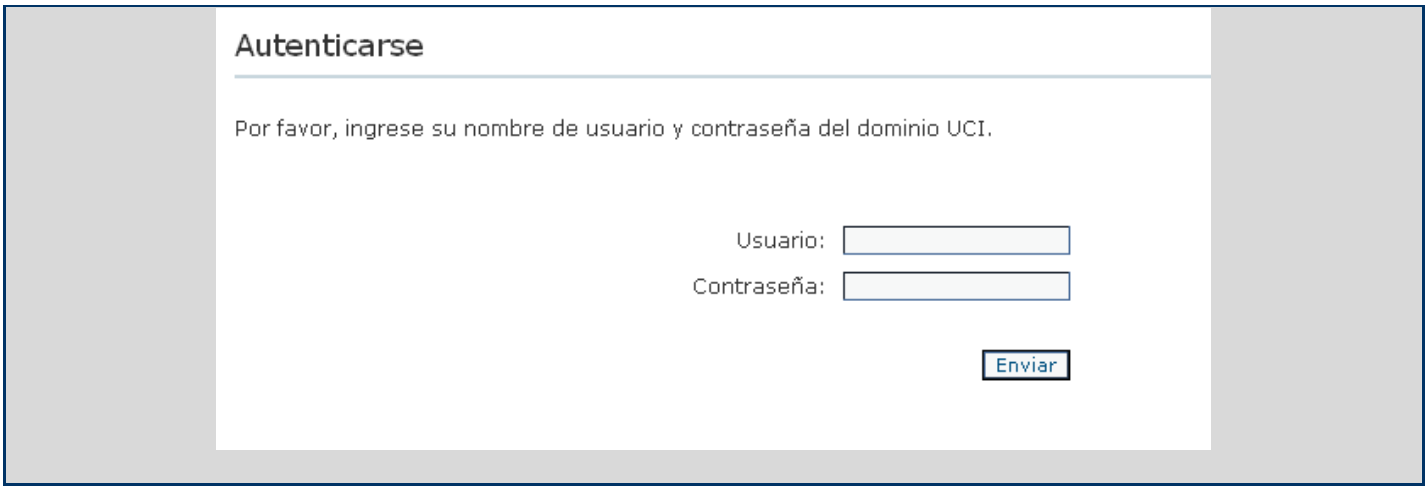

Tabla 6: Descripción del CU: Autenticarse.

Nombre del CU	**Gestionar Investigador**
Actores	Administrador (inicia).
Objetivo	Gestionar los datos de los investigadores.
Resumen	El administrador tiene la posibilidad de insertar, modificar o eliminar cada uno de los investigadores del sistema.
Referencias	**R2.**
Precondiciones	El Administrador debe estar previamente autentificado.
Pos condiciones	El Administrador debe haber insertado, eliminado o modificado uno o más investigadores.
Curso Normal de los Eventos	
Acciones del Actor	**Respuesta del Sistema**
1. El administrador selecciona la opción “Gestión de Investigadores” del menú.	2. El sistema muestra una interfaz con la lista de los investigadores y diferentes opciones que le permiten al administrador:
2.1 Registrar, ver sección “Registrar nuevo investigador”.	
2.2. Eliminar, ver sección “Eliminar investigador”.	
2.3. Modificar, ver sección “Modificar investigador”.	

Sección1. Registrar nuevo Investigador	
3.El administrador toma la opción de registrar nuevo investigador	4. El sistema muestra la interfaz para agregar al investigador
5. El administrador introduce los datos del nuevo investigador y selecciona la opción “Aceptar”.	6. El sistema valida los datos de entrada.
	7. El sistema almacena la información en su base de datos y se muestra el listado con el nuevo investigador.
	8. Retornar al paso 2, culminando así el caso de uso.
Curso alterno de los eventos	
Acción 6	6.1 El sistema solicita los datos del nuevo investigador ya que hubo un error a la hora de entrar los datos.
	6.2 El sistema retoma el paso 2.
Sección 2. Eliminar Investigador	
3. El administrador elige la opción “Eliminar investigador”.	4. El sistema elimina el investigador seleccionado de la base de datos.
	5. El sistema muestra el nuevo listado sin el investigador
	6. Retomar el paso 2.
Sección 3. Modificar investigador.	
3. El administrador selecciona la opción de “Modificar investigador”.	4. El sistema modifica los datos seleccionados por el administrador.
	5. El sistema muestra el nuevo listado con los datos del investigador ya modificados.
	6. Retomar paso 2.
Curso alterno de los eventos	
3.1 El administrador selecciona la opción de cancelar	3.1.2 El sistema cancela la operación de modificar investigador.
	3.1.3 Retomar paso2.
Prioridad	Crítico

Tabla 7: Descripción del CU: Gestionar Investigador.

Nombre del CU	**Asignar Incidente.**
Actores	Administrador (inicia).
Objetivo	Asignar incidente de seguridad al investigador.
Resumen	El administrador tiene la responsabilidad de asignar los incidentes de seguridad que han sido reportados a los investigadores.
Referencias	**R5.**
Precondiciones	El administrador debe estar previamente autentificado.
Poscondiciones	El administrador debe haber asignado el incidente a un investigador.
Curso Normal de los Eventos	
Acciones del Actor	**Respuesta del Sistema**
	1. El sistema muestra el listado de incidentes sin asignar.
2. El administrador escoge el incidente que va a asignar.	3. El sistema busca en la base de datos el listado de los investigadores. 4. El sistema muestra el listado de los investigadores.
5. El administrador selecciona el investigador del listado mostrado y selecciona el botón ¨ Aceptar ¨.	6. El sistema le asigna el incidente al investigador. 7 El sistema muestra el listado actualizado.
Curso Alterno de los Eventos	
Acción 5.1. El administrador selecciona el botón ¨Cancelar¨.	6.1 El sistema muestra el listado sin actualizar.
Prioridad	Crítico

Tabla 8: Descripción del CU: Asignar Incidente.

Nombre del CU	**Insertar noticia.**
Actores	Administrador (inicia).
Objetivo	Insertar noticias en el sistema.
Resumen	El Administrador tiene la responsabilidad de insertar las noticias en el sistema.
Referencias	**R6.1**
Precondiciones	El administrador debe estar previamente autentificado.
Poscondiciones	El administrador debe haber insertado la noticia.
Curso Normal de los Eventos	
Acciones del Actor	**Respuesta del Sistema**
	1. El sistema muestra la interfaz para insertar la noticia.
2. El administrador llena los datos requeridos para insertar la noticia y selecciona la opción ¨Guardar¨.	3. El sistema verifica que los datos están correctos. 4. El sistema almacena la noticia.
Curso Alterno de los Eventos	
Acción 2.1. El administrador selecciona el botón ¨Cancelar¨.	6.1 El sistema cancela la opción y no se almacena la noticia.
	3.1 El sistema verifica que los datos no están correctos y retorna al paso 2.
Prioridad	Secundario

Tabla 9: Descripción del CU: Insertar Noticia.

Paquete Entrada de Datos.

Nombre del CU	**Gestionar Incidentes**
Actores	Investigador (inicia).
Objetivo	Gestionar los incidentes de seguridad.
Resumen	El sistema permite que el investigador realice algunas operaciones, estas pueden ser modificar el estado y prioridad de los incidentes, así como eliminarnos.
Referencias	**R1.2, R1.3**
Precondiciones	Solo el investigador puede realizar los cambios, por tanto debe estar autenticado en el sistema.
Poscondiciones	Quedan registrados incidentes de seguridad.

Curso Normal de los Eventos	
Acciones del Actor	**Respuesta del Sistema**
1. El administrador selecciona la opción “Gestión de incidentes” del menú.	2. El sistema muestra una interfaz con la lista de los incidentes que el investigador tiene pendientes a resolver

	y muestra entonces de cada incidente diferentes opciones a realizar como son:
2.1 Modificar, ver sección "Modificar Incidente".	
2.2. Eliminar, ver sección "Eliminar Incidente".	
Sección1. Modificar Incidente.	
3. El administrador toma la opción de modificar incidente.	4. El sistema muestra la interfaz para modificar el incidente.
5. El administrador puede modificar tanto el estado como la prioridad del incidente y selecciona la opción "Aceptar".	6. El sistema actualiza la bases de datos y almacena la nueva información
	7. El sistema muestra el incidente ya modificado.
	8. Retornar al paso 2, culminando así el caso de uso.
Sección 2. Eliminar Incidente.	
3. El administrador elige la opción "Eliminar incidente".	4. El sistema elimina el incidente seleccionado de la base de datos.
	5. El sistema muestra el nuevo listado sin el incidente.
	6. Retomar el paso 2.
Prioridad	Crítico

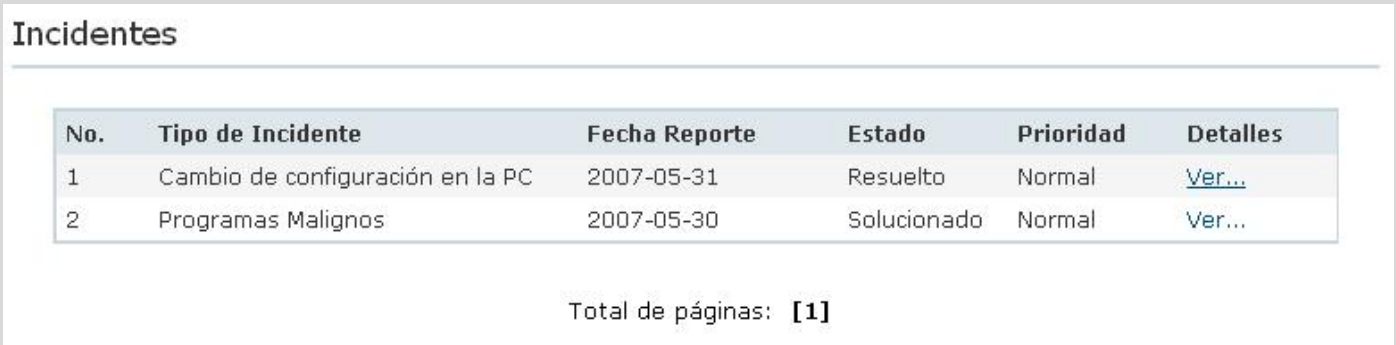

Tabla 10: Descripción del CU: Gestionar Incidente.

Nombre del CU	**Reportar Incidente**
Actores	Usuario (inicia)
Objetivo	Reportar los incidentes de seguridad.

Resumen	El caso de uso se inicia cuando el reportador solicita reportar un incidente de seguridad. El sistema muestra una planilla con los datos que el reportador debe llenar para tener una mayor información sobre el incidente ocurrido.
Referencias	**R1.1**
Precondiciones	El usuario debe autenticarse para registrar un reporte.
Poscondiciones	Queda registrado un nuevo incidente de seguridad.

Curso Normal de los Eventos	
Acciones del Actor	**Respuesta del Sistema**
1. El reportador selecciona la opción "Reportar Incidente" del menú de opciones.	2. El sistema muestra un formulario que el reportador debe llenar.
	3. El sistema al autenticarse el reportador tomas sus datos automáticamente.
4. El reportador llena la planilla con los datos necesarios y da en el botón "Aceptar".	5. El sistema verifica que todos los datos entrados por el usuario estén correctos.
	6. El sistema almacena todos los datos registrados por el usuario.
	7. El sistema muestra un mensaje de confirmación que contiene el número de reporte.
	8. El sistema envía un e-mail de confirmación al reportador.
	9. El sistema asigna el incidente al investigador que menos casos pendientes tenga.
Curso alterno de los eventos.	
4. El reportador desea cancelar el llenado de planilla, para esto oprime el botón "Cancelar".	4.1 El sistema cancela la planilla realizada por el reportador.
	5.1 Si los datos no están correctos, se muestra un mensaje que especifique los errores y se retorna al paso 4.
Prioridad	Crítico

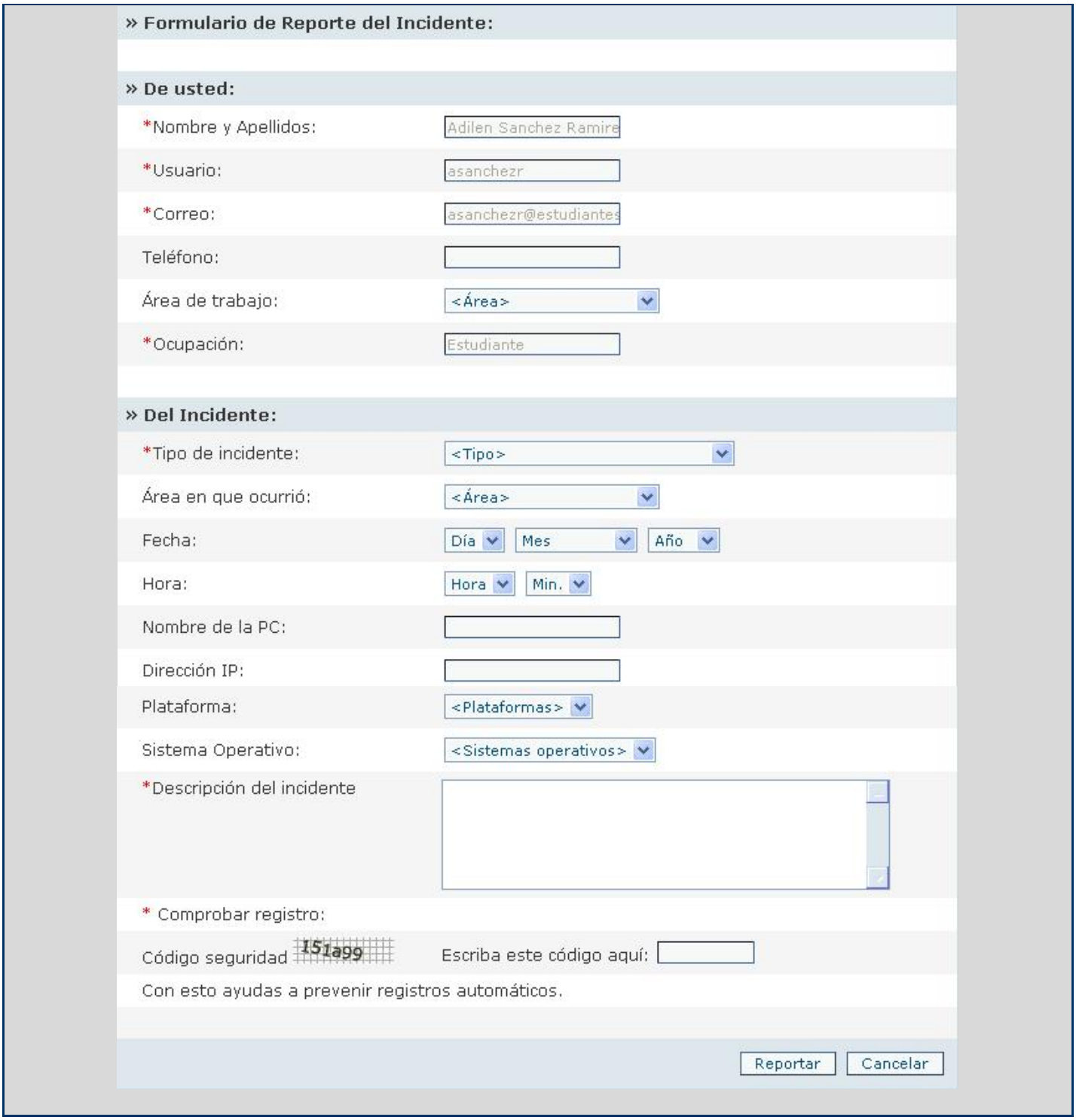

Tabla 11: Descripción del CU: Reportar Incidente

Nombre del CU	**Gestionar Solución a Incidentes.**
Actores	Investigador (inicia)
Objetivo	Darle solución a los incidentes una vez resueltos.
Resumen	El investigador puede almacenar la solución a los incidentes una vez resueltos los mismos.
Referencias	**R7**
Precondiciones	El investigador debe estar previamente autenticado.
Pos condiciones	Queda almacenada la solución al incidente.

Curso Normal de los Eventos	
Acciones del Actor	**Respuesta del Sistema**
	1. El sistema muestra el listado de los incidentes resueltos por el investigador.
2. El investigador selecciona el incidente al que va a insertarle la solución.	3. El sistema muestra la interfaz para insertar los datos de la solución.
4. El investigador llena los datos solicitados y le da al botón ¨Aceptar¨.	5. El sistema verifica que los datos estén correctos. Retornar Acción 4.
6. El investigador desea terminar el llenado de solución a incidentes y le da al botón ¨Terminar¨	7. El sistema retorna a la página principal de la sesión del investigador.
Curso alterno de los eventos.	
4.1 El investigador desea cancelar el llenado de la solución, para esto oprime el botón “Cancelar”.	4.1 El sistema cancela la solución realizada por el investigador.
	5.1 Si los datos no están correctos, se muestra un mensaje que especifique los errores y se retorna al paso 4.
Prioridad	Crítico

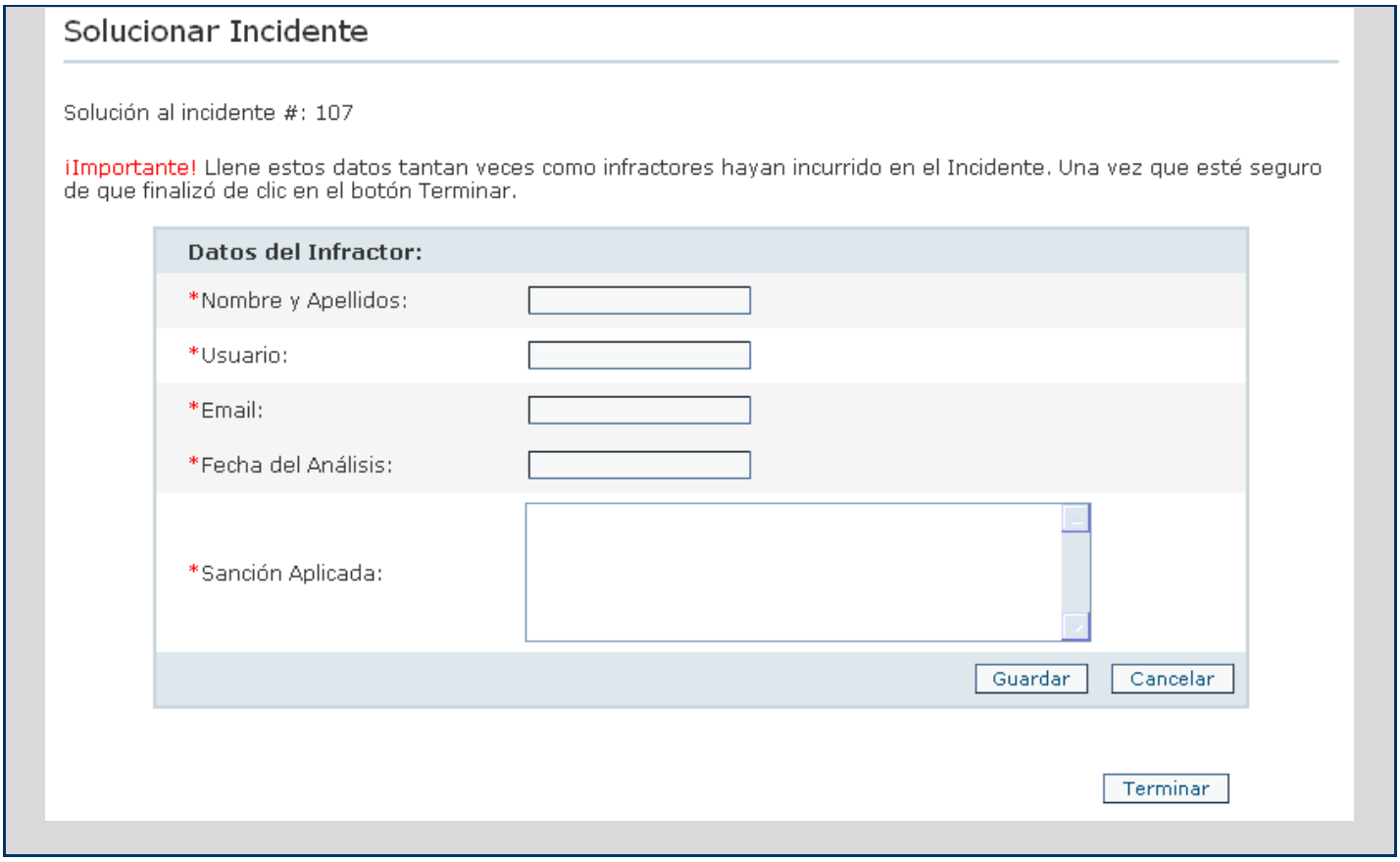

Tabla 12: Descripción del CU: Gestionar Solución de Incidentes.

Paquete Salida de datos.

Nombre del CU	**Generar Reportes.**
Actores	Investigador (inicia).
Objetivo	Generar un reporte específico solicitado por el investigador.
Resumen	El caso de uso se inicia cuando el investigador solicita un reporte sobre los incidentes almacenados en la base de datos del sistema. Se puede realizar reportes por diferentes criterios ya sean: tipo de incidentes, fecha, investigador que resolvió el incidente, por el estado que tiene el incidente ya sea Pendiente, En proceso, Concluido, o por la prioridad que le haya dado el investigador. El sistema brinda la posibilidad al investigador de escoger

	varias de estas categorías para hacer un mismo reporte.
Referencias	**R4.**
Precondiciones	El investigador debe estar previamente autentificado.
Poscondiciones	Se muestra el reporte solicitado.

Curso Normal de los Eventos	
Acciones del Actor	**Respuesta del Sistema**
1. El investigador selecciona la opción de "Generar reportes".	2. El sistema muestra la interfaz con las diferentes categorías que se le brinda al investigador para que realice el reporte.
3. El investigador selecciona la(s) categoría(s que se le brinda para realizar el reporte. Este puede realizar el reporte por tipo de incidente, área, fecha, investigador, estado y prioridad y puede seleccionar varias categorías al mismo tiempo.	4. El sistema realiza consultas a la base de datos en dependencia de la solicitud del investigador.
	5. El sistema muestra la información obtenida.
Curso alterno de los eventos.	
	2.1 El sistema no muestra el reporte porque no se seleccionó ninguna categoría y retorna al paso 3.
Prioridad	Crítico.

Tabla 13: Descripción del CU: Generar Reportes

Nombre del CU	**Mostrar Solución a Incidentes.**
Actores	Investigador (inicia).
Objetivo	Listar las soluciones de los incidentes almacenadas por el investigador.
Resumen	El CU se inicia cuando el investigador solicita ver el listado de las soluciones de los incidentes; en este el sistema le muestra el listado correspondiente al investigador.
Referencias	**R7.1**
Precondiciones	El investigador debe estar previamente autentificado.
Pos condiciones	Es mostrado el listado de soluciones a incidentes.
Curso Normal de los Eventos	
Acciones del Actor	**Respuesta del Sistema**
1. El investigador selecciona la opción de "Ver Solución a Incidentes".	2. El sistema busca en la base de datos la tabla Infractor.
	3.El sistema muestra el listado de soluciones
Curso alterno de los eventos.	
	3.1 El sistema no muestra el listado porque no existe ningún incidente resuelto hasta el momento.
Prioridad	Crítico.

Tabla 14: Descripción del CU: Mostrar Solución de Incidentes.

2.6 Conclusiones:

En este capítulo se han descrito los procesos del negocio relacionados con la gestión de incidentes de seguridad, en el mismo se detallaron los actores, casos de uso y entidades u objetos del negocio. Se elaboraron los modelos de casos de uso del negocio con la realización de cada caso de uso y de objetos del negocio. Con todo esto se pudo llegar a una mejor comprensión del problema a tratar. Además se comenzó a desarrollar la propuesta de solución haciendo uso de los procesos del negocio, definiendo las funcionalidades que debe tener del sistema así como las características que este debe tener llegando así al primer prototipo de interfaz del sistema.

Capítulo 3. Análisis y diseño del sistema.

3.1 Introducción.

Con las funcionalidades ya definidas y descritas en el flujo de trabajo de Requerimientos, se realiza en este capítulo el modelo del análisis y el diseño respectivamente para de esta forma describir detalladamente como se desarrollará la aplicación Web. El modelo del análisis ayuda a refinar los requisitos sobre los aspectos internos del sistema, incluidos sus recursos compartidos internos, y también ofrece una primera aproximación al diseño. El modelo del diseño pretende crear un plano del modelo del modelo de implementación, y debe ser mantenido durante todo el ciclo del software. Además se desarrollara el modelo de datos y se describen los estándares de diseño seguidos para desarrollar el sistema.

3.2 Modelo del Análisis.

El Modelo de Análisis ofrece una especificación más precisa de los requisitos que la que se tiene como resultado de la captura de requisitos, incluyendo al modelo de casos de uso. Este modelo se describe utilizando el lenguaje de los desarrolladores y puede por tanto introducir un mayor formalismo y ser utilizado para razonar sobre funcionamientos internos del sistema.

Un Modelo del Análisis puede considerarse como un primera aproximación al modelo de diseño (aunque es un modelo por si mismo), y es por tanto una entrada fundamental cuando se da forma al sistema en el diseño y en la implementación. Esto se debe a que debería ser mantenible el sistema en su conjunto y no solo la descripción de sus requisitos. [10].

3.2.1 Diagramas de clases de análisis.

3.2.1.1 Entrada de Datos.

CU Gestionar Incidente.

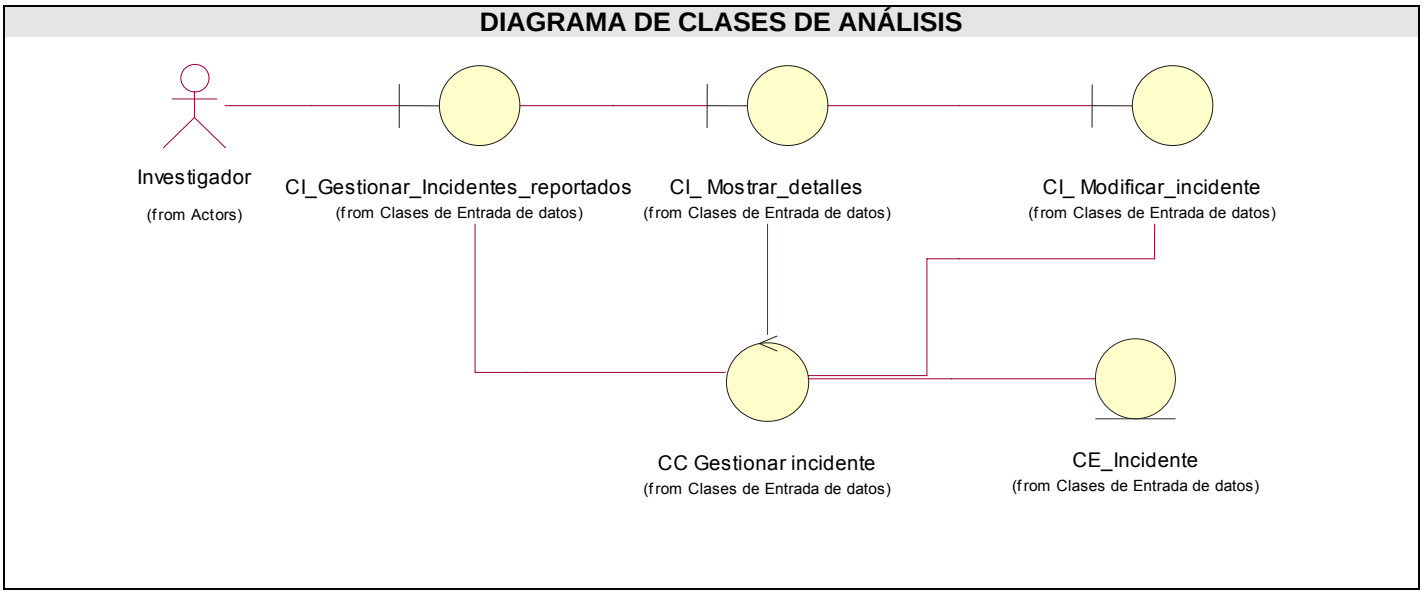

Figura 9: Diagrama de Clases del Análisis: CU Gestionar Incidente.

CU Gestionar Solución Incidente.

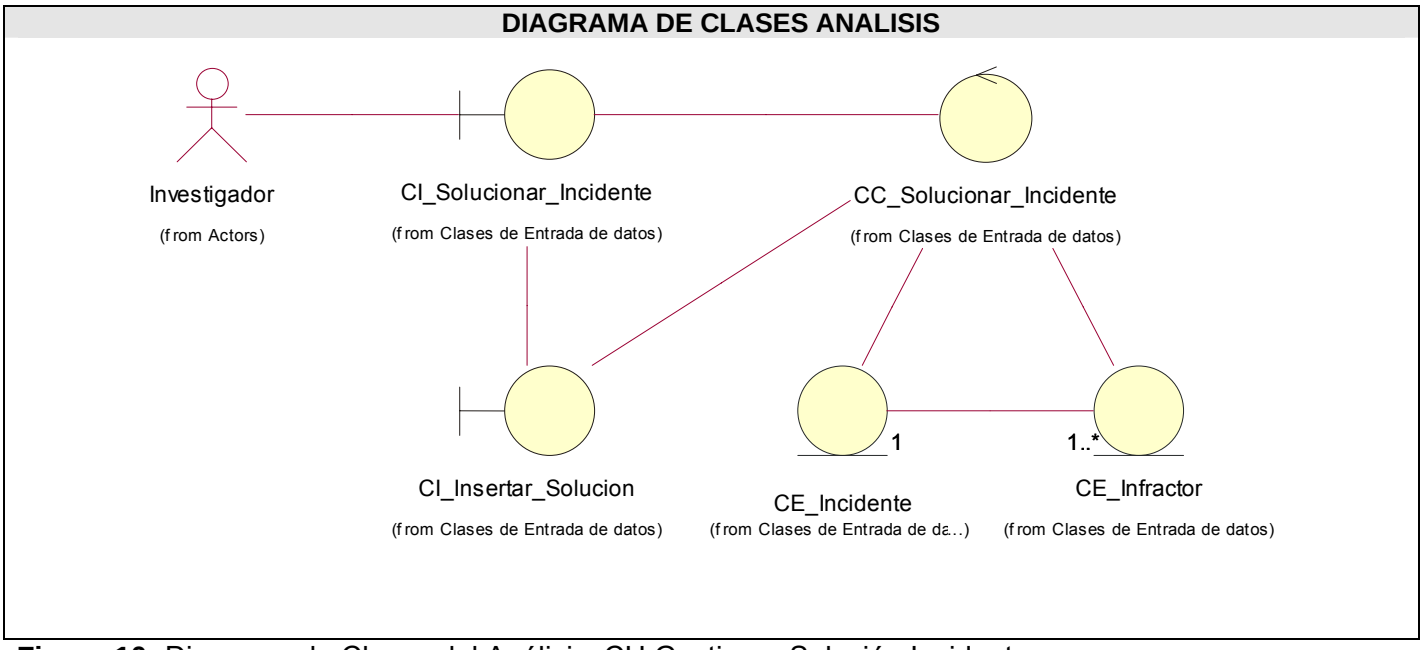

Figura 10: Diagrama de Clases del Análisis: CU Gestionar Solución Incidente.

CU Reportar Incidente.

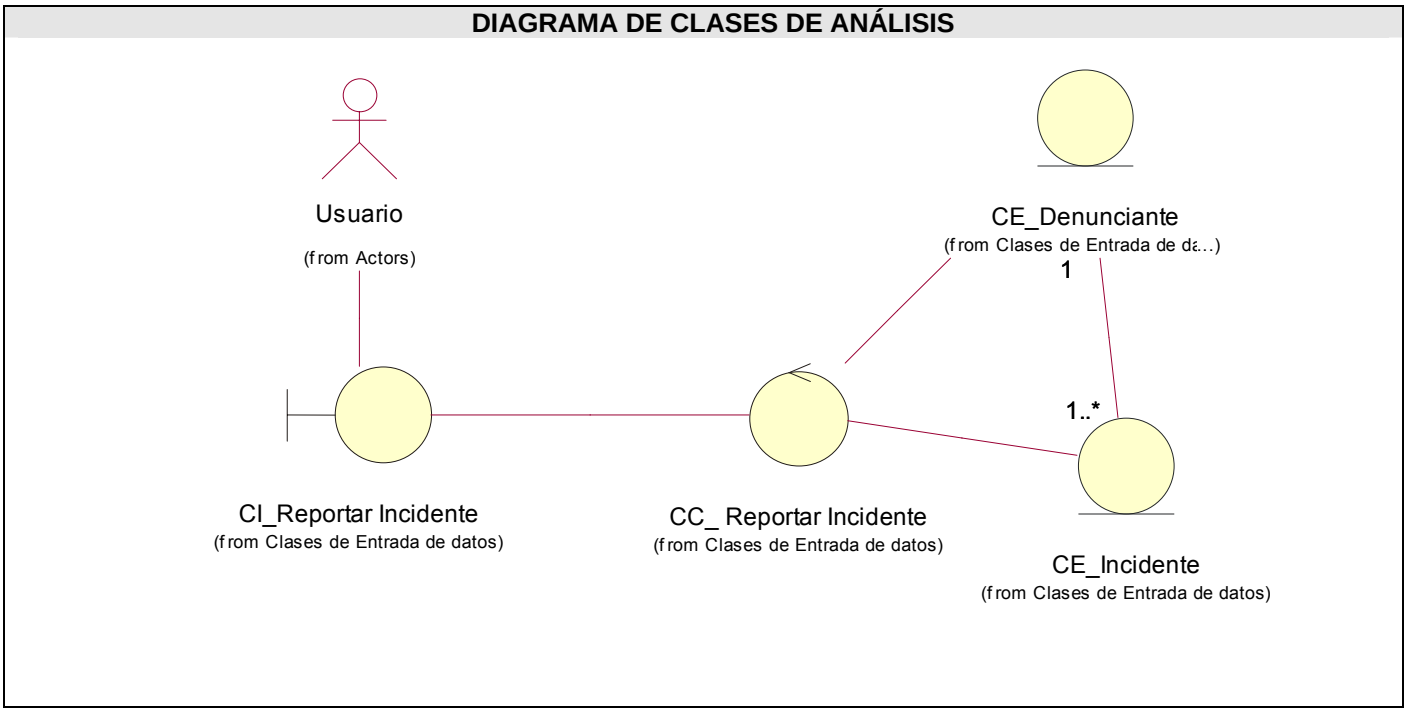

Figura 11: Diagrama de Clases del Análisis: CU Reportar Incidente.

3.2.1.2 Salida de Datos

CU Generar Reportes.

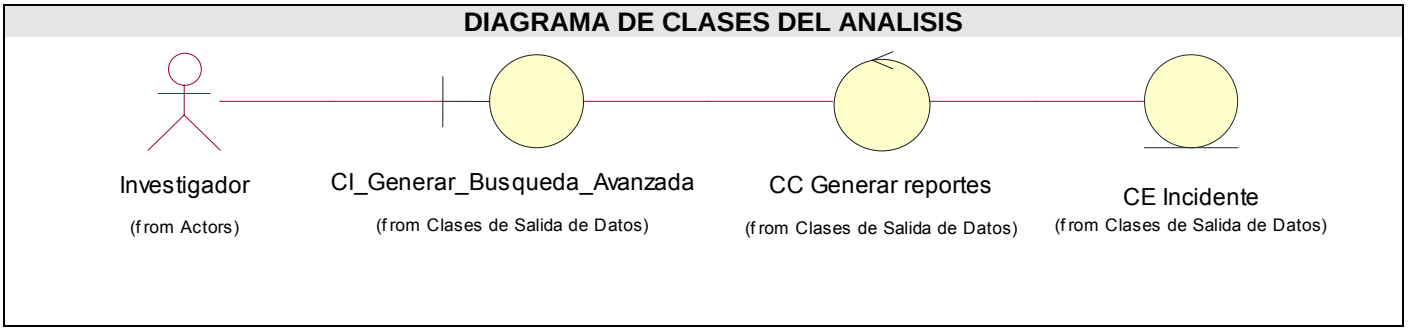

Figura 12: Diagrama de Clases del Análisis: CU Generar Reportes.

CU Mostrar Solución Incidentes.

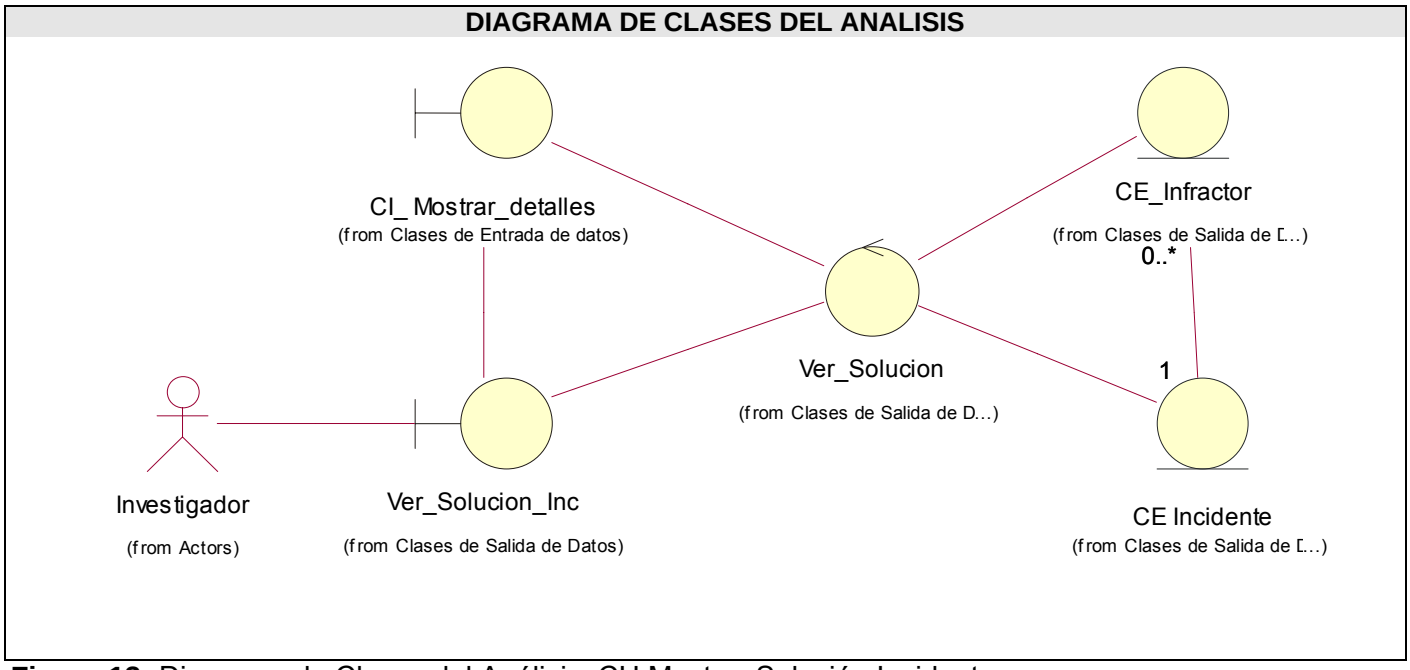

Figura 13: Diagrama de Clases del Análisis: CU Mostrar Solución Incidentes.

3.2.1.3 Seguridad

CU Autenticar usuario.

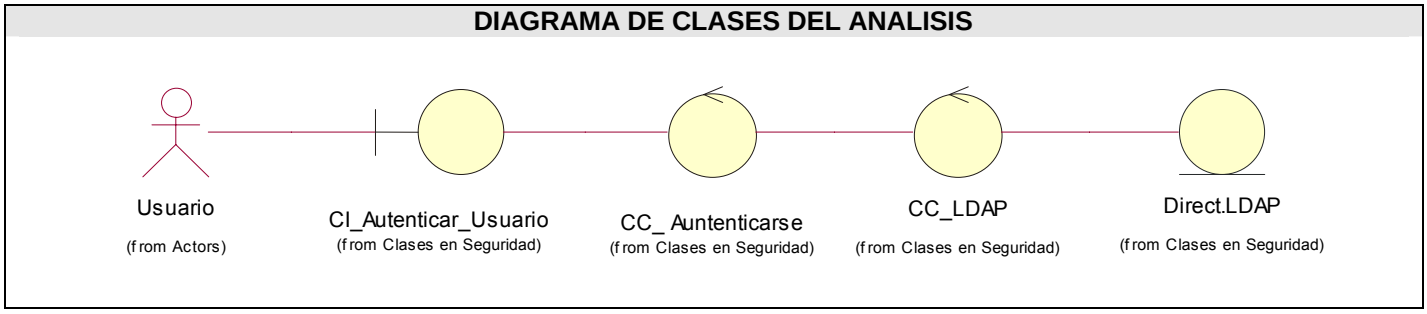

Figura 14: Diagrama de Clases del Análisis: CU Autenticar usuario.

CU Asignar Incidente.

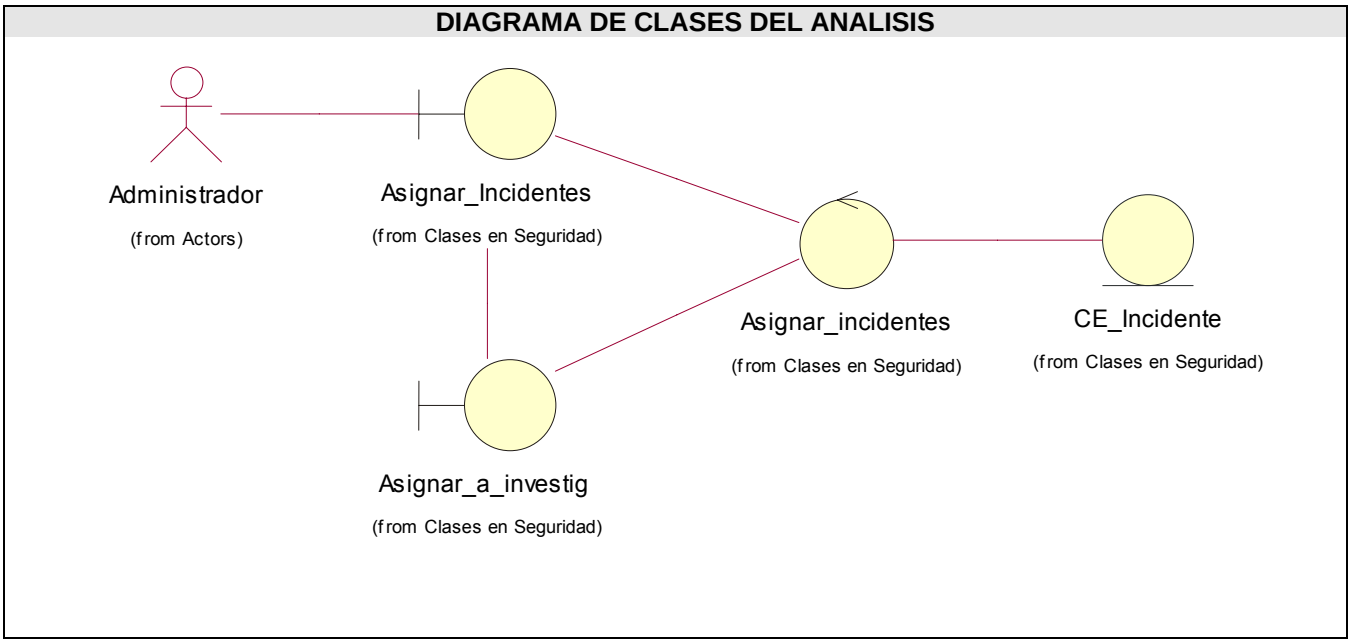

Figura 15: Diagrama de Clases del Análisis: CU Asignar Incidente.

CU Gestionar investigador.

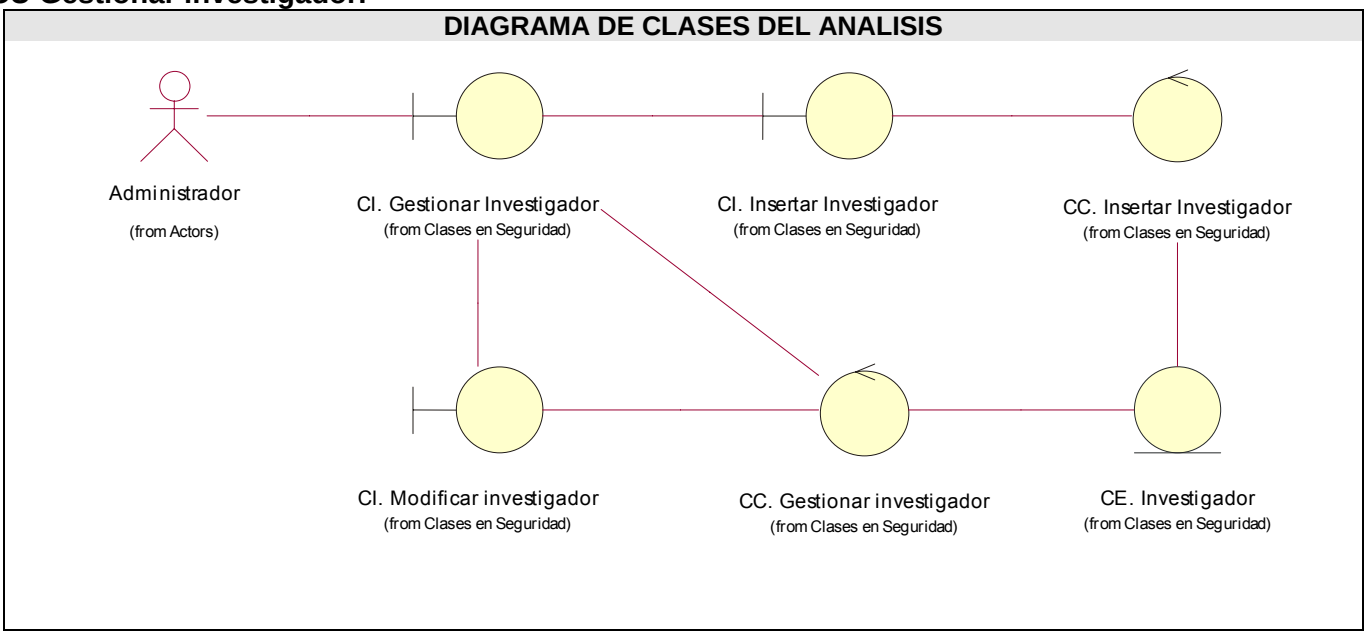

Figura 16: Diagrama de Clases del Análisis: CU Gestionar investigador.

CU Insertar Noticias

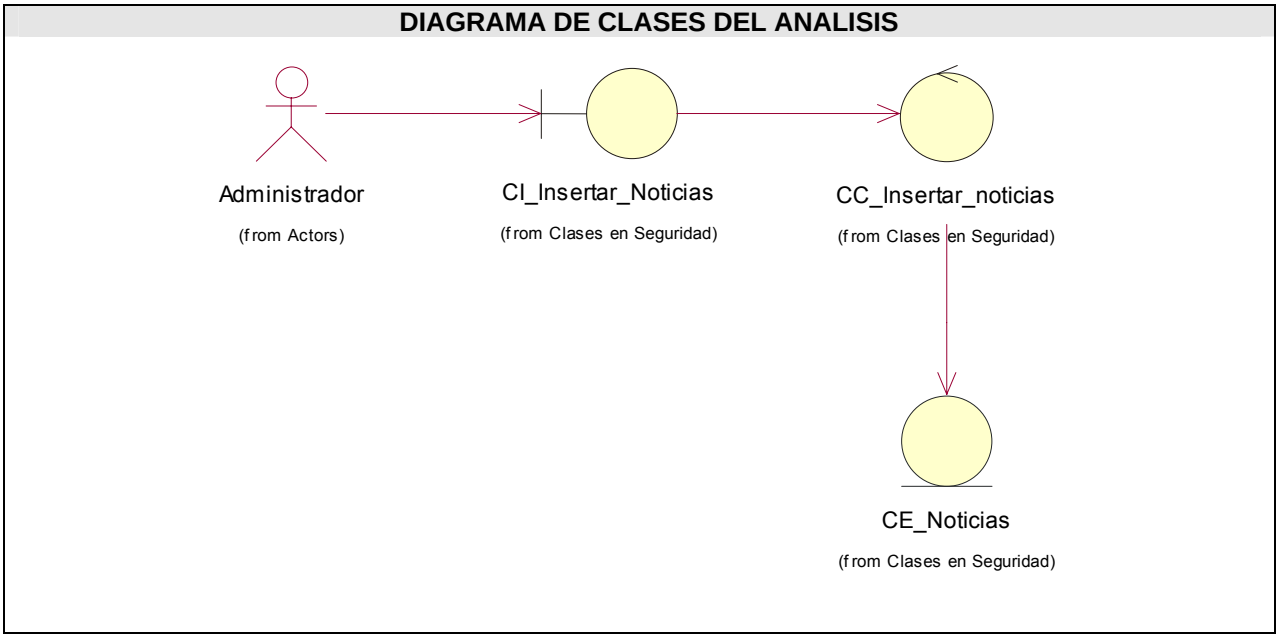

Figura 17: Diagrama de Clases del Análisis: CU Insertar Noticias.

3.3 Modelo del Diseño.

3.3.1 Diagramas de clases y de interacción del diseño.

3.3.1.1 Entrada de Datos.

CU Gestionar Incidente.

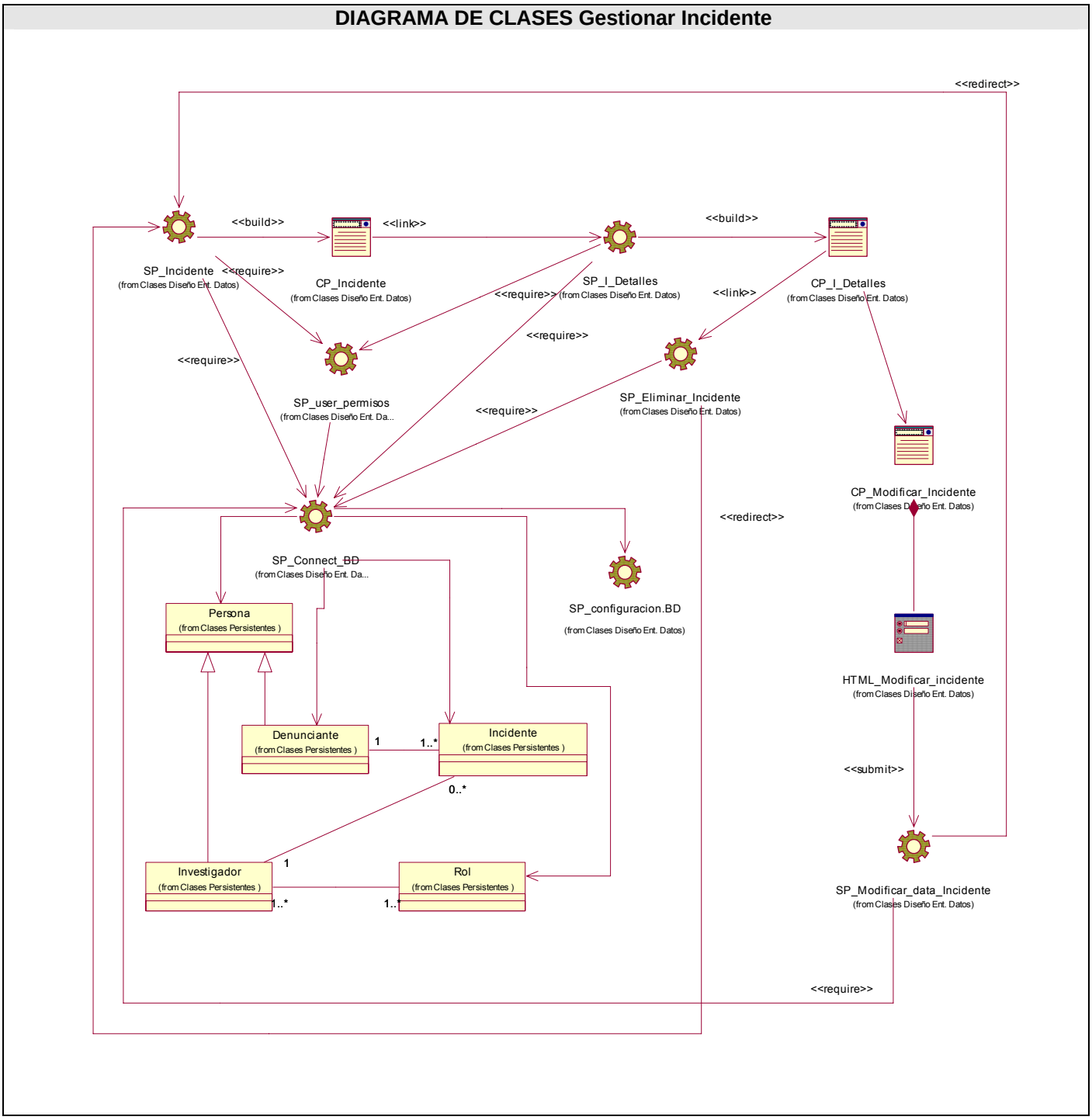

Figura 18: Diagrama de Clases del Diseño CU Gestionar Incidente.

CU Reportar Incidente.

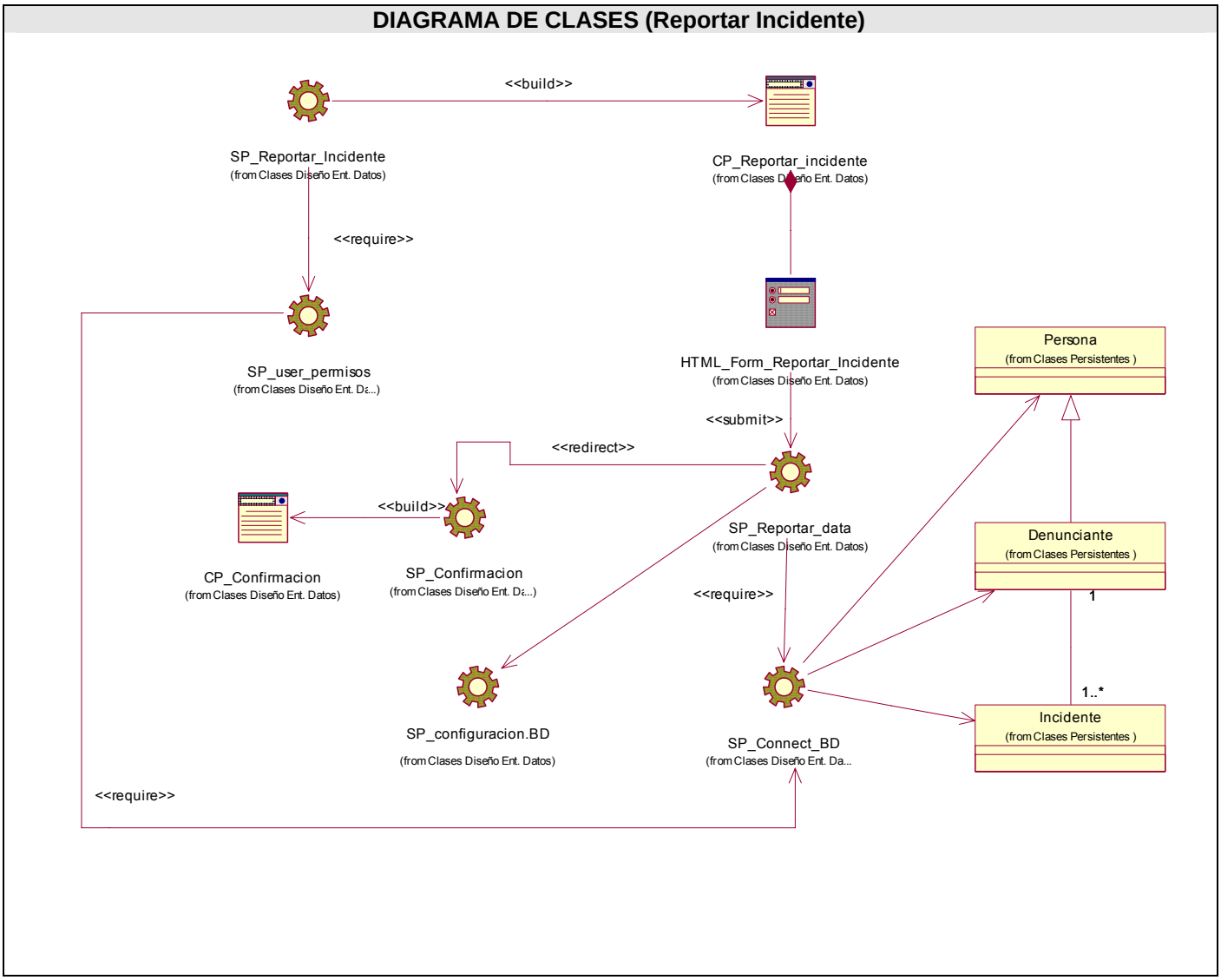

Figura 19: Diagrama de Clases del Diseño CU Reportar Incidente.

CU Gestionar Solución Incidente.

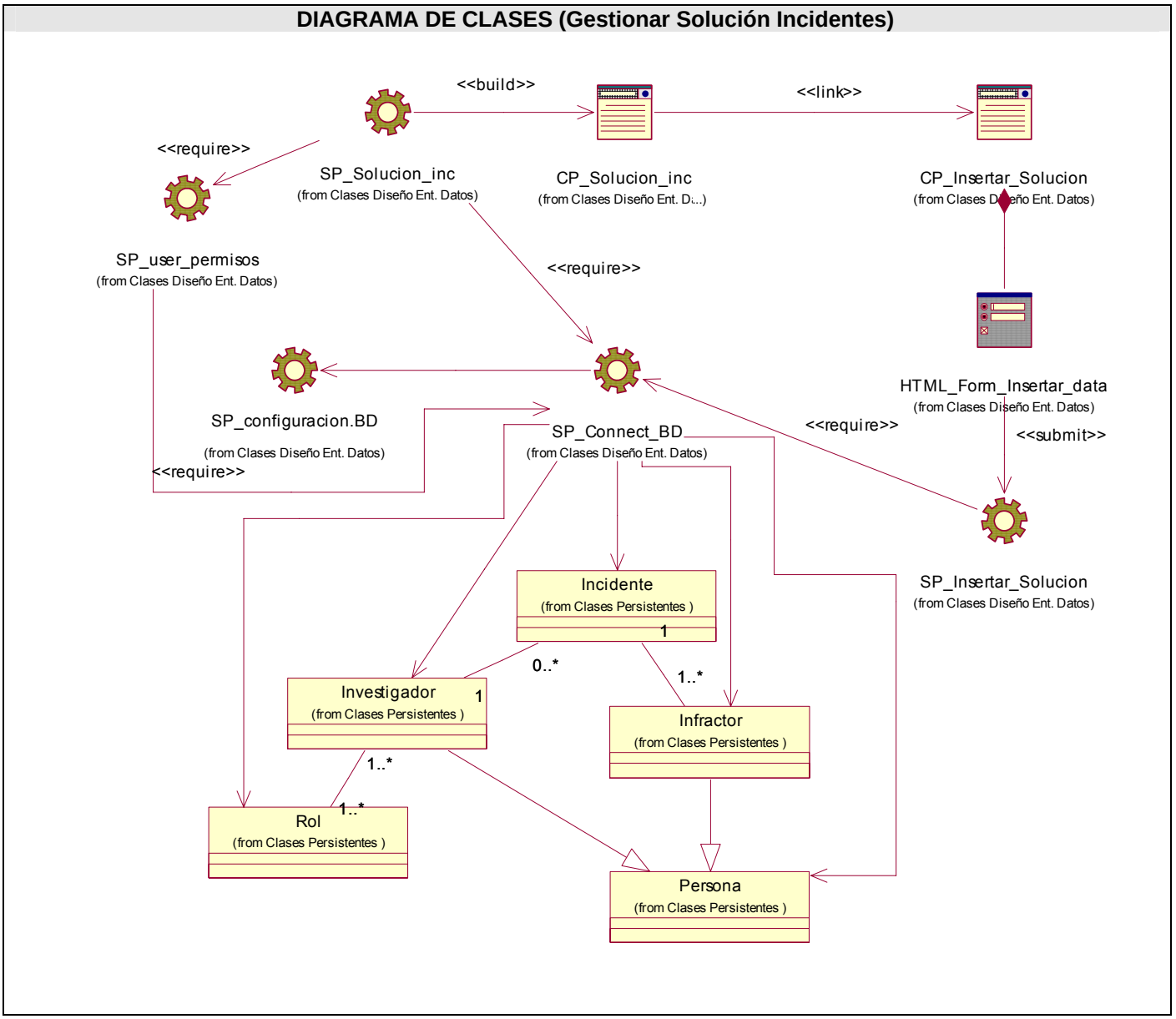

Figura 20: Diagrama de Clases del Diseño CU Gestionar Solución Incidentes.

3.3.1.2 Salida de Datos.

CU Generar Reportes

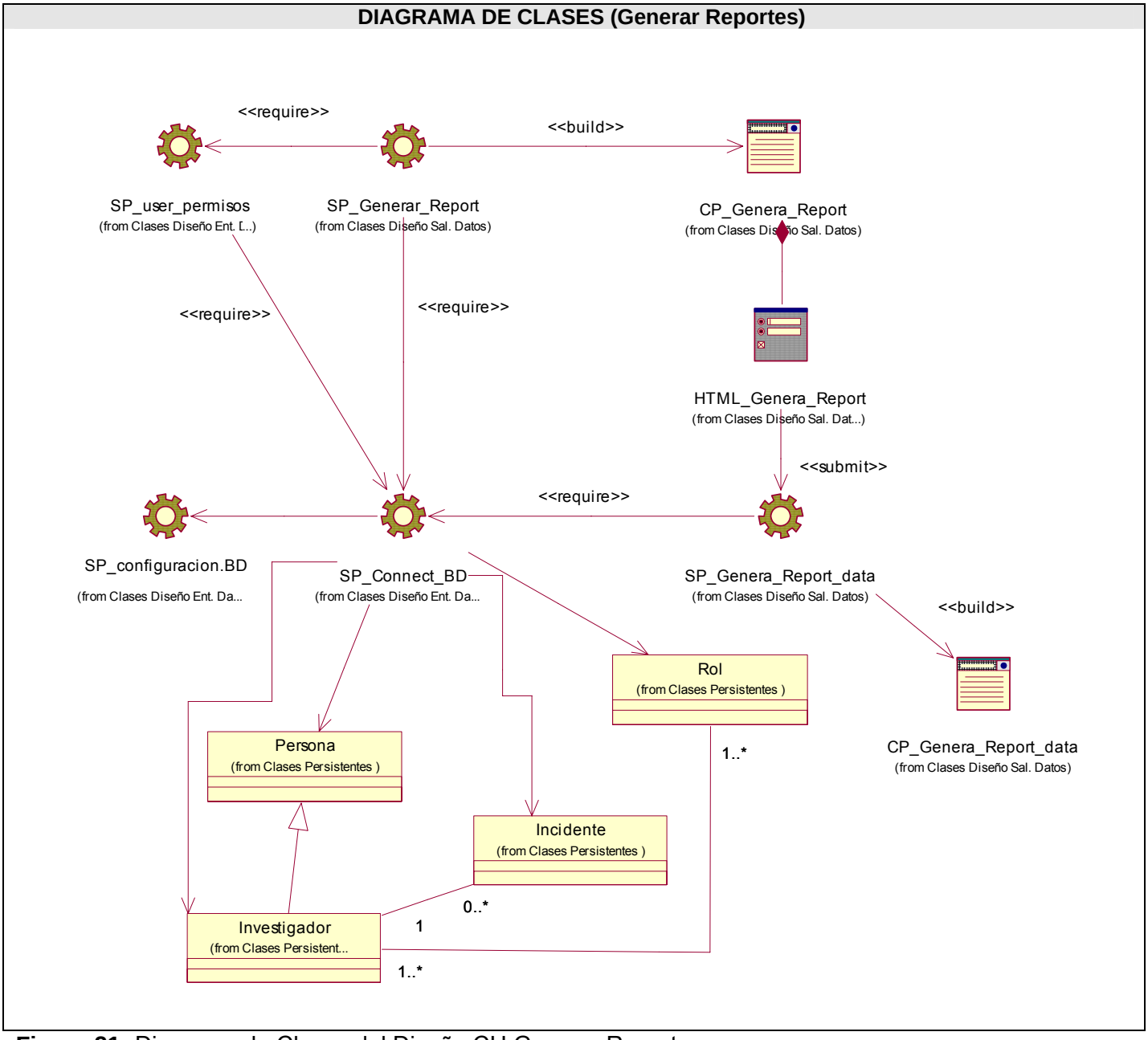

Figura 21: Diagrama de Clases del Diseño CU Generar Reportes.

CU Mostrar Solución Incidentes.

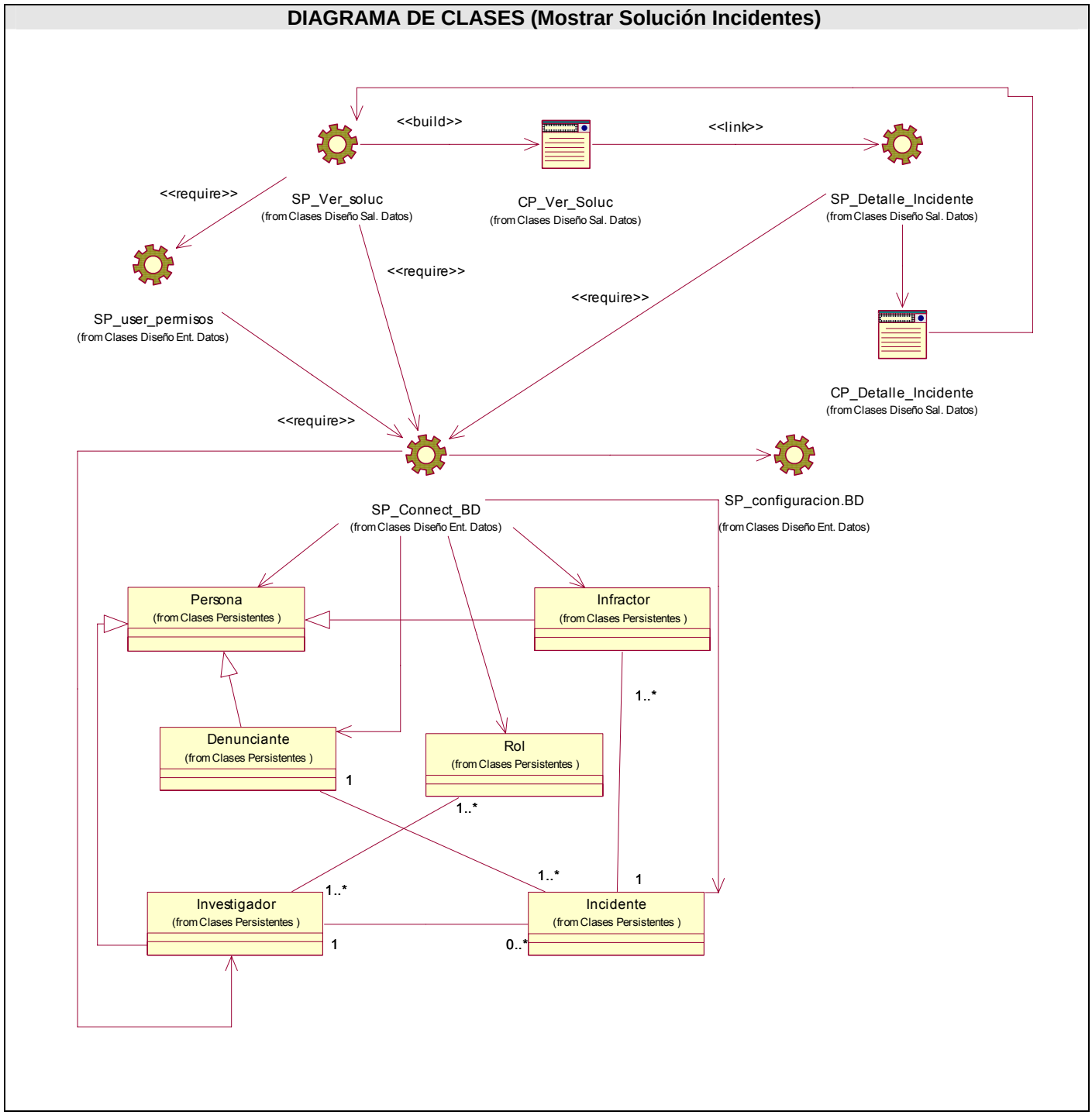

Figura 22: Diagrama de Clases del Diseño CU Mostrar Solución Incidentes.

3.3.1.3 Seguridad.

CU Asignar Incidentes.

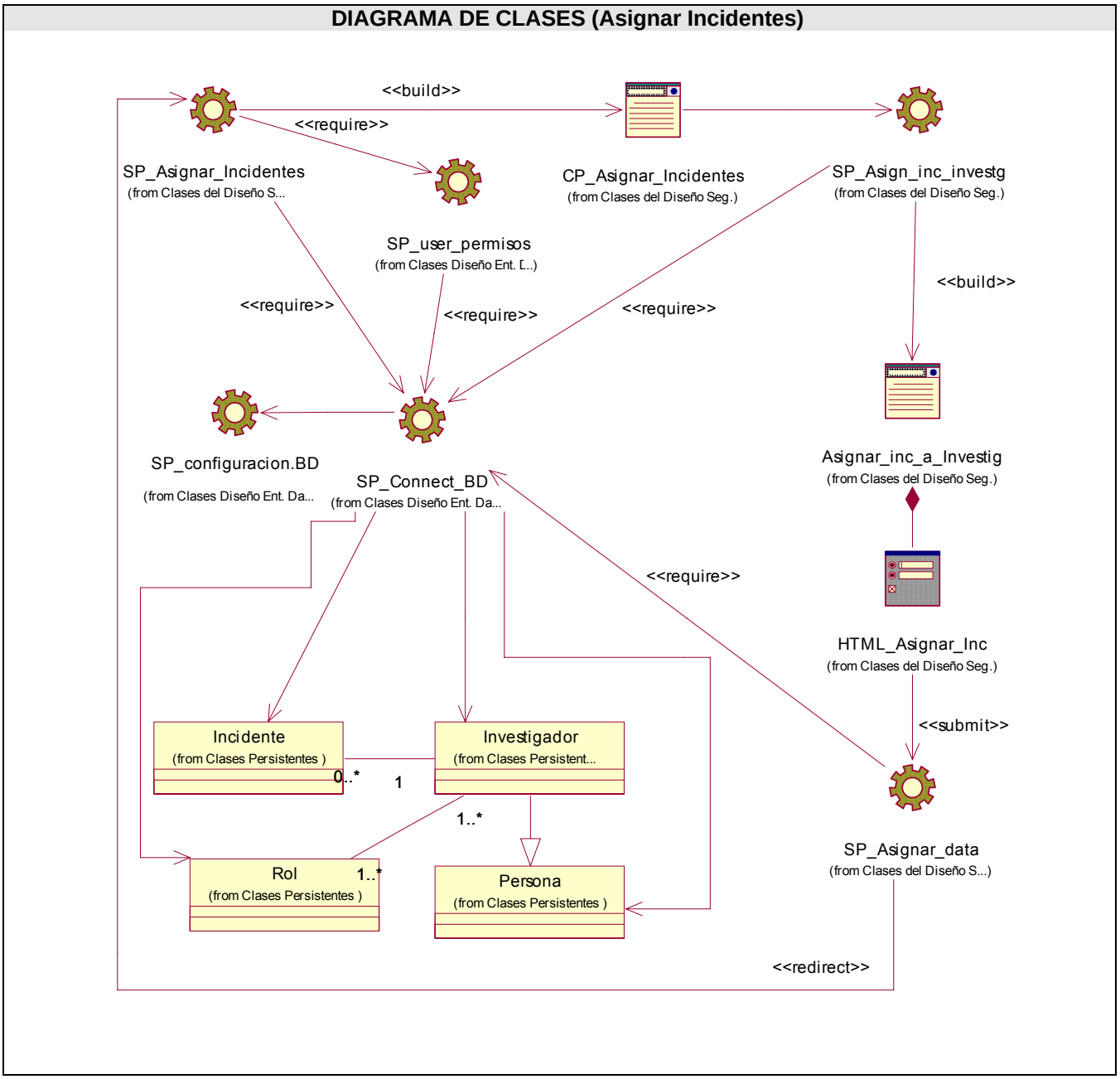

Figura 23: Diagrama de Clases del Diseño CU Asignar Incidentes.

CU Autenticar Usuario

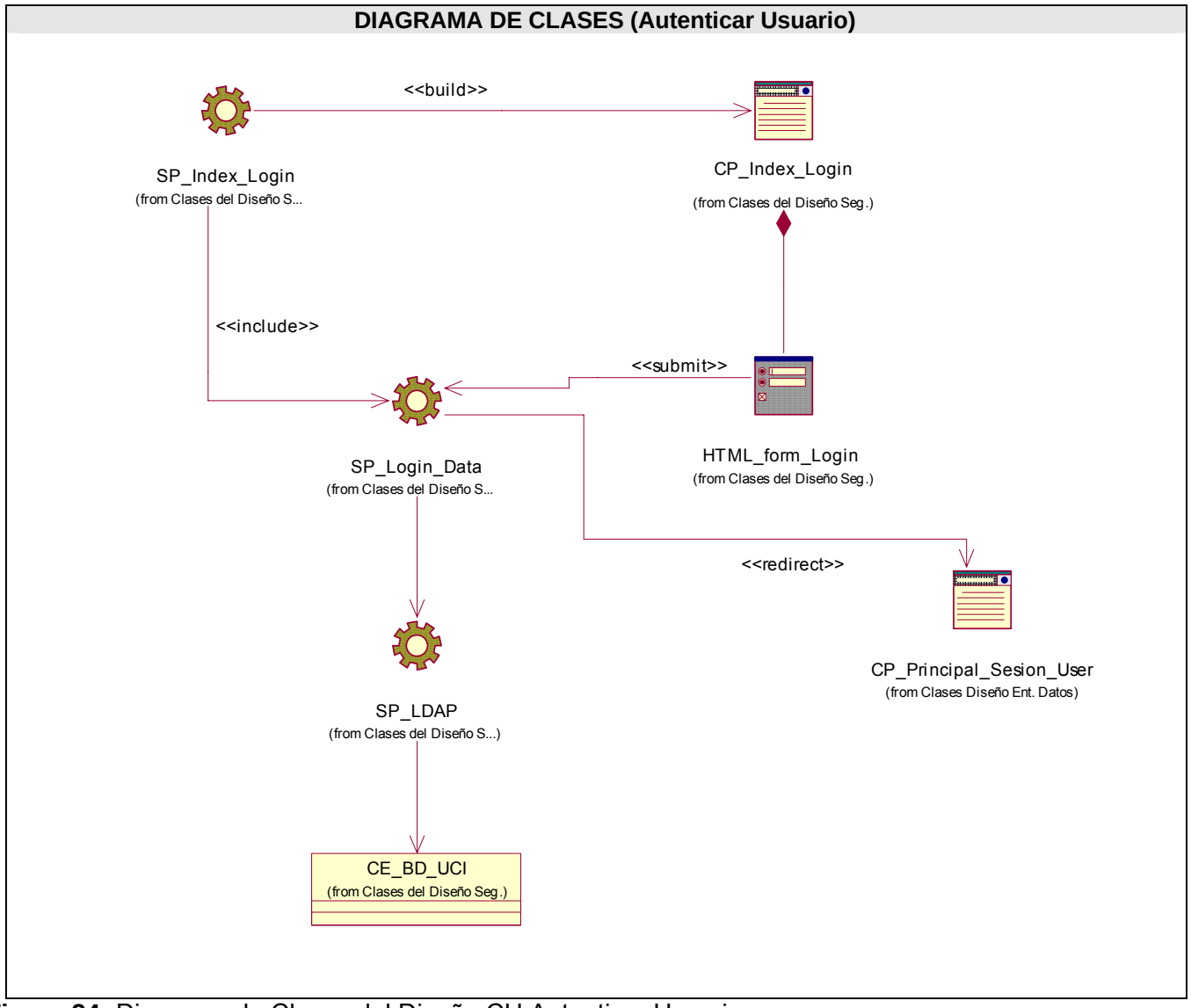

Figura 24: Diagrama de Clases del Diseño CU Autenticar Usuario.

CU Gestionar Investigador.

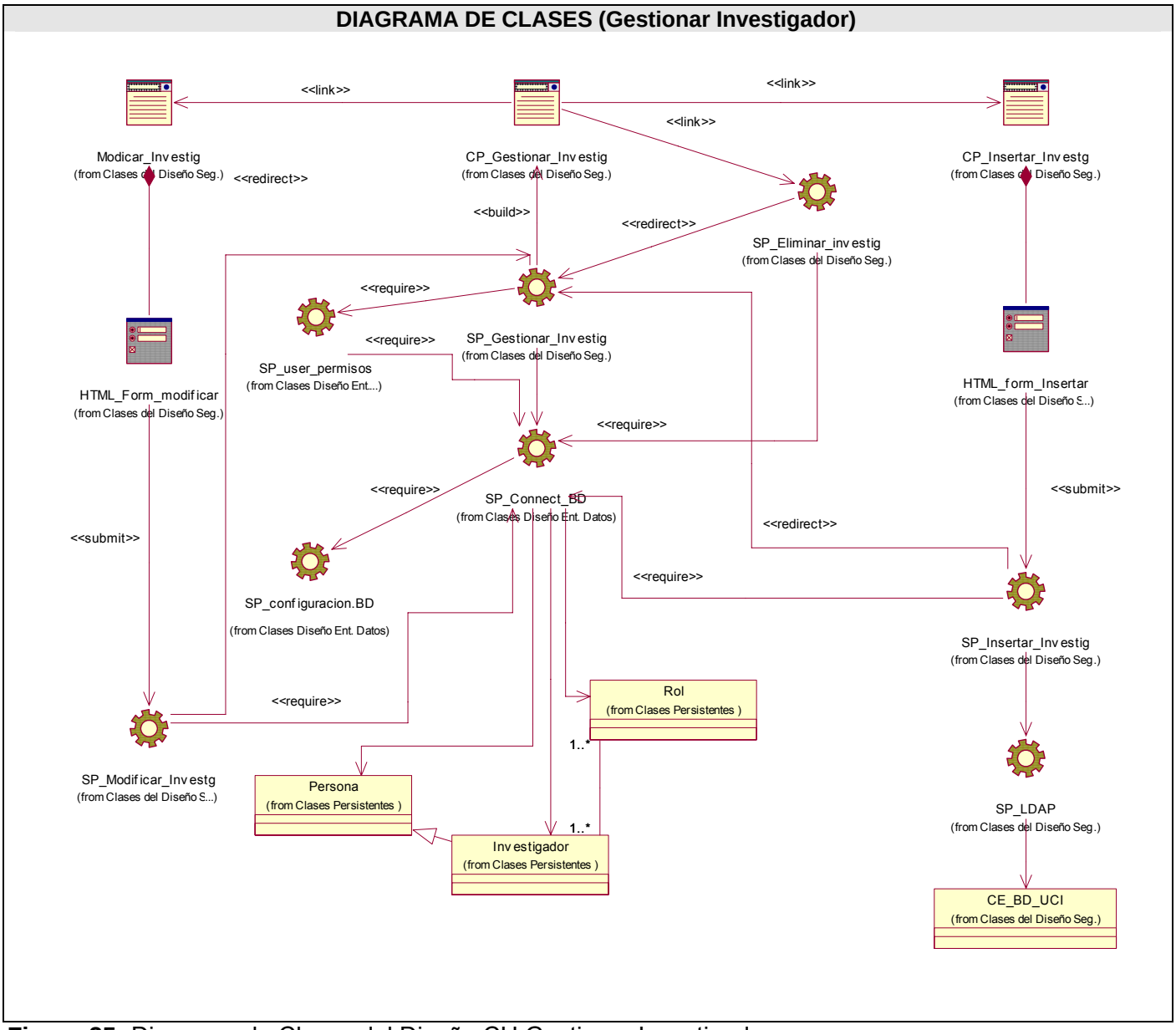

Figura 25: Diagrama de Clases del Diseño CU Gestionar Investigador.

CU Insertar Noticias.

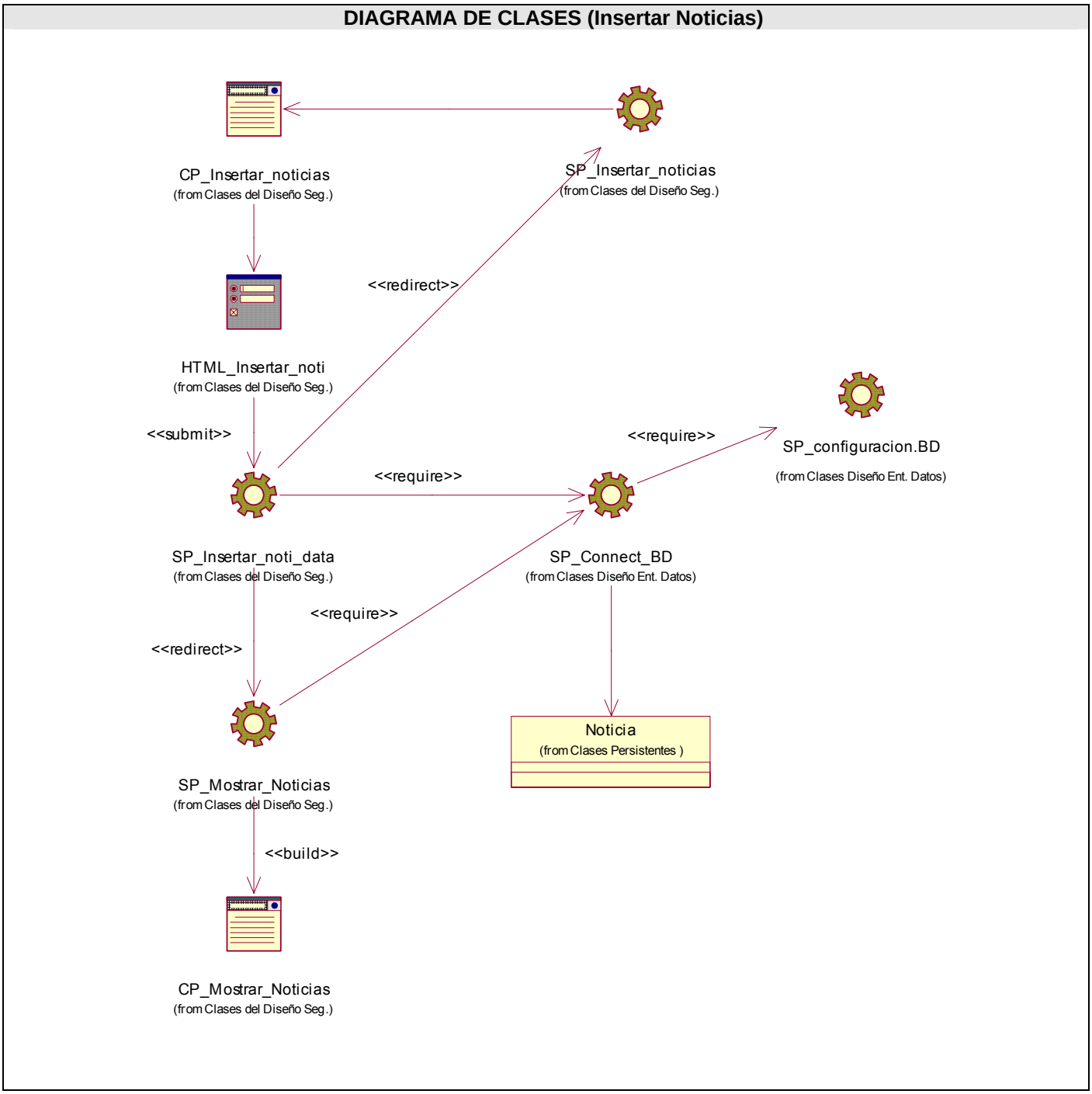

Figura 26: Diagrama de Clases del Diseño CU Insertar Noticias.

3.3.2 Diseño de la Base de Datos.

3.3.2.1 Modelo Lógico de Datos.

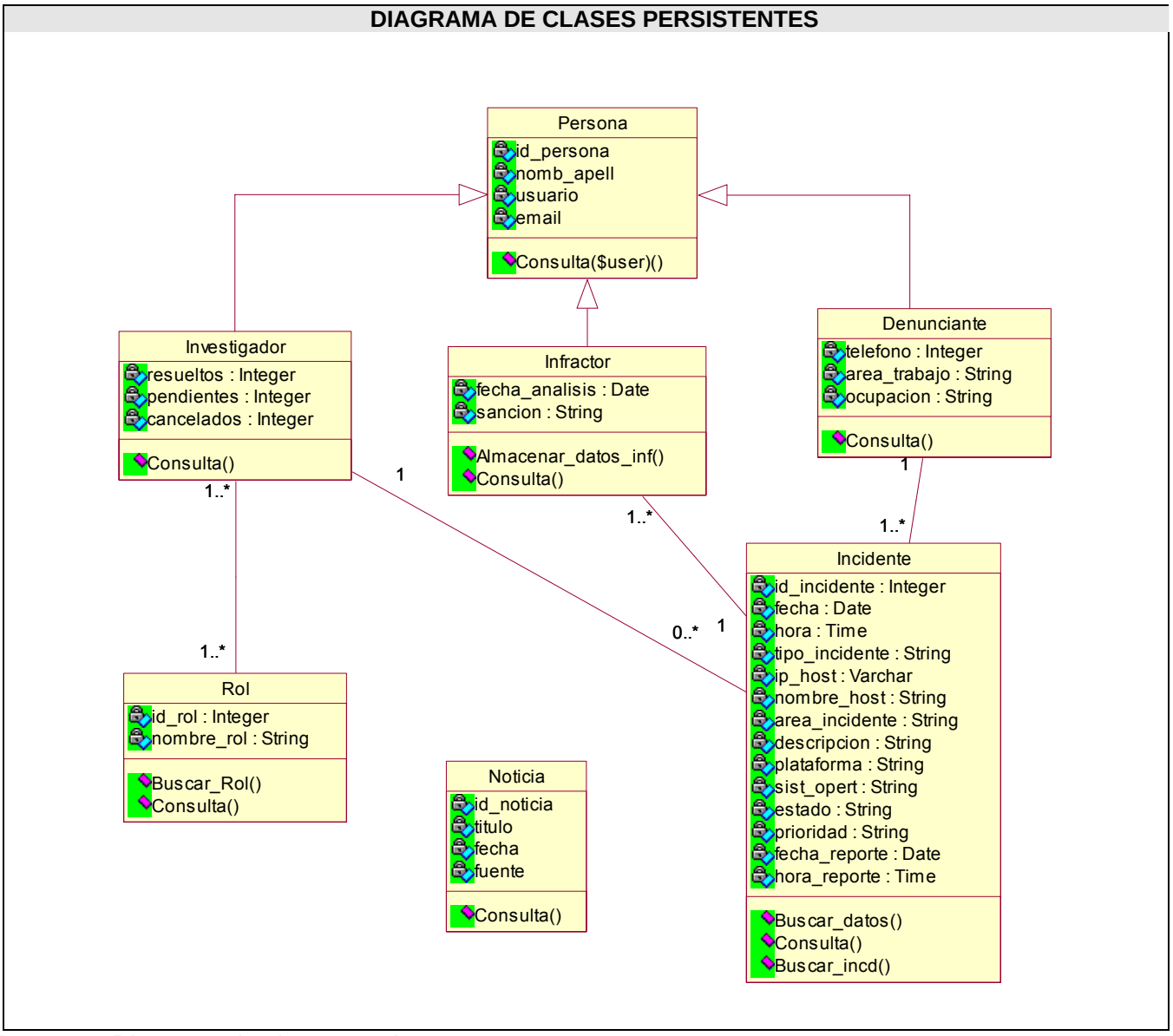

Figura 27: Diagrama de clases persistentes.

3.3.2.2 Modelo Físico de Datos.

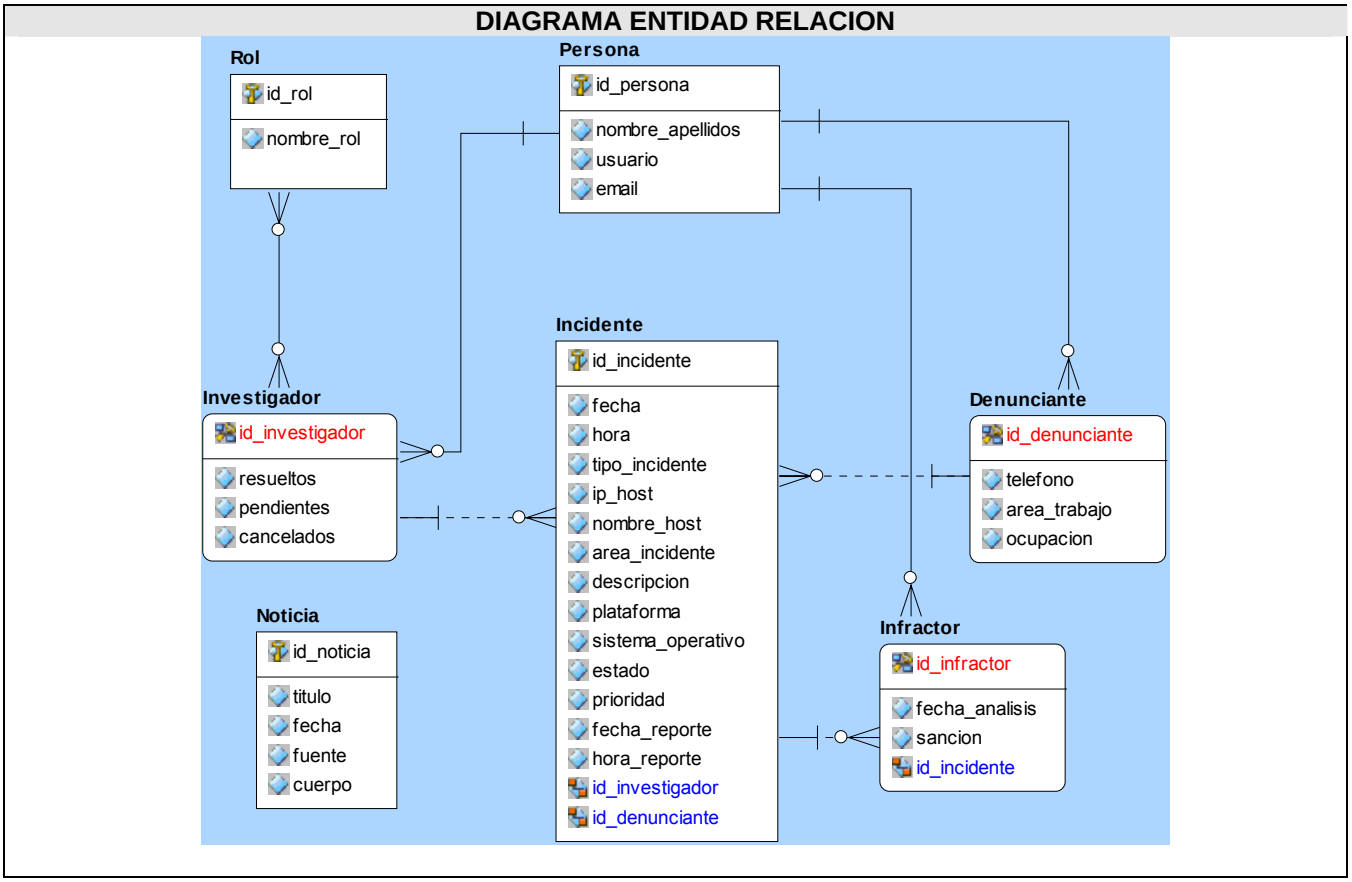

Figura 28: Diagrama Entidad Relación.

MODELO DE DATOS

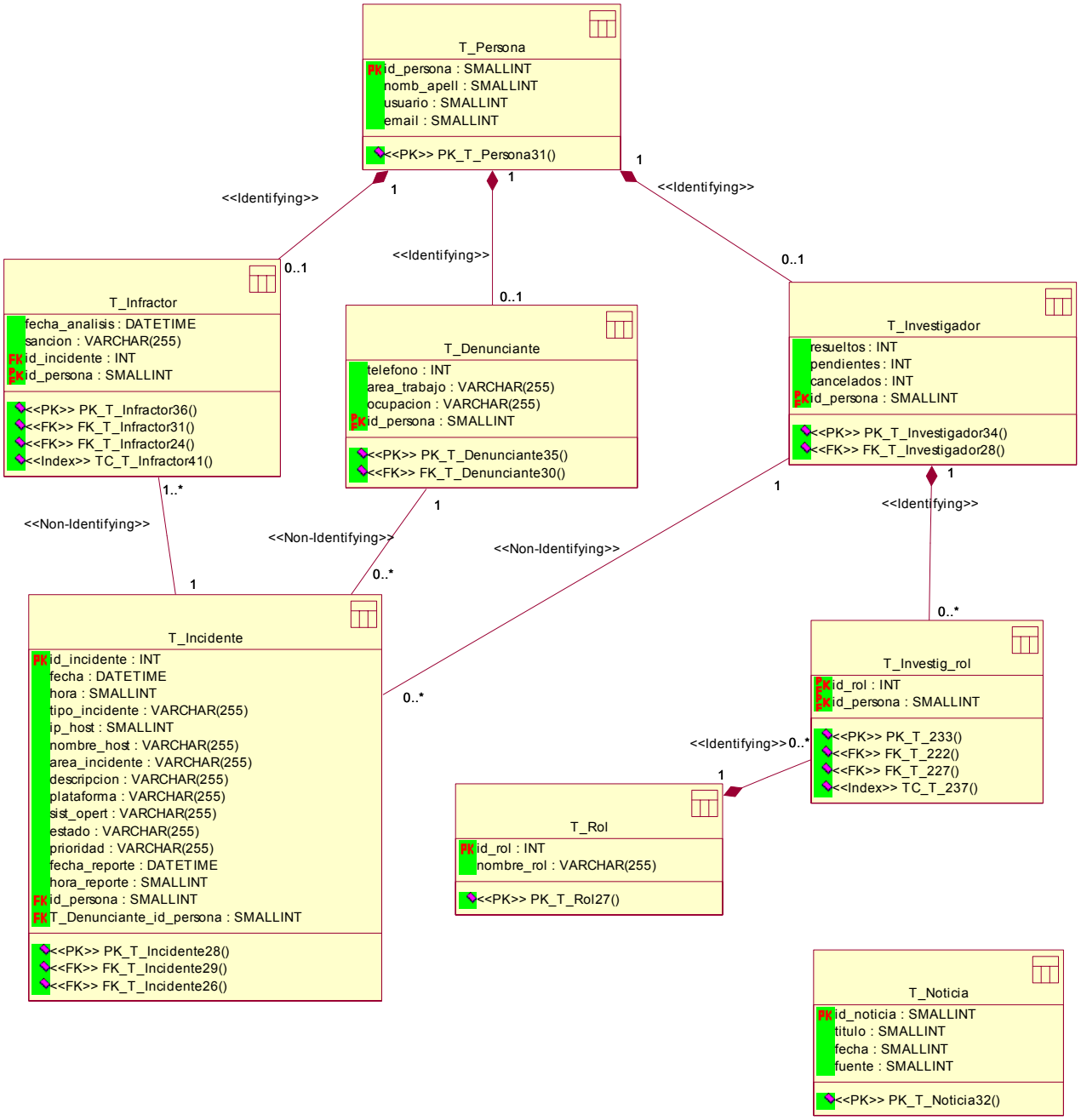

Figura 29: Modelo de Datos.

3.4 Definiciones de diseño.

El diseño de la interfaz de una aplicación, el tratamiento de los errores así como el seguimiento de la seguridad de esta tienen una gran importancia en el desarrollo y en el éxito de una aplicación. A continuación se describen los principios de diseño seguidos para el desarrollo de la aplicación en cuestión.

3.4.1 Tratamiento de errores.

El tratamiento de errores tiene gran importancia en la aplicación, de él depende el buen funcionamiento de la misma ya que se evita la inserción de datos erróneos.

Los errores generalmente se muestran al usuario en forma de mensajes generados por funciones JavaScript. Se ejecutan a nivel de cliente. Los mensajes que se emitirán serán breves pero informativos, explicando en qué consiste el error y si es necesaria la forma de resolverlos.

3.4.2 Seguridad.

La seguridad es muy necesaria en el sistema ya que la información que se procesa en el mismo es confidencial. Con el fin de garantizarla y de que cada usuario acceda solo a los datos a los que se les permita, se definen varios niveles de acceso a la información para los usuarios. Un primer nivel o nivel básico donde están las funciones asociadas al usuario general o común, que requieren poca responsabilidad. El segundo nivel está compuesto por funciones de mayor complejidad y que pueden destruir información relacionada a las entidades del sistema, a este nivel pertenecen los investigadores. El tercer nivel está conformado con las funciones administrativas del sitio y del sistema. Se debe identificar al usuario antes de que pueda realizar cualquier acción sobre el sistema. Se usan mecanismos de encriptación de los datos que por cuestiones de seguridad no deben viajar al servidor en texto claro, como es el caso de las contraseñas.

3.4.3 Interfaz.

La Interfaz de Usuario (IU) de un programa es un conjunto de elementos hardware y software de una computadora que presentan información al usuario y le permiten interactuar con la información y con el computadora. También se puede considerar parte de la IU la documentación (manuales, ayuda, referencia, tutoriales) que acompaña al hardware y al software.

Si la IU está bien diseñada, el usuario encontrará la respuesta que espera a su acción.

Los programas son usados por usuarios con distintos niveles de conocimientos, desde principiantes hasta expertos. Es por ello que no existe una interfaz válida para todos los usuarios y todas las tareas. Debe

permitirse libertad al usuario para que elija el modo de interacción que más se adecué a sus objetivos en cada momento. La mayoría de los programas y sistemas operativos ofrecen varias formas de interacción al usuario.

Para lograr esto, en el desarrollo de la interfaz, se han tenido en cuenta los siguientes aspectos:

- El sitio será diseñado para una resolución de pantalla de 1024x768, la cual será la resolución óptima, pero además se ajusta en gran medida a la resolución a que esté configurada la computadora del cliente.
- Se tendrá en cuenta la uniformidad del sitio utilizando plantillas y páginas de estilo que permitirán presentar de la misma manera la presentación de los contenidos. Entre estos estándares se tiene por ejemplo; el uso de la letra Verdana tamaño 10, el color predominante será el azul el cual es muy agradable y refrescante a la vista, el idioma que se utiliza es el español.
- La facilidad del usuario de poder navegar desde cualquier punto a otro dentro de la aplicación, esta dado por la utilización de un menú en la parte izquierda el cual tendrá vínculos a todas las páginas a que puede acceder el usuario de acuerdo a los privilegios del mismo.
- Se hará un buen aprovechamiento y optimización del espacio libre.
- Se tratará de minimizar la utilización de imágenes en los formularios para no sobrecargar el sitio y disminuir el tiempo de espera.

3.5 Conclusiones.

En este capítulo se realizaron los modelos del análisis y el diseño para cada uno de los casos de uso del sistema, así como también el modelo de datos. Se describieron los principios de diseño utilizados como por ejemplo los estándares de interfaz, tratamiento de errores y la forma en que se tratara la seguridad en la aplicación Web.

Capítulo 4. Implementación.

4.1 Introducción.

El presente capítulo tiene como objetivo realizar los diagramas correspondientes a implementación y despliegue que conforman lo que se conoce como Modelo de Implementación. En la implementación se inicia con el resultado del diseño y se implementa el sistema en términos de componentes organizando a estos de acuerdo a los nodos específicos en el modelo de despliegue.

4.2 Modelo de Implementación.

4.2.1 Diagrama de Despliegue.

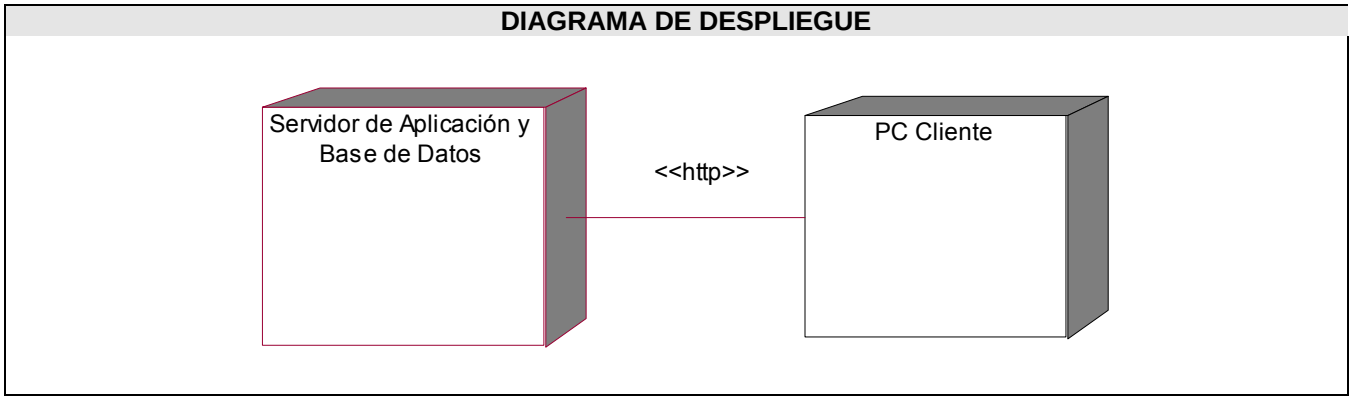

Figure 30: Diagrama de Despliegue.

4.2.2 Diagramas de Componentes.

El diagrama de componentes muestra las dependencias lógicas entre componentes software, sean estos componentes fuentes, binarios o ejecutables. Los componentes software tienen tipo, que indica si son útiles en tiempo de compilación, enlace o ejecución.

4.2.2.1 Diagrama de componentes Base de Datos.

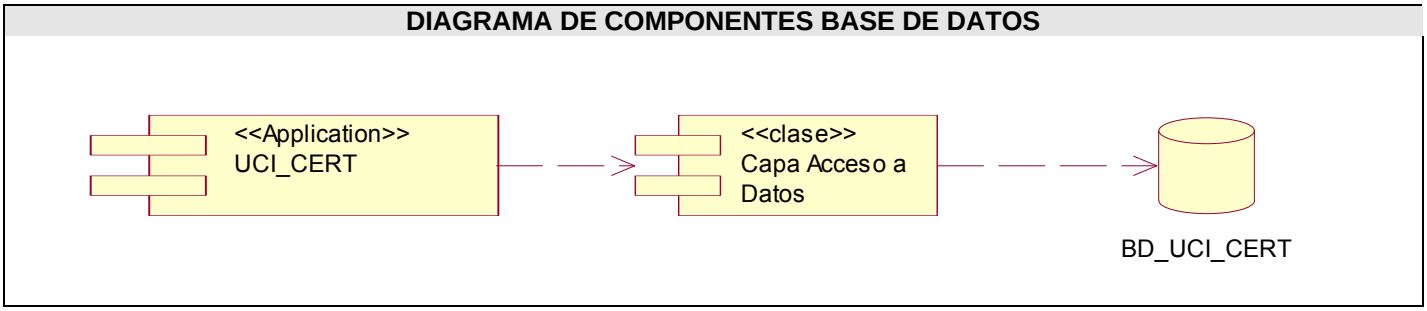

Figura 31: Diagrama de componentes Base de Datos.

4.2.2.2 Diagrama de componentes Código Fuente.

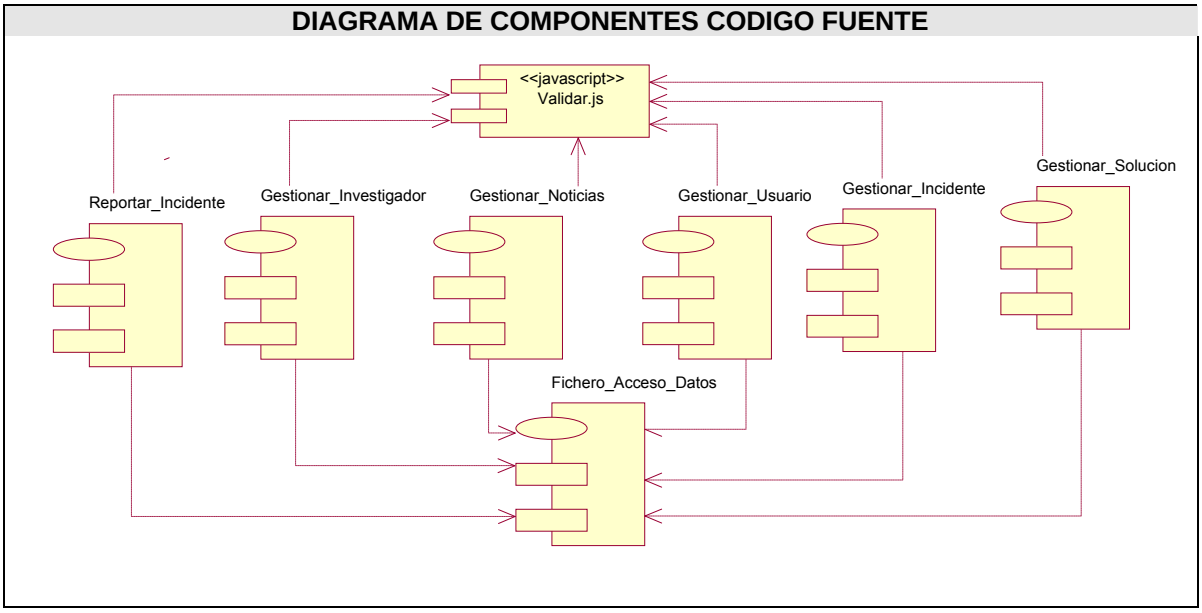

Figura 32: Diagrama de componentes Código Fuente.

4.2.2.3 Diagrama de componentes Código Ejecutable.

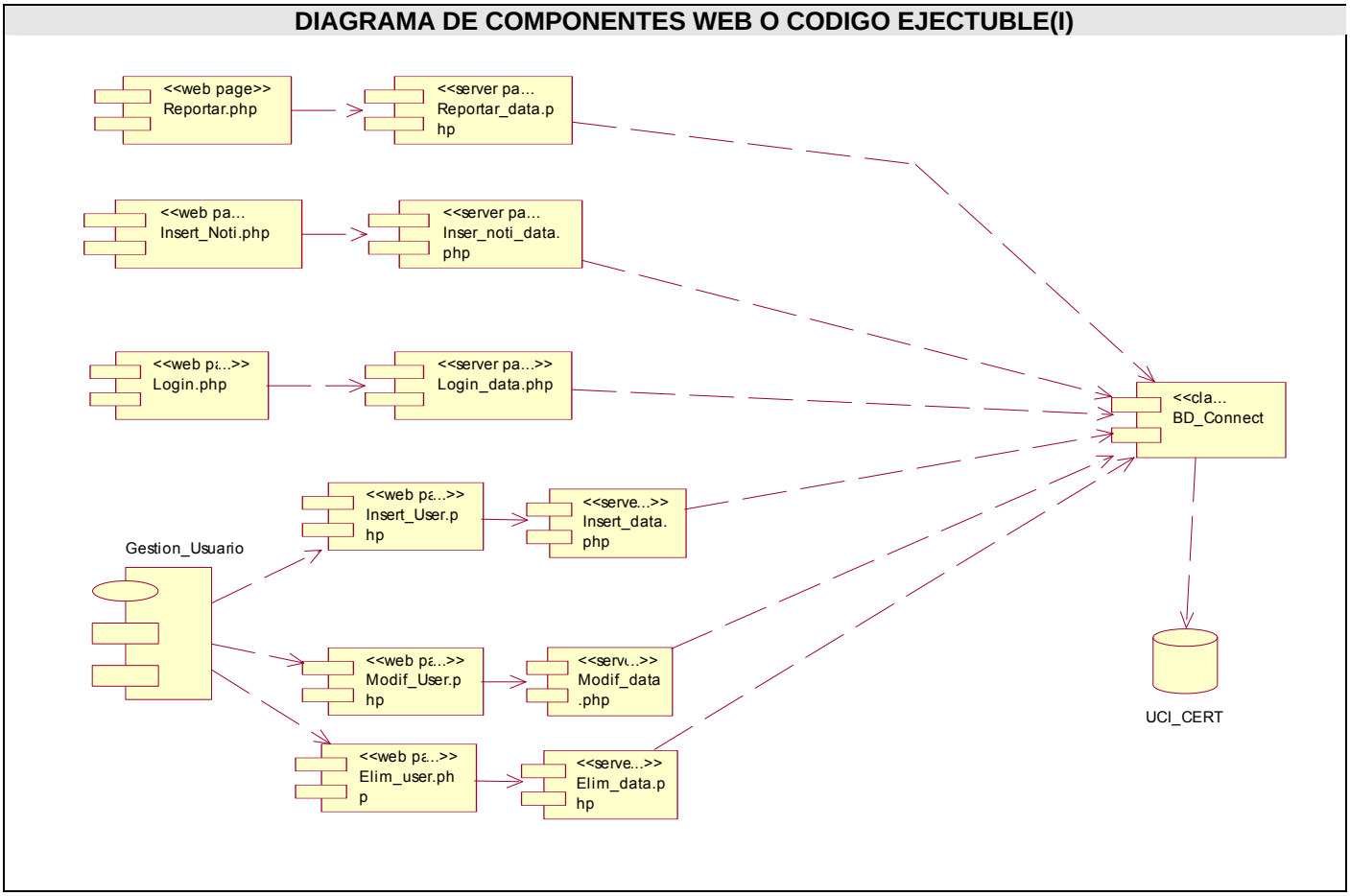

Figura 33: Diagrama de componentes Web o Código Ejecutable I.

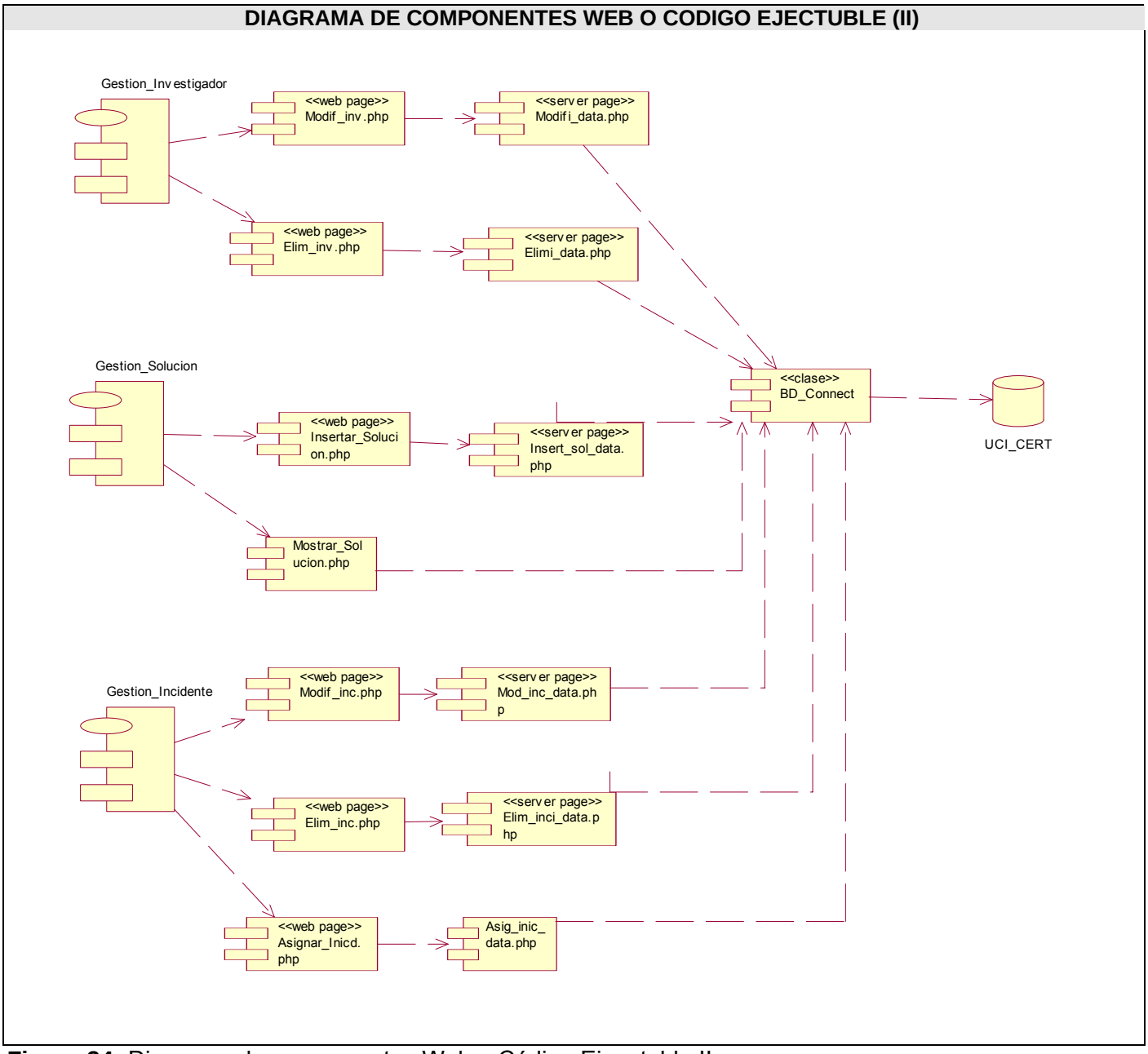

Figura 34: Diagrama de componentes Web o Código Ejecutable II.

4.3 Conclusiones.

En este capitulo se desarrolló el Modelo de Implementación en el cual se describieron los componentes a construir y su organización y dependencia entre nodos físicos en los que funcionará la aplicación.

Capítulo 5. Análisis de factibilidad.

5.1 Introducción.

Para la realización de un proyecto es de suma importancia el análisis del costo y los beneficios que reportará. Como resultado de este análisis se obtiene el tiempo de desarrollo en meses, costo y la cantidad de personas que se necesitan para desarrollar el proyecto. El objetivo de la planificación y el análisis de la factibilidad es instaurar planes razonables para desarrollar la Ingeniería de Software y operar los cambios de los proyectos de Software incluyendo la actividad de estimar los resultados del proyecto, el costo y esfuerzo. La planificación se logra mediante un procesamiento de la información que lleve a estimaciones razonables.

En este capítulo se describe la estimación de costos del sistema propuesto y sus beneficios.

5.2 Planificación basada en casos de uso.

Paso 1. Identificar los Puntos de casos de uso Desajustados.

El primer paso para la estimación es el cálculo de los puntos de casos de uso sin ajustar, este valor se obtiene aplicando la siguiente formula matemática:

UUCP=UAW+UUCW

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.

Para calcular el peso de los actores sin ajustar se hace un análisis de la cantidad de actores presentes en el sistema y la complejidad de cada uno de ellos como se muestra en la siguiente tabla:

Tipo de actor	Descripción	Factor de peso	Actores	Total
Simple	Sistema con sistema a través de interfaz de programación.	1	0	0
Medio	Sistema con sistema mediante protocolo de	2	0	0

	interfaz basada en texto.			
Complejo	Persona que interactúa con el sistema mediante interfaz gráfica.	3	3	9

UAW = S (Factor * Actores)

UAW = 9

Tipo de CU	Descripción	Peso	Cantidad de CU	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	6	30
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	3	30
Complejo	El caso de uso tiene más de 8 transacciones.	15	0	0

UUCW = Sumatoria (Factor * Cant. CU)

UUCW = 60

UUCP = UAW + UUCW

UUCP = 9 + 60

UUCP= 69

Paso 2. Cálculo de los Puntos de casos de uso ajustados.

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 de ambiente.

El factor de complejidad técnica (TCF) se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada factor se cuantifica en un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).

Factor	Descripción	Peso	Valor asignado	Total
T1	Sistema distribuido	2	3	6
T2	Tiempo de respuesta	1	4	4
T3	Eficiencia del usuario final	1	1	1
T4	Funcionamiento Interno complejo	1	2	2
T5	El código debe ser reutilizable	1	3	3
T6	Facilidad de instalación	0.5	1	0.5
T7	Facilidad de uso	0.5	2	1.5
T8	Portabilidad	2	2	4
T9	Facilidad de cambio	1	2	3
T10	Concurrencia	1	3	3
T11	Incluye objetivos especiales de seguridad	1	3	3
T12	Provee acceso directo a terceras partes	1	2	2
T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	1	1
			Sumatoria	34

TCF = 0.6 + 0.01 * Sumatoria (Peso * Valor)

TCF = 0.6 + 0.01 * 34

TCF = 0.94

T1: El sistema es distribuido ya que se puede acceder desde cualquier PC cliente, en la UCI.

T2: El tiempo de respuesta de be ser alto debido a que se conectarán simultáneamente varios usuarios.

T3: Al sistema podrá acceder cualquier usuario del dominio UCI.

T4: El sistema utiliza para realizar las búsquedas consultas a bases de datos.

T5: El código fuente puede ser reutilizable para otras implementaciones.

T6: El sistemas requiere un Servidor web Apache, con MySQL como gestor de bases de datos, PHP como lenguaje de programación.
T7: El sistema tendrá una interfaz amigable y fácil de usar.
T8: Será una aplicación con una gran capacidad para ser ejecutada en diferentes sistemas informáticos.
T9: Estará disponible a realizarle cambios necesarios.
T10: El sistema presenta una concurrencia normal.
T11: Los usuarios tendrán determinados accesos y permisos en el sistema.
T12: El sistema brindará acceso a los diferentes módulos.
T13: Su fácil uso permite que el personal no tenga que alcanzar un alto nivel de capacitación.

El factor de ambiente (EF) está relacionado con las habilidades y entrenamiento del grupo de desarrollo que realiza el sistema. Cada factor se cuantifica con un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).

Factor	Descripción	Peso	Valor asignado	Total
E1	Familiaridad con el modelo de proyecto utilizado	1.5	4	6
E2	Experiencia en la aplicación	0.5	3	1.5
E3	Experiencia en la orientación a objetivos.	1	2	4
E4	Capacidad del analista líder.	0.5	4	2
E5	Motivación.	1	5	5
E6	Estabilidad de requerimientos	2	4	8
E7	Personal Part–Time	-1	4	-4
E8	Dificultad del lenguaje de programación	-1	3	-3
			Sumatoria	19.5

EF = 1.4 - 0.03 * Sumatoria (Peso * Valor)
EF = 1.4 – 0.03 * 19.5
EF = 0.815

UCP = UUCP * TCF * EF

UCP= 69 * 0.94 * 0.815

UCP = 52.86

Paso 3. Estimación de esfuerzo a través de los puntos de casos de uso.

E = UCP * CF

Donde:

E: Esfuerzo estimado en horas hombres.

UCP: Punto de casos de usos ajustados.

CF: Factor de conversión.

Para obtener el factor de conversión (CF) se cuentan cuantos valores de los que afectan el factor ambiente (E1...E6) están por debajo de la media (3), y los que están por arriba de la media para los restantes (E7, E8). Si el total es 2 o menos se utiliza el factor de conversión 20 Horas-Hombre / Punto de Casos de uso. Si el total es 3 o 4 se utiliza el factor de conversión 28 Horas-Hombre / Punto de Casos de uso. 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.

En este caso se puede decir que:

CF = 20 Horas-Hombre / Punto de Casos de uso.

E = 52.86 * 20

E = 1056.0

Paso 4. Calcular esfuerzo de todo el proyecto.

Actividad	Porcentaje %	Horas-Hombres
Análisis	10	264.0
Diseño	20	528.0
Implementación	40	1056.0
Pruebas	15	396.0
Sobrecarga (otras actividades)	15	396.0

Total	100	**2640**

El valor de esfuerzo calculado anteriormente representa el esfuerzo del Flujo de Trabajo implementación, por comparación salen el resto de los esfuerzo y la suma de ellos es el esfuerzo total (ET). Para la realización del proyecto se trabaja diario 8 horas, 6 días a la semana y un mes tiene como promedio 4 semanas se trabajan 24 días; la cantidad de horas que puede trabajar una persona mensual es de 192 horas.

Si *ET = 2640 horas-hombre* y por cada 192 horas se tiene 1 mes eso daría un *ET = 13.75 mes-hombre.* Esto significa que 1 persona puede realizar el problema analizado en un margen de 13 meses aproximadamente.

-Costo del Proyecto.

Se asume como salario promedio mensual $50.00

CH: Cantidad de hombres.

Tiempo: Tiempo total del proyecto.

CH = 2 hombres

CHM = 2 * Salario Promedio

CHM = 100.00 $/mes

Costo = CHM * ET / CH

Costo = 100.00 *13.75 / 2

Costo = $ 687.5≈ $688.00

Tiempo = ET / CH

Tiempo = 13.75 / 2

Tiempo = 6.875 ≈ 7 meses.

De los resultados obtenidos se interpreta que con 2 hombres trabajando en el proyecto el mismo se desarrolla en 7 meses y su costo total se estima que sea $688.00.

5.3 Beneficios tangibles e intangibles.

Los beneficios que se obtendrán con el desarrollo de este software serán fundamentalmente intangibles, ya que permite mantener el control más detallado y organizado sobre los incidentes de seguridad,

Facilitando así el trabajo a los investigadores. También implica un mayor aprovechamiento por parte de los investigadores ya que se puede medir el rendimiento de los investigadores.
Al desarrollo de todo producto informático va asociado un costo, el justificarlo depende de los beneficios tangibles e intangibles que produce.

5.4 Análisis de costos y beneficios.

La utilización del sistema de gestión desarrollado permitirá gestionar los incidentes informáticos que ocurran en la universidad y tener un control sobre los mismos de forma organizada. Este nuevo recurso del que dispondrá este centro le permitirá un mayor cumplimiento de las actividades planificadas. Además mejora las condiciones de trabajo de los investigadores que son los que trabajan de forma directa para solucionar los incidentes informáticos.

5.5 Conclusiones.

Después del estudio realizado se puede concluir que se han obtenido datos satisfactorios en relación con la cantidad de información disponible. La estimación por Puntos de casos de usos resulta muy efectiva para estimar el esfuerzo requerido en el desarrollo de los primeros casos de uso de un sistema. Además de que existe un balance apropiado costo-beneficio.

Conclusiones.

Con la realización de este trabajo se propuso una solución al problema de la gestión de los reportes de incidentes informáticos en la UCI. Se hicieron investigaciones y comparaciones con otros sistemas existentes en el mundo que gestionan incidentes de seguridad los cuales sirvieron como punto de partida y guía para comenzar a desarrollar la aplicación.

El desarrollo de este software dio cumplimiento al objetivo trazado: *Desarrollar un Sistema que sea capaz de realizar la Gestión de los Reportes de Incidentes Informáticos en la UCI,* debido a que los requerimientos soportan al sistema y los casos de uso satisfacen las necesidades funcionales.

Se realizó el Análisis y Diseño de una aplicación capaz de gestionar los incidentes de seguridad y brindar otras facilidades a los investigadores, lo cual sirvió de apoyo para la parte de implementación. .

Se construyó una base de datos, donde se almacena toda la información necesaria de los incidentes, para de esta forma garantizar la veracidad de la misma.

El sistema resultante está provisto de un ambiente cómodo, fácil de entender, que cumple los estándares de diseño y utiliza técnicas modernas de programación orientada a objetos.

El análisis de factibilidad realizado arrojó resultados satisfactorios en cuanto a costos y beneficios.

Con la realización de este trabajo se obtuvieron una serie de beneficios que se mencionan a continuación:

- Se obtiene la propuesta de una aplicación que elimina el trabajo de forma manual e insuficiente.
- Permitir un menor tiempo de respuesta ante una solicitud y una mayor confiabilidad en la información obtenida.
- Facilitar y organizar el trabajo de los investigadores.
- Verificar el rendimiento de los investigadores.
- Lograr una seguridad y protección de los datos consecuente con el nivel de seguridad requerido.
- Minimizar los costos por concepto de confiabilidad y agilidad en el manejo de datos relativos a la toma de decisiones.
- El valor social del sistema se expresa en la contribución a mejorar las condiciones de trabajo, desempeño y equidad de los especialistas del área.

Luego de todo este proceso de trabajo se puede concluir que UCI_CERT es un sistema que da solución a la situación problémica que lo originó, que cumple los objetivos que se trazaron para su creación y que su utilización significará una mejora considerable en la calidad y eficiencia de los procesos que automatiza.

Recomendaciones.

Se recomienda:

1. Poner a prueba el sistema durante un período de tiempo significativo, para comprobar su desempeño y que las funcionalidades del sistema se correspondan con la realidad de lo que se desea.
2. Continuar el estudio con el objetivo de añadir nuevas funcionalidades al sistema.

Referencias Bibliográficas.

[1] Merce Molist, Los Rescatadores, 18/04/2001, [Disponible en: http://ww2.grn.es/merce/2001/certs.html]

[2] Sitio oficial en español de las Organización de Naciones Unidas, http://www.un.org/spanish/, 2006-2007.

[3] CARRION, Hugo Daniel. Tesis "Presupuestos para la Punibilidad del Hacking", 2006, [Disponible en: www.delitosinformáticos.com/tesis.htm]

[4]Inarnet.net,18/05/2006, [Disponible en: http://inarnet.net]

[5] Centro de Alerta Temprana sobre Virus Informáticos, Marzo, 2002[Disponible en http://alerta-antivirus.red.es]

[6] ArCERT, 14/10/2004, [Disponible en: http://www.arcert.gov.ar/webs/csirt_faq.html]

[7] IRIS-CERT, [Disponible en: http://www.rediris.es/cert/]

[8] UNAM-CERT, [Disponible en: http://www.cert.org.mx/]

[9] CU-CERT, [Disponible en: http://www.cucert.co.cu/]

[10] José Guillermo Valle, Arquitectura Cliente-Servidor, 2005, [Disponible en http:// www.monografias.com/trabajos24/arquitectura-cliente-servidor]

[*] Apache Software Foundation, [Disponible en: http://www.apache.org]

[11] I. Jacobson, G. Booch, J.Rumbaugh, El Proceso Unificado de Desarrollo de Software, Madrid, Addison Wesley, 2000, 485.

[12] I. Jacobson, G. Booch, J.Rumbaugh, El Lenguaje Unificado de Modelado. Manual de Referencia, Madrid, Addison Wesley, 2000, 528.

Bibliografía.

- Sitio de Seguridad Informática, UCI, https://seguridad.uci.cu/.
- Wikipedia, la enciclopedia libre, http://es.wikipedia.org/.
- Sitio oficial en español de las Organización de Naciones Unidas, http://www.un.org/spanish/, 2006-2007.
- IRIS-CERT, http://www.rediris.es/cert/servicios/iris-cert/
- UNAM-CERT, http://www.cert.org.mx/.
- CU-CERT, http://www.cucert.co.cu/.
- http://www.arcert.gov.ar/
- http://www.delitosinformaticos.com
- http://www.iec.csic.es/CRIPTONOMICON/siep.html
- http://www.monografias.com/trabajos16/seguridad-informatica/seguridad-informatica.shtml
- http://www.elrinconcito.com/
- http://blog.hispasec.com/laboratorio/141
- http://escert.upc.edu/index.php/web/es/index.html
- http://www.tb-security.com/serv_seg_respuesta.htm
- http://www.vsantivirus.com/mm-aniversario-cert.htm
- http://www.e-ghost.deusto.es/docs/TutorialMySQL.html
- http://www.salnet.com.ar/inv_mysql/pag01_intro.htm
- http://www.webtaller.com/maletin/articulos/lenguajes-programacion-web.php
- http://marty.anstey.ca/programming/php/articles/PHPvsASP
- http://www.htmlpoint.com/perl/perl_02.htm
- http://www.tejedoresdelweb.com/307/article-5813.html
- http://ascii.eii.us.es/docs/2002-03/php/php4.html
- http://www.ultrasist.com.mx/tecnologias/asp.htm
- http://www.csi.map.es/csi/silice/Global76.html
- http://es.wikipedia.org/wiki/Arquitectura_de_tres_niveles
- El Proceso Unificado De Desarrollo De Software, I.Jacobson Ivar, Booch Grady, Rumbaugh James, Madrid, Addison Wesley.

- Lenguaje Unificado de Modelado. Manual de Referencia, I. Jacobson, G. Booch, J.Rumbaugh, El, Madrid, Addison Wesley.
- Aprendiendo UML en 24 horas, Joseph Schmuller, México, 2000, 448.
- Ingeniería del Software. Un enfoque práctico, Roger S. Presuman.
- Conferencias y Clases Prácticas de Ingeniería del Software I y II, Seguridad Informática, Curso 2006-2007, Disponibles en http://teleformacion.uci.cu/
- Cursos Optativos de PHP, MySQL, Disponibles en http://teleformacion.uci.cu/
- Manual de PHP, Colectivo de autores, 2005, http://www.php.net/docs.php.
- Desarrollo Web con PHP y MySQL, Luke Welling y Laura Thomson
- Conferencia “Flujo de trabajo de gestión de Proyectos”, de Ingeniería del Software I, Curso 2006-2007, Disponible en http://teleformacion.uci.cu/
- Peralta, Mario. Estimación del esfuerzo basada en casos de uso. Centro de Ingeniería del Software e Ingeniería del Conocimiento (CAPIS), Buenos Aires, Argentina.

Glosario de términos y siglas.

Por orden de aparición en el documento.

- RUP: Rational Unified Process, Proceso Unificado de Desarrollo de Software en español.
- UML: Lenguaje Unificado de Modelado, usado para modelar sistemas de software.
- CASE: Computer Aided Software Engineering. 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.
- PHP: Hypertext Pre-Processor. Es un ambiente script del lado del servidor que permite crear y ejecutar aplicaciones Web dinámicas e interactivas.
- MySQL: Es un sistema de gestión de bases de datos relacional que cuentan con todas las características de un motor de BD comercial: transacciones atómicas, triggers, replicación, llaves foráneas entre otras. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar.
- SQL: Structured Query Language. Es un lenguaje declarativo de acceso a bases de datos que permite especificar diversos tipos de operaciones sobre las mismas. Aúna características del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar información de interés de una base de datos.
- Arpanet: Advanced Research Projects Agency Network, red de computadoras creada por encargo del Departamento de Defensa de los Estados Unidos como medio de comunicación para los diferentes organismos del país. El primer nodo se creó en la Universidad de California y fue la espina dorsal de Internet hasta 1990, tras finalizar la transición al protocolo TCP/IP en 1983.
- UNIX: Sistema operativo portable, multitarea y multiusuario; desarrollado en principio por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McIlroy.
- CERT: Computer Emergency Response Team. Su objetivo es facilitar la respuesta a problemas de seguridad informática que afecten a los sistemas centrales de Internet.
- Troyanos: Se denomina troyano (o caballo de Troya, traducción más fiel del inglés Trojan horse aunque no tan utilizada) a un programa malicioso capaz de alojarse en computadoras y permitir el acceso a usuarios externos, a través de una red local o de Internet, con el fin de recabar información o controlar remotamente a la máquina anfitriona.

- Worms: Un gusano es un virus informático o programa auto replicante que no altera los archivos sino que reside en la memoria y se duplica a sí mismo.
- Root kits: Programas que son insertados en una computadora después de que algún atacante ha ganado el control de un sistema. Generalmente incluyen funciones para ocultar los rastros del ataque, como es borrar los log de entradas o encubrir los procesos del atacante. Pueden incluir puertas traseras, permitiendo al atacante obtener de nuevo acceso al sistema o también pueden incluir exploits para atacar otros sistemas.
- Ancho de banda: Cantidad de datos que se pueden transmitir en una unidad de tiempo por la red.
- Sniffers: Programa de captura de las tramas de red. Generalmente se usa para gestionar la red con una finalidad docente, aunque también puede ser utilizado con fines maliciosos.
- UNAM: Universidad Nacional Autónoma de México.
- HTML: HyperText Markup Language. Lenguaje usado para escribir documentos para servidores World Wide Web.
- IIS: Internet Information Services, es una serie de servicios para los ordenadores que funcionan con Windows.
- FTP: File Transfer Protocol, es un protocolo de transferencia de ficheros entre sistemas conectados a una red TCP basado en la arquitectura cliente-servidor, de manera que desde un equipo cliente nos podemos conectar a un servidor para descargar ficheros desde él o para enviarle nuestros propios archivos independientemente del sistema operativo utilizado en cada equipo.
- XML: Siglas en ingles de eXtensible Markup Language, es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.
- MD5: Acrónimo de Message-Digest Algorithm 5, Algoritmo de Resumen del Mensaje 5. Es un algoritmo de reducción criptográfico de 128 bits ampliamente usado.
- IMAP: Acrónimo inglés de Internet Message Access Protocol, es un protocolo de red de acceso a mensajes electrónicos almacenados en un servidor.
- SNMP: El Protocolo Simple de administración de red o SNMP es un protocolo de la capa de aplicación que facilita el intercambio de información de administración entre dispositivos de red.
- PDL: Public Documentation License, es una licencia de documentación pública que es utilizada (entre otros proyectos) por OpenOffice.org.

- ODBC: Open DataBase Connectivity, es un estándar de acceso a Bases de Datos desarrollado por Microsoft Corporation, su objetivo es hacer posible el acceder a cualquier dato de cualquier aplicación, sin importar qué Sistema Gestor de Bases de Datos los almacene.
- Investigador: Son aquellas personas encargadas de investigar los incidentes computacionales y darle una solución a los mismos.
- Incidente Computacional: Son todos aquellos sucesos resultados de acciones malintencionadas y que comprometen leve o gravemente la confidencialidad, disponibilidad y/o integridad de la información digital.

Anexos.

Anexo 1. Modelo del Negocio.

Diagrama de actividad CU Gestionar Incidente

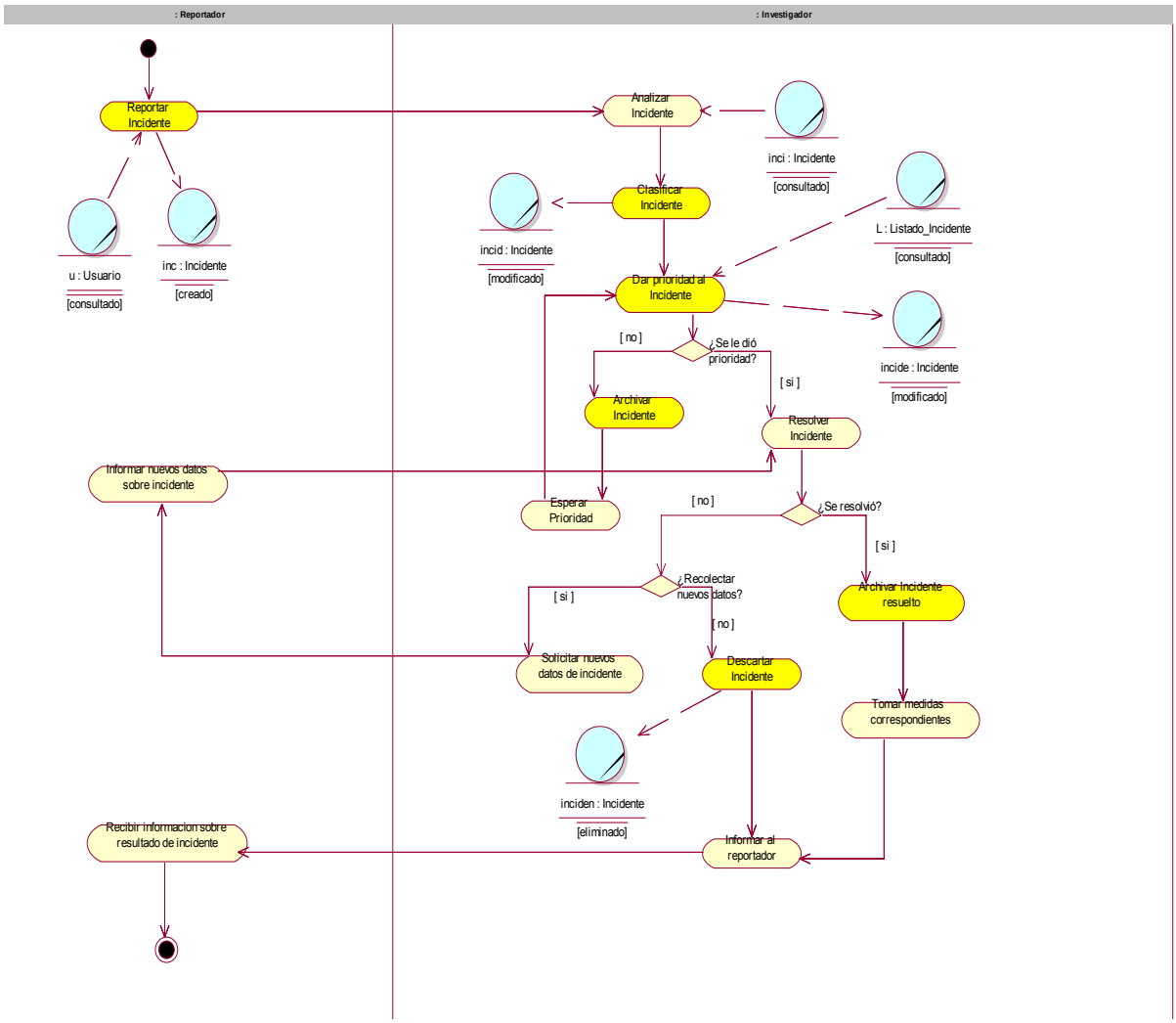

Figura 35: Diagrama de actividad CU Gestionar Incidente.

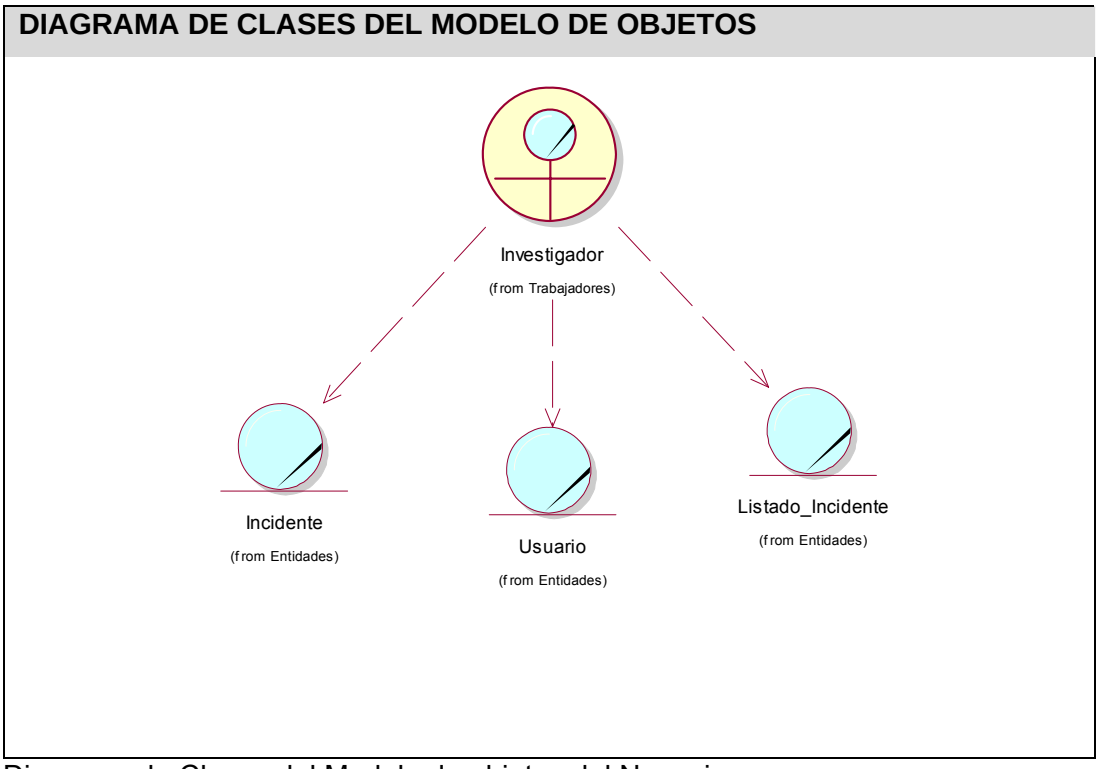

Figura 36: Diagrama de Clases del Modelo de objetos del Negocio.

Anexo 2. Diagramas de Interacción del diseño.

CU Gestionar Incidente.

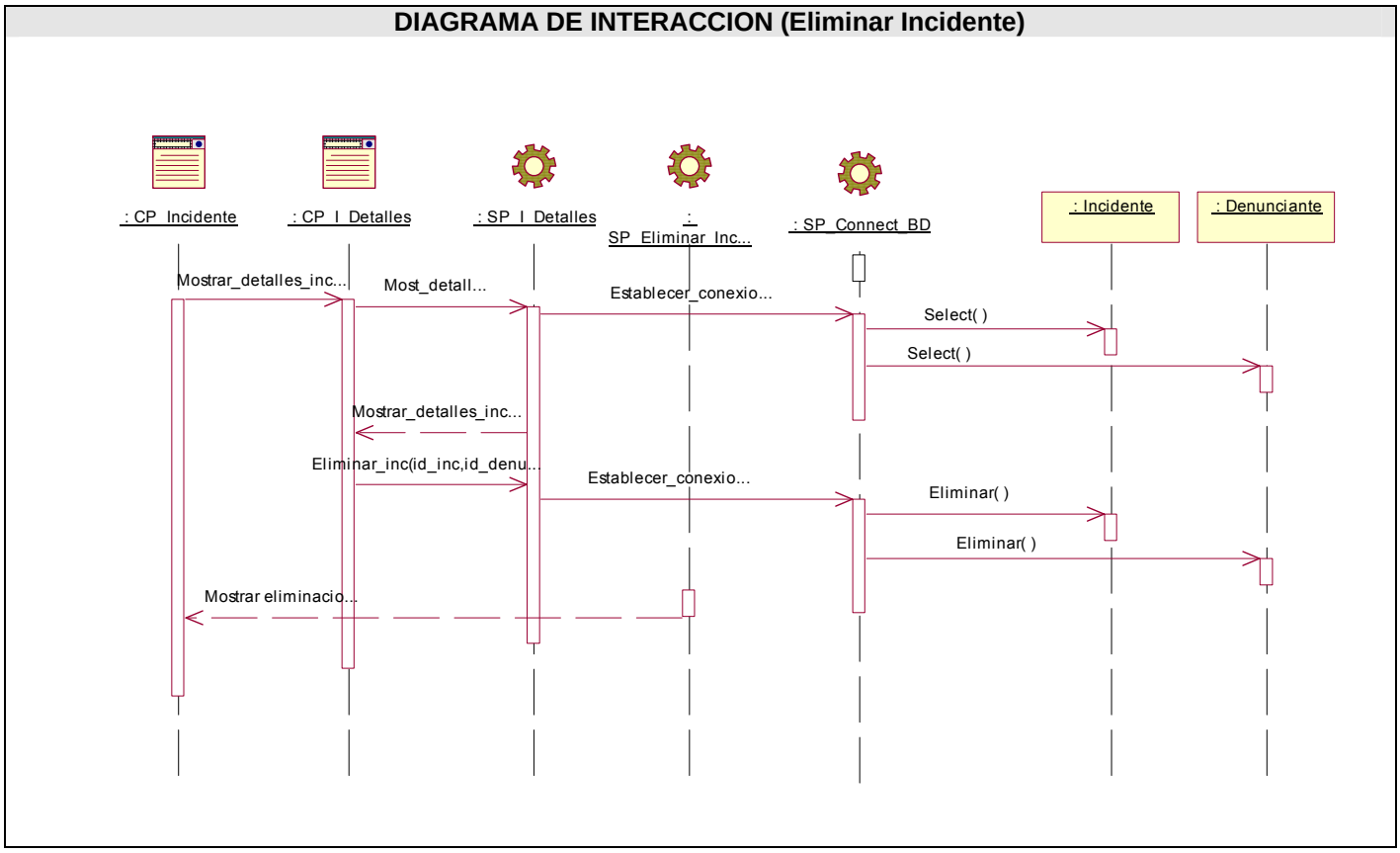

Figura 37: Diagrama de Interacción del Diseño. Escenario Eliminar Incidente.

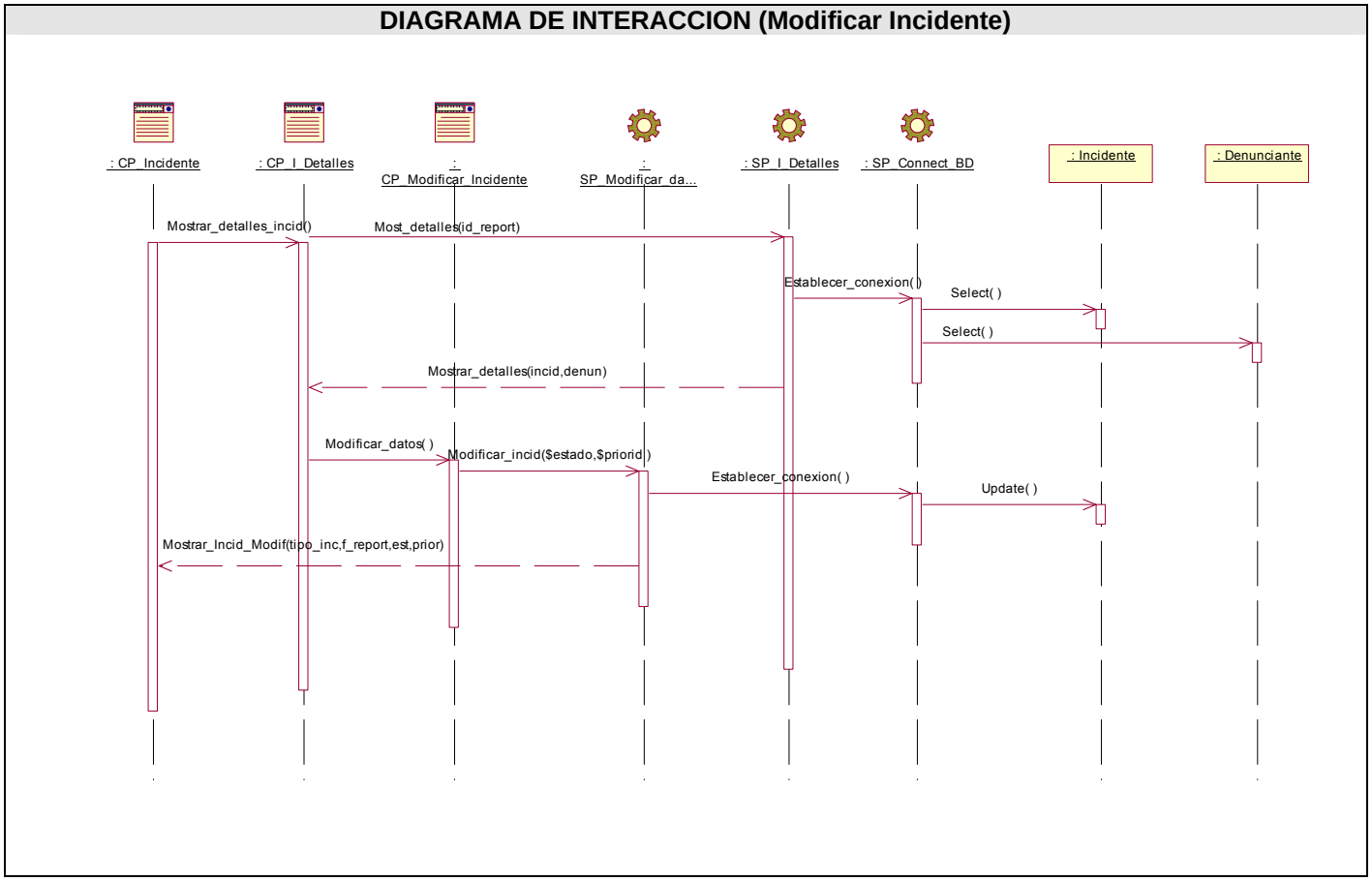

Figura 38: Diagrama de Interacción del Diseño. Escenario Modificar Incidente.

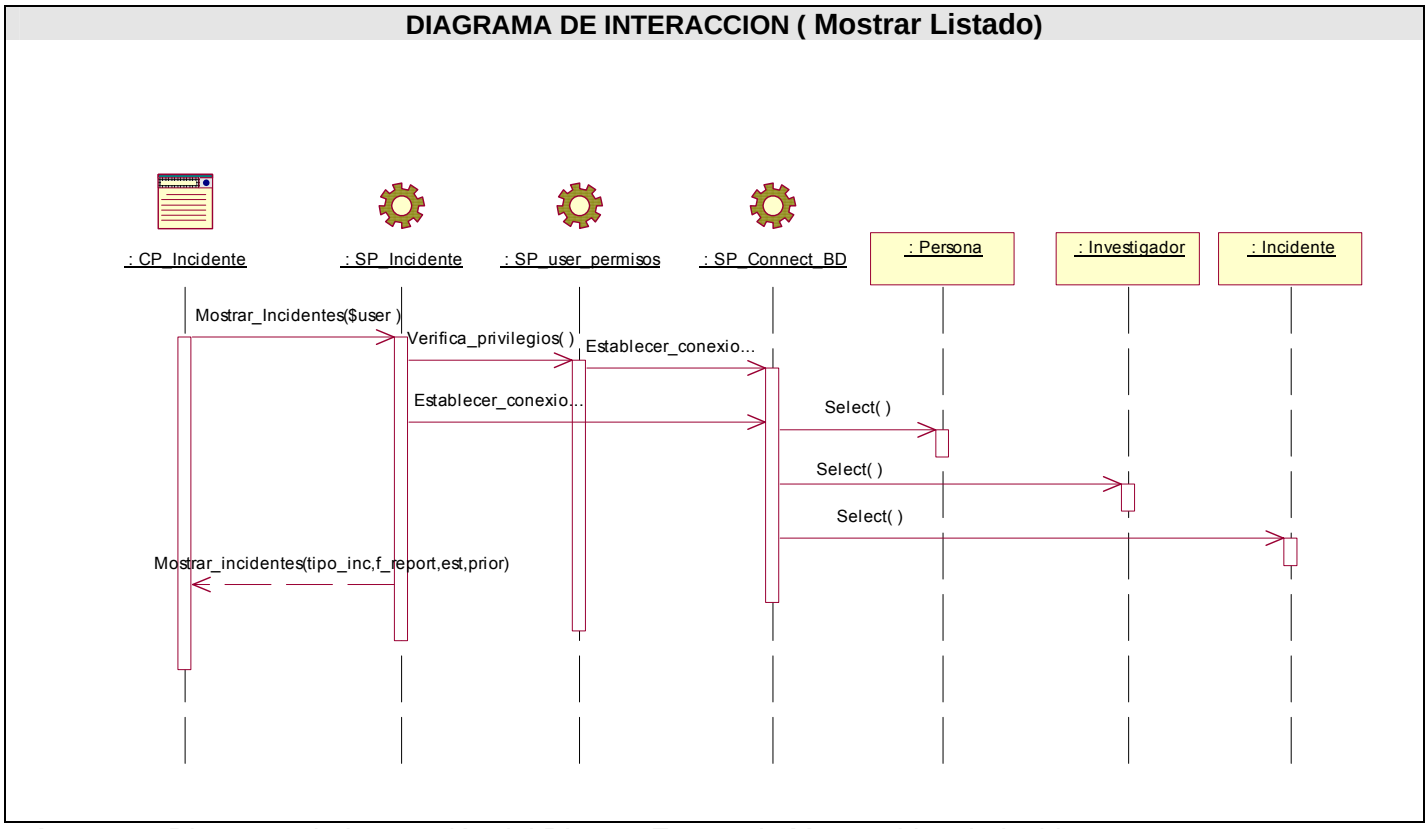

Figura 39: Diagrama de Interacción del Diseño. Escenario Mostrar Listado Incidentes.

CU Reportar incidente.

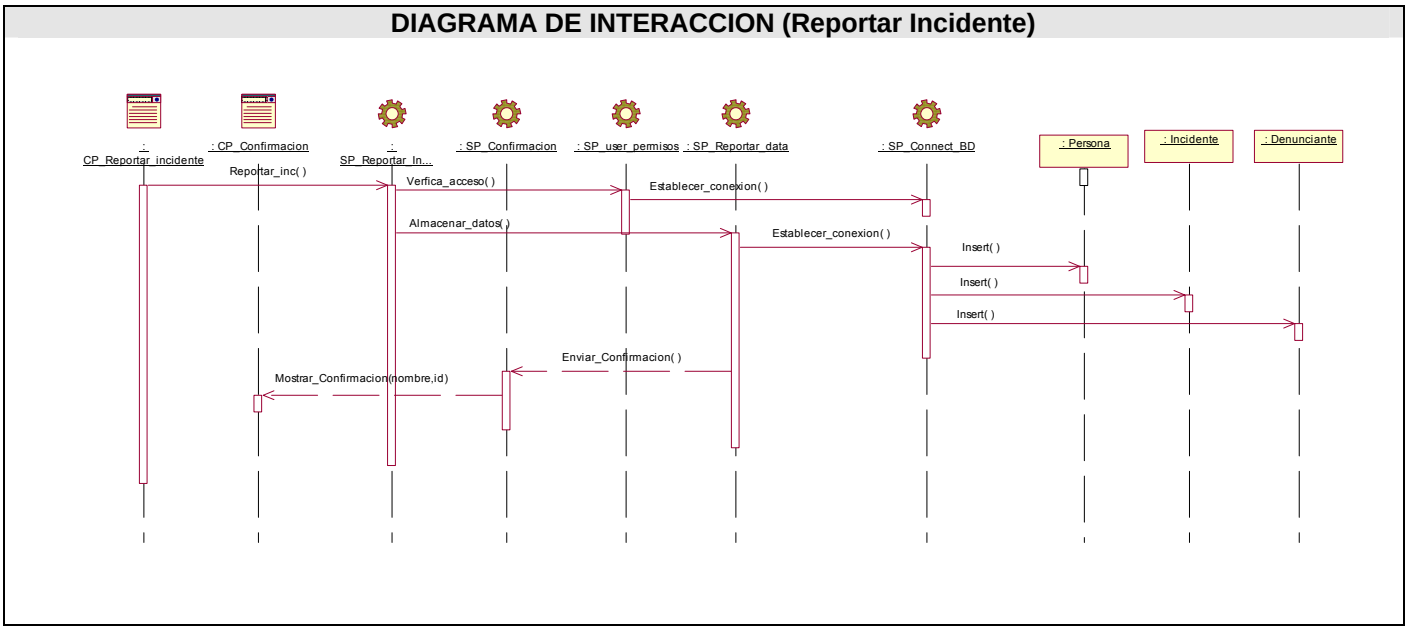

Figura 40: Diagrama de Interacción del Diseño CU Reportar Incidente.

CU Gestionar solución a incidente.

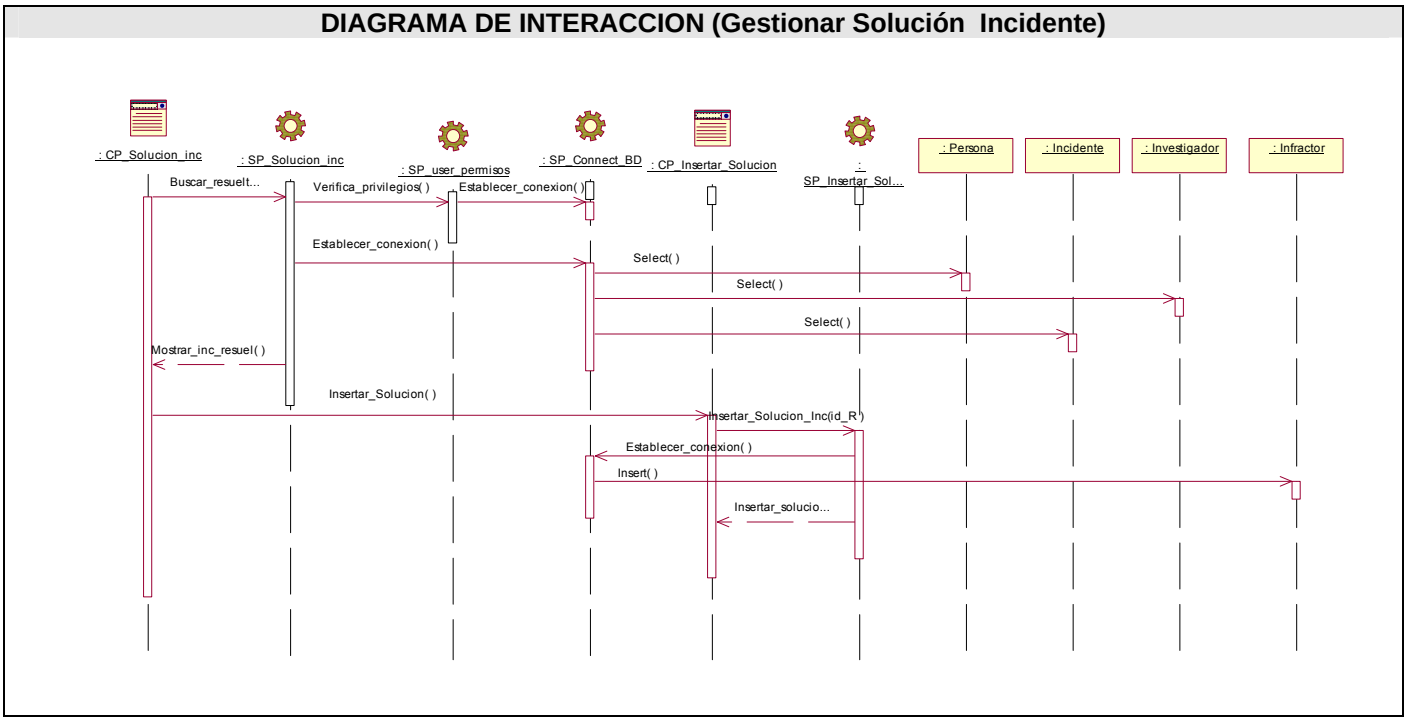

Figura 41: Diagrama de Interacción del Diseño CU Gestionar Solución Incidente.

CU Generar reporte.

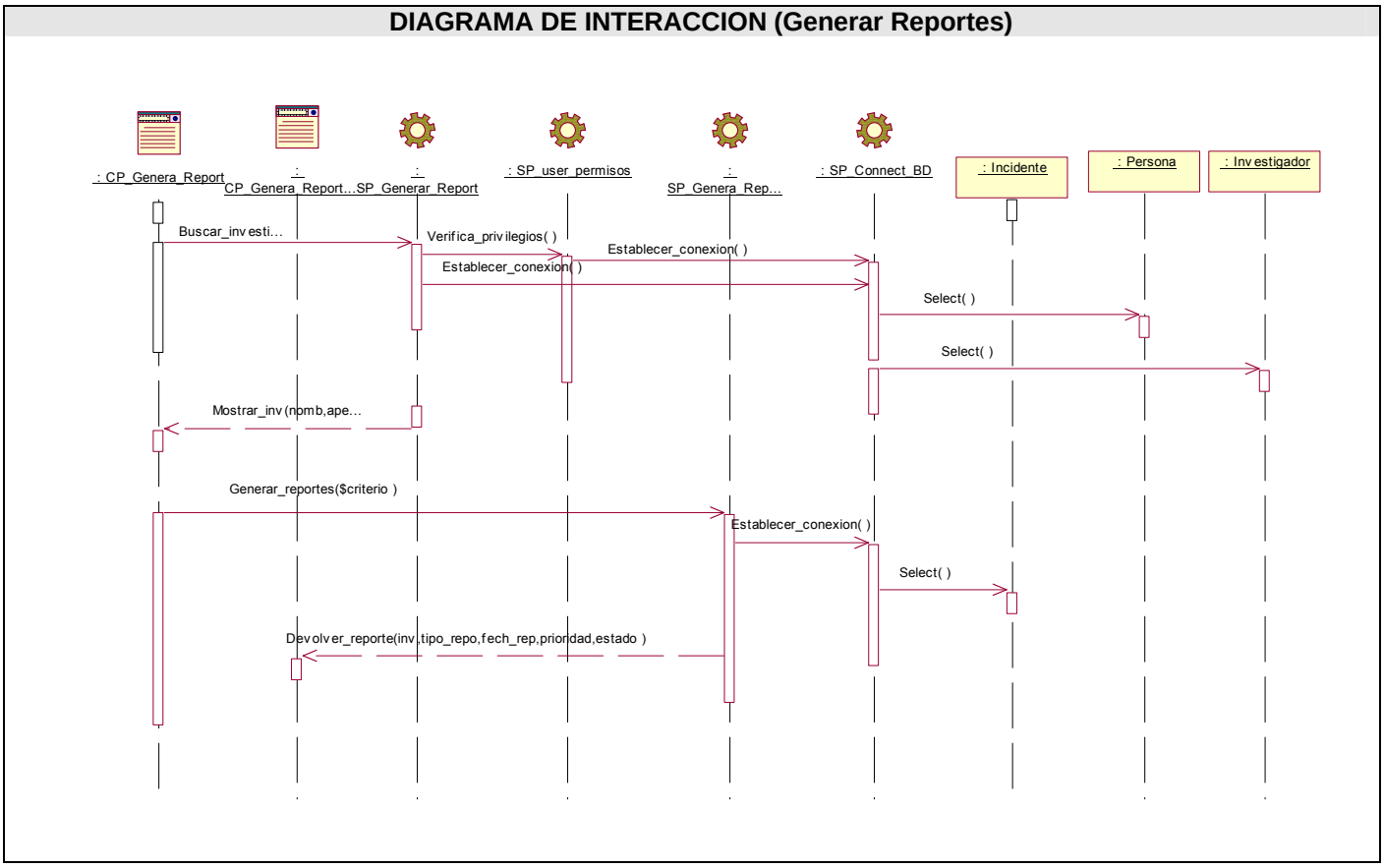

Figura 42: Diagrama de Interacción del Diseño CU Generar Reportes.

CU Mostrar solución a Incidente.

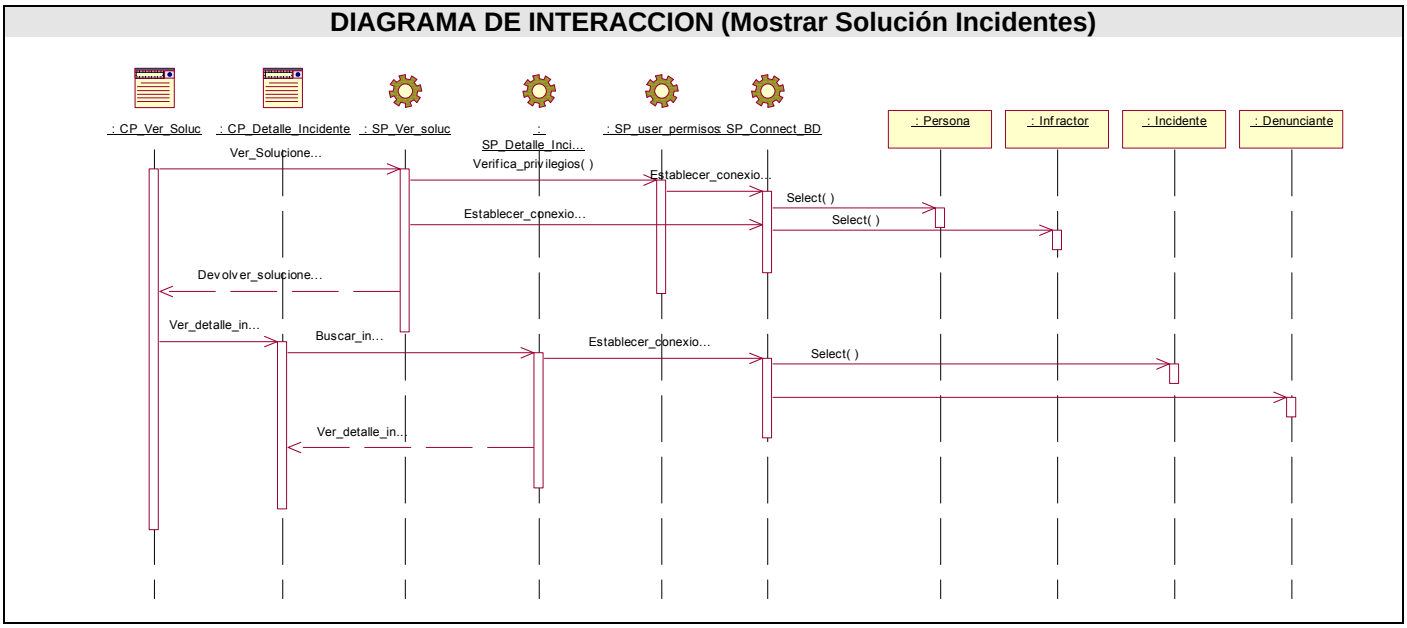

Figura 43: Diagrama de Interacción del Diseño CU Mostrar Solución Incidentes.

CU Asignar incidente.

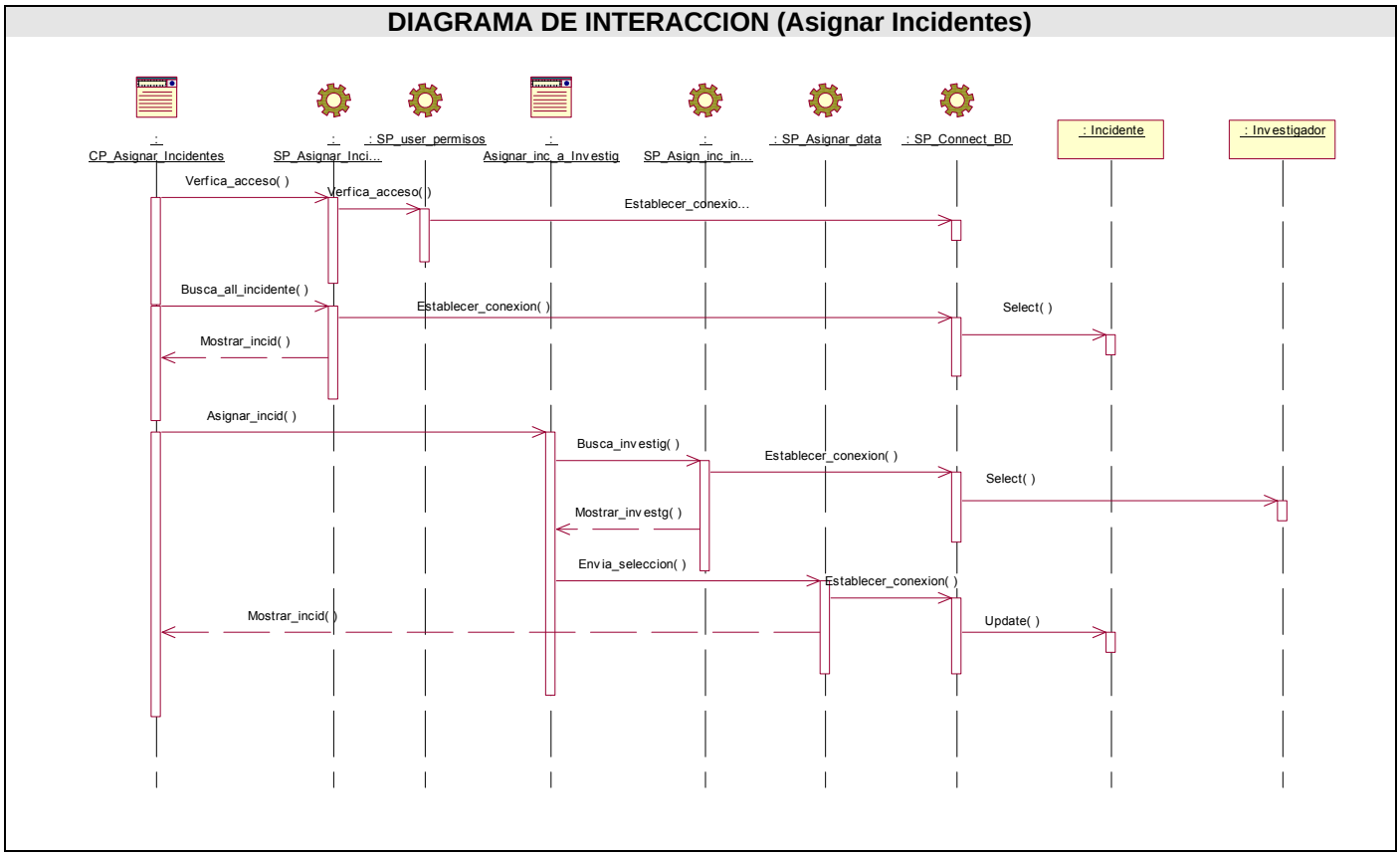

Figura 44: Diagrama de Interacción del Diseño CU Asignar Incidentes.

CU Autenticar Usuario.

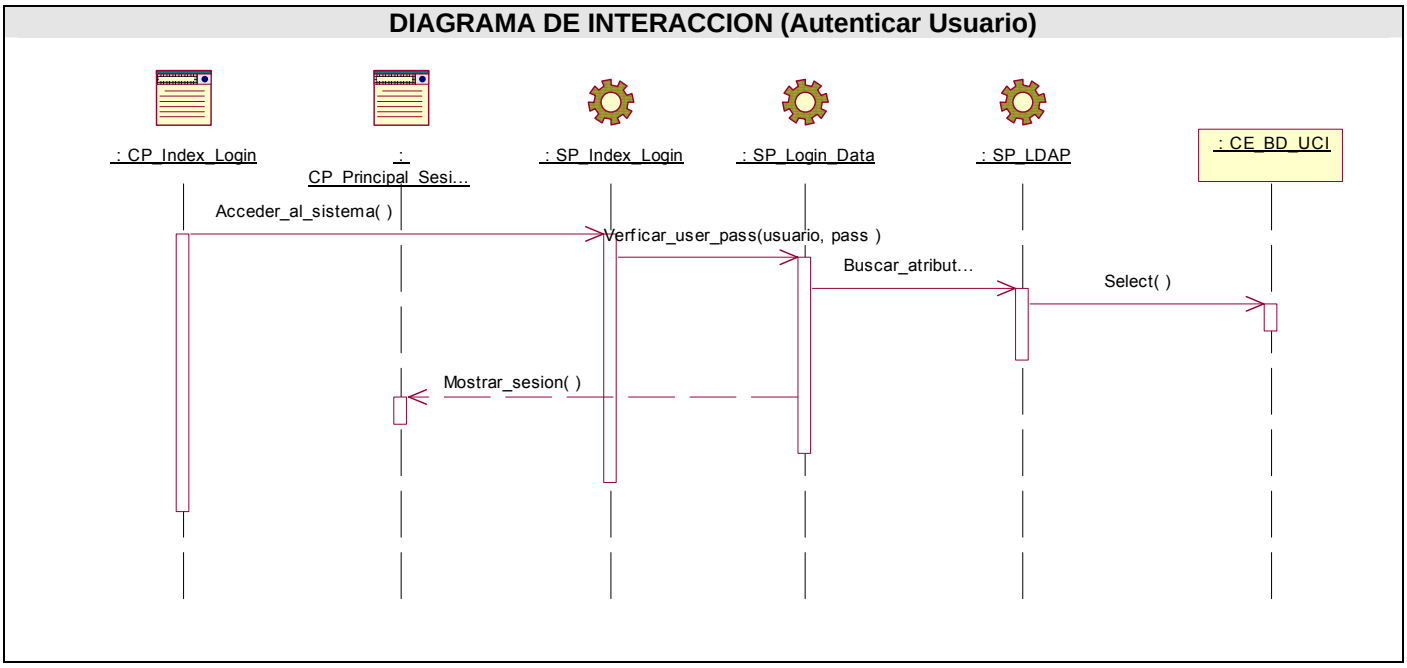

Figura 45: Diagrama de Interacción del Diseño CU Autenticar Usuario.

CU Gestionar Investigador.

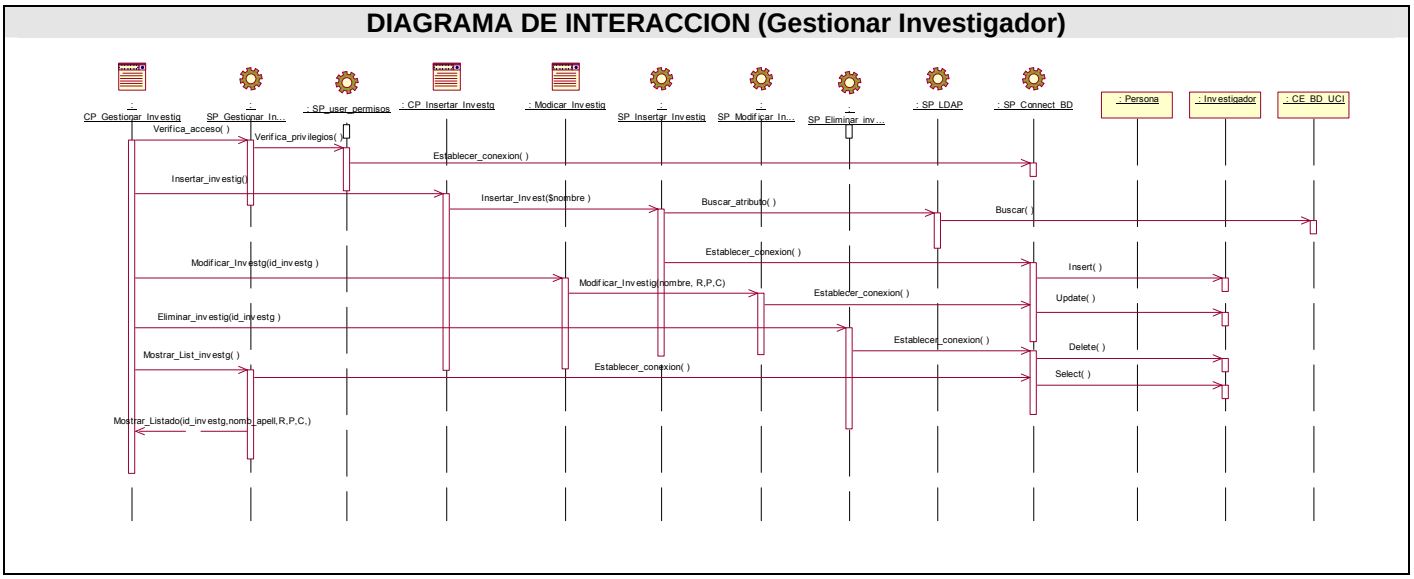

Figura 46: Diagrama de Interacción del Diseño CU Gestionar Investigador.

CU Insertar Noticia.

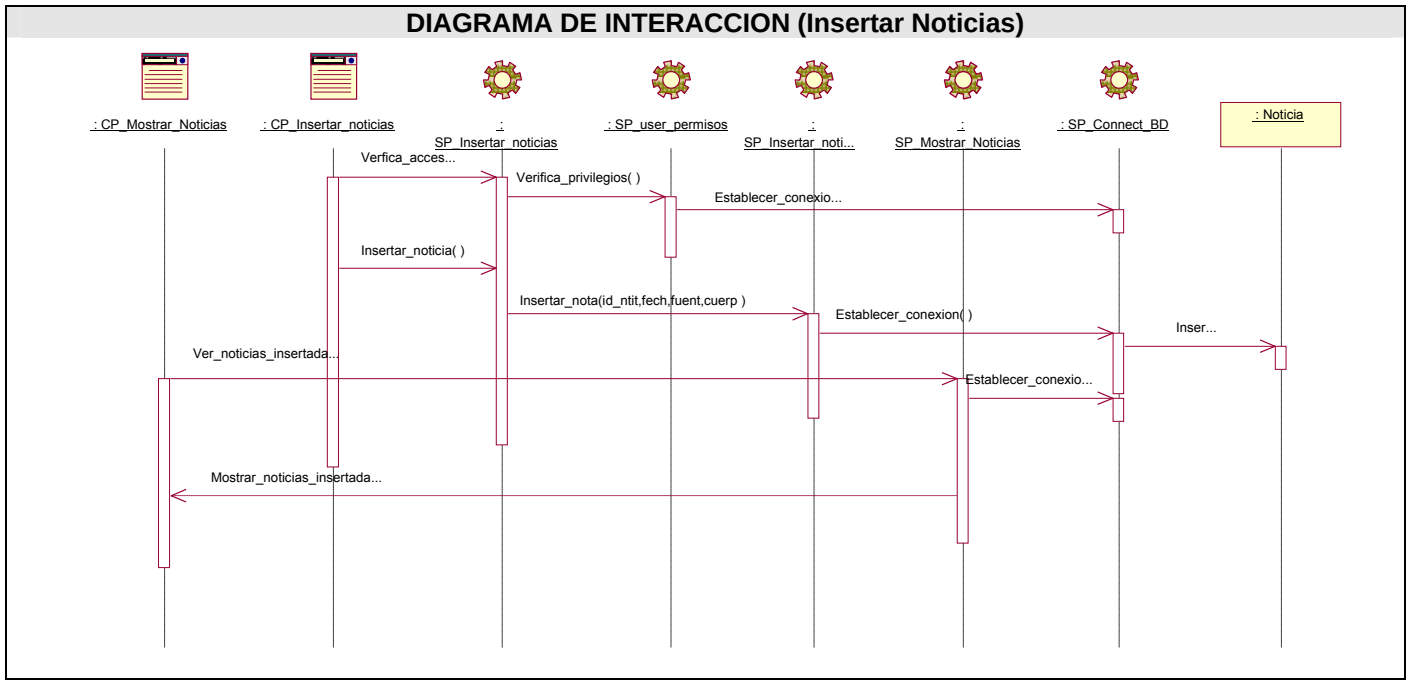

Figura 47: Diagrama de Interacción del Diseño CU Insertar Noticias.