Universidad de las Ciencias Informáticas

Facultad 5

Asistente para la generación de planes de seguridad informática.

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

Autor (es): Yamila del Carmen Rodríguez Gámez.
Roilán Galán Pérez.

Tutor (es): Ing. Ruth Yurina Vega Cutiño.
Ing. Dayren Martínez Sousa.

Ciudad de La Habana
Junio de 2009

DATOS DE CONTACTO

Ing. Ruth Yurina Vega Cutiño (ruth@uci.cu)

Graduada de Ingeniería Informática, Profesor Asistente, Aspirante a MSc. en Telemática, 5 años de experiencia en el tema, 8 años de graduada.

Ing. Dayren Martínez Sousa (dsousa@uci.cu)

Graduada de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Profesor Adiestrado de la Facultad 5, con 1 año de graduada y 3 años de experiencia en el tema.

"Aquellos que pueden dejar la libertad esencial para adquirir un poco de seguridad temporal, no merecen ni libertad, ni seguridad".

Benjamín Franklin.

Agradecimientos Roilán.

A la Revolución y a nuestro Comandante en Jefe Fidel por hacer realidad este gran sueño y forjarnos como profesionales.

A mi madre que siempre confió en mi y siempre estuvo ahí cuando la necesitaba, eres lo mas grande que tengo en mi vida, a ti mami te dedico este gran triunfo, tu mas que nadie te lo mereces, de todo corazón, te quiero con la vida… ojala y la vida te regalara la eternidad para que puedas estar siempre a mi lado, siempre te voy a llevar en mi corazón.

A mi papá por estar en mi vida en los momentos más difíciles y por brindarme su cariño.
A mi abuela Cuca y a mi tío Alejandro, gracias por regalarme su cariño y apoyo incondicional.

A mi novia, Liset, gracias por aguantarme tanto tiempo, gracias por brindarme tu cariño, amor y ternura, eres lo mejor que me ha podido pasar, te quiero mucho chiquitica.

A la familia de Liset por siempre estar preocupada por mi y por darme tanto cariño.

A mi hermana Malena y a mi hermana Yanelis, las quiero mucho.

A mi padrastro Jacinto, gracias por preocuparte siempre por mí y por cuidar muy bien de mi mamá.

A mis tías Giselita y Gloria, a mi tío Pillo, a mi tío Rafelito y a Mayte, a mi primo Canito y a Sol, gracias a todos por ser una parte importante de mi vida.

Imposible dejar de mencionar a mi segunda familia, ellos también forman un parte fundamental de mi vida, a mi segundo padre Omar, a mis otras 2 madres Marta y Gladys, a mis hermanos Héctor y Omarito, los quiero mucho a todos, especialmente a Héctor, a José, a mi tío Héctor y a mi tía Ricela, a mi abuela Carmen y mi abuelo Honorio, a mis primos Rubén, Robin y Carmencita, a todos les guardo un espacio grande en mi corazón.

A mis 2 tutoras, Ruth y Dayren, muchas gracias, por su ayuda y los consejos en todo momento y por el gran esfuerzo que realizaron con nosotros, las molestias rindieron frutos...muchas gracias.

A todos mis amigos de la universidad, especialmente a Darluin, Carlos Mario y Flavio, el piquete de siempre, a todos los amigos y amigas que de una forma u otra siempre pude contar con ellos.

A Liu, Lizandra y Reiner por su ayuda de una forma u otra.
A mi compañera de tesis, aunque a veces no confiabas en mí, pero bueno, ya ves, nos graduamos, nunca pierdas la esperanza.

A la Universidad por haberme convertido en un profesional, a conocer muchas amistades de diversos rincones y por los gratos y malos momentos que nunca voy a poder olvidar y que siempre van a estar en mi memoria.

Me gustaría agradecer a todas las personas que tengo en mi mente, pero nunca acabaría, a todas esas personas, muchas gracias, de veras.

Agradecimientos Yamila

A Dios ante todo.

A mi mamá por siempre estar conmigo en las buenas y malas, por llorar junto conmigo cada vez que me daban los ataques, por ser más que mi madre, mi amiga, mi más preciado tesoro, por ser mi fuente de inspiración.

A mi papá por darme siempre todo lo que he necesitado, porque este también era tu sueño y lo he cumplido para ti, aunque nunca te lo diga te quiero mucho.

A mi hermana por ser tan cariñosa conmigo, tu representas una parte importante de mi corazón y aunque nos peleamos mucho te quiero con el alma.

A Daniel por quererme como si fuera su hija, por preocuparse siempre por mí y porque se que estas muy orgulloso de mí, también te quiero mucho.

A mi Ner porque contigo he pasado los mejores momentos, por creer que si podía cuando yo misma no lo creía, por todos tus consejos, por estos dos años de sacrificio para poder estas juntos, por darme tanto amor, por aguantar tanto llanto por teléfono y por prestarme la laptop. Gracias por todo, te amo.

A mis abuelos Mauro, Nieve, Maby y Antidio por sus consejos, por siempre guiarme por el buen camino, por ser los mejores abuelos del mundo, los adoro.
A mi abuelita Oda y a mi abuelito Mime que en paz descanse por todos sus consejos y por su cariño.

A mis tíos, Alberto, Charo, Yurima, Katy, José y Sonia por estar orgullosos de su sobrina ya ingeniera, por apoyarme a llegar hasta el final y a mis primos los más grandes y los más pequeñitos por hacerme reír, los quiero a todos.
A toda mi familia en general.

A Roilán por aceptarme como su compañera de tesis cuando no tenia tema de tesis, por todos estos días de sacrificio sentado frente a la computadora, por siempre ser optimista y creer que el trabajo se podía terminar, gracias por todo.

A mis amistades las que siempre creyeron en mi Yelena, Yoenia y Aylet porque aun estando lejos siempre me tienen presente, gracias por estar.
A mis buenos amigos Luisi y Darian porque siempre he podido contar con ustedes.
A todos mis compañeros de universidad en especial a Daylenis, Odalys, Marilín, Dary, Mileivys, de todos siempre tuve un buen consejo, un hombro donde llorar cuando no tenía con quien desahogarme.

A mis dos tutoras Ruth y Dayren, por las veces que tuvimos que molestarlas, porque sin su ayuda no hubiésemos podido salir adelante.

A las personas que de una manera u otra hicieron su aporte para que este trabajo saliera adelante, muchas gracias por todo.
A Fidel y la Revolución.

Dedicatoria Roilán.

A mi madre, solo tú sabes lo mucho que me esforcé para darte este mérito que tanto ansiabas, ahora es nuestro triunfo, a ti te lo debo todo en esta vida, eres la mejor del mundo.

A mi novia, quien supo aconsejarme y guiarme por el mejor camino, nada de esto hubiera sido posible si tú no hubieses formado parte de mi vida. Te quiero.

A mi padre, a mi abuela, a mis tíos y a mis tías, a mis hermanas y a mis hermanos, a mis primos y a mis primas, a toda mi familia en general.

A mi familia de crianza, este triunfo también es de ustedes.

A la universidad y a la Revolución.

Dedicatoria Yamila

A mi mamá por representarlo todo para mí, por ser lo más grande que tengo en la vida.

A mi papá por sacrificarse tanto por mí, por darme siempre lo mejor.

A mi otro papá, Daniel, por ser tan bueno conmigo.

A mi hermanita del alma por ser especial y porque eres lo que más quiero.

A mis abuelos Nieve y Mauro por su amor incondicional y por quererme tanto.

RESUMEN

Con el objetivo de proteger los recursos informáticos y la información que en estos se almacena surge el Plan de Seguridad Informática (PSI) que es el documento que orienta en el uso adecuado de las tecnologías y medios informáticos y organiza toda la actividad de seguridad informática de una entidad.

El presente trabajo tiene como objetivo desarrollar una herramienta sobre plataforma web que permita la gestión de los planes de seguridad informática en la Universidad de las Ciencias Informáticas.

Para darle solución al objetivo de la investigación se analizan las metodologías de desarrollo, tecnologías, técnicas, lenguajes y herramientas más utilizadas, a partir de las cuales se hace una propuesta sobre las que deben ser empleadas en la realización del asistente. Se realiza la fase de modelación del negocio de la metodología RUP, y posteriormente el análisis, diseño e implementación del sistema a partir de la captura de requisitos. La aplicación se desarrolló en el lenguaje libre PHP mediante un conjunto de herramientas como Dreamweaver, Zend Studio y el servidor web Apache.

El Sistema para la Gestión de los Planes de Seguridad Informática brinda la posibilidad de manejar de manera sencilla y eficiente la información que se gestiona relacionada con los planes de seguridad informática en cada área de la universidad. Garantiza una mejora en el proceso de desarrollo y en la gestión de la calidad de los planes. Logrando así que el sistema brinde funcionalidades que posibiliten crear un flujo informativo rápido, confiable y seguro.

PALABRAS CLAVE

Activo, amenaza, análisis de riesgos, informático, impacto, plan de seguridad informática, riesgo, seguridad, seguridad informática.

TABLA DE CONTENIDOS

INTRODUCCIÓN

La seguridad informática consiste en asegurar que los recursos del sistema de información ya sea material informático o programas de una organización sean utilizados de la manera que se decidió y que el acceso a la información que se encuentra contenida en el mismo así como su modificación sólo sea posible a las personas que se encuentren autorizadas y dentro de los límites establecidos.

Se puede entender como seguridad un estado de cualquier tipo de información (informático o no) que nos indica que ese sistema está libre de peligro, daño o riesgo. Se entiende como peligro o daño todo aquello que pueda afectar su funcionamiento directo o los resultados que se obtienen del mismo.

Actualmente la seguridad informática ha adquirido gran auge, dadas las cambiantes condiciones y nuevas plataformas de computación disponibles, situación que desemboca en la aparición de nuevas amenazas en los sistemas informáticos. Esto ha llevado a la necesidad de que se desarrollen documentos y directrices que orientan en el uso adecuado de estas tecnologías para obtener el mayor provecho de las ventajas que brindan. Estos documentos son los planes de seguridad informática que guían todo el proceso de la actividad de seguridad informática en las organizaciones.

El Plan de Seguridad Informática se instituye como una exigencia para todas las entidades, en el que se deben reflejar las políticas y el sistema de medidas para la Seguridad Informática, teniendo en cuenta los resultados obtenidos en el análisis de riesgos y vulnerabilidad realizado. Actualmente no se conoce un software que gestione toda la información contenida en los planes de seguridad informática y de esta forma humanizar el proceso de confección de los mismos.

La Universidad de las Ciencias Informáticas surgida al calor de la batalla de ideas, es un proyecto creado por la Revolución con el objetivo de formar ingenieros informáticos comprometidos con la patria, este centro no está exento al gran avance de la informática y las TIC's; es una de las instituciones con más avances tecnológicos en todo el país y que produce software no solo para Cuba sino también para otros países.

En la UCI existen muchas áreas de diversos tamaños y es de vital importancia la seguridad. En las mismas se realizan los planes seguridad informática por parte de un personal autorizado, estos son elaborados manualmente, lo que trae consigo que se violen pasos y/o se afecte el orden de los mismos, se incumple el tiempo de entrega de los planes, factores que atentan contra la gestión adecuada del plan y pierde calidad el proceso de desarrollo de los planes de seguridad informática. Todos estos elementos evidencian una **situación problémica** a resolver.

Dada la situación problémica descrita resalta la interrogante de: ¿Cómo elevar la calidad de confección de planes de seguridad informática en la Universidad de las Ciencias Informáticas? convirtiéndose esta en un **problema científico** a resolver con la presente investigación.

Se escoge por tanto como rama de la ciencia a estudiar los software de gestión para la generación de planes de seguridad. Limitando el **campo de acción** en las Metodologías para la generación de planes de seguridad informática.

Por tales motivos se ha decidido plantear como **objetivo general** de este trabajo:

- ✓ Desarrollar un software de gestión sobre plataforma Web que permita la generación de planes de seguridad informática en la UCI.

Para darle solución al objetivo general se han propuesto varias **tareas a investigar** para organizar el trabajo y dividir el problema en áreas más reducidas, tales como:

- ✓ Analizar las metodologías para la generación de planes de seguridad informática.
- ✓ Analizar el proceso de desarrollo de los planes de seguridad informática.
- ✓ Seleccionar la metodología de desarrollo, herramientas y lenguajes adecuados para crear un software de gestión.
- ✓ Diseñar un software de gestión que posea interfaz sencilla, fácil de manejar, y con una arquitectura robusta.
- ✓ Implementar un software de gestión basándose en el diseño realizado.

Como **hipótesis** se plantea que: Si se realiza un software para la gestión de planes de seguridad informática, entonces será posible asegurar y controlar la calidad de la gestión de los planes de seguridad informática en la UCI.

Para la realización de este trabajo se utilizan los métodos científicos de la investigación, como es el caso de los métodos teóricos y los métodos empíricos. Entre los **métodos teóricos** se utiliza específicamente el *Analítico-Sintético*, este permite buscar la esencia de los fenómenos, así como los rasgos que lo caracterizan y los distinguen, y el objetivo de este método en la presente investigación es analizar las teorías, documentos, entre otros; que estén directamente relacionados con las metodologías y los planes de seguridad informática, permitiendo por esta vía la extracción de los elementos más importantes que se relacionan con el objeto de estudio.

También se utiliza el método *Hipotético-Deductivo*, mediante la forma de razonamiento que plantea este método se obtienen un grupo de conocimientos generalizadores para darle cumplimiento al objeto de estudio, el mismo se usa para realizar suposiciones en determinados aspectos relacionados con el tema, y así poder aplicar e incorporar conocimientos en el desarrollo de las tareas, y de esta forma llegar a la conclusión de que se podrá cumplir con el trabajo de diploma.

Conjuntamente con los métodos teóricos, se emplearán **métodos Empíricos** como la E*ntrevista* pues para la realización de este trabajo de diploma es necesario establecer conversaciones planificadas con el fin de obtener información y opiniones del entrevistado sobre el estado del arte del objeto de estudio.

El contenido de esta investigación esta desglosado en 4 capítulos, las conclusiones generales, recomendaciones, referencias bibliográficas y bibliografía utilizada, un glosario de términos y los anexos que complementan el cuerpo del trabajo y son necesarios para su entendimiento.

Capítulo I. Fundamentación Teórica: Este capítulo tiene como objetivo abarcar brevemente el tema de seguridad en redes para de esta manera profundizar sobre la necesidad de un plan de seguridad

informática, su estructura e importancia. Consecutivamente tiene como objetivo investigar sobre la metodología vigente para la realización de planes de seguridad informática en Cuba y analizar el estado del arte de las herramientas existentes para el análisis de riesgo.

Capítulo II. Lenguajes, Metodologías y Herramientas: En este capítulo se analizan las metodologías de desarrollo, las herramientas y lenguajes de programación para realizar aplicaciones Web, y se determinan las que se emplearán en la elaboración del asistente para la generación de planes de seguridad informática. Se estudia la arquitectura cliente – servidor y se aborda acerca de la seguridad en aplicaciones Web.

Capítulo III. Características del sistema: Se realiza el estudio del funcionamiento del negocio, se describen los actores y trabajadores involucrados y se incluye conjuntamente el diagrama de actividades y el diagrama de clases del modelo de objetos. Además, se detallan elementos imprescindibles como los requerimientos funcionales y no funcionales del sistema, así como los actores del mismo, se representa el modelo de casos de uso y se realiza la expansión de los casos de uso del sistema.

Capítulo IV. Análisis, Diseño e Implementación del sistema: En este capítulo se muestran los elementos del análisis y diseño de la solución tales como los diagramas de clases del análisis, los diagramas de interacción y los diagramas de clases del diseño con extensiones web, además se aborda el tema de los patrones de diseño utilizados. Por otra parte se añade una descripción de la base de datos mediante el diagrama de clases persistentes y el modelo de datos. Consecutivamente en parte que describe la implementación del sistema se muestra la estructura del sistema en términos de componentes a través del diagrama de componentes.

1
FUNDAMENTACIÓN TEÓRICA

Este capítulo tiene como objetivo realizar un estudio acerca de los fundamentos teóricos para la comprensión de la solución del problema, para lo cual se realiza un análisis de la seguridad informática en las redes de computación, también se estudia detenidamente acerca de las metodologías para la generación de planes de seguridad informática así como las etapas en el desarrollo de los planes de seguridad informática, desglosando cada una de ellas lo que conforma la estructura del PSI. Se aborda las metodologías y herramientas para realizar el análisis de riesgo.

1.1 Seguridad Informática en redes.

Una red informática es un conjunto de ordenadores o computadoras, de host autónomos y dispositivos especiales conectados entre sí, que pueden comunicarse compartiendo datos y recursos sin importar la localización física de los distintos dispositivos. (Otrera, 2008)

Mediante la compartición de información y recursos en una red, los usuarios de los sistemas informáticos de una organización podrán hacer un mejor uso de los mismos, perfeccionando de este modo el rendimiento global de la entidad. Entre las ventajas que supone el tener instalada una red, se pueden mencionar las siguientes:

- ✓ Mayor facilidad en la comunicación entre usuarios.
- ✓ Reducción en el presupuesto para la adquisición de software.
- ✓ Reducción en el presupuesto para la adquisición de hardware.
- ✓ Mejoras en la administración de los equipos y programas.
- ✓ Mejoras en la integridad de los datos.
- ✓ Mayor seguridad para acceder a la información.

Para obtener todas estas ventajas que provee el uso de una red se deben tener instalados una serie de servicios de red como son:

- ✓ Acceso
- ✓ Ficheros
- ✓ Impresión
- ✓ Información
- ✓ Correo electrónico
- ✓ Entre otros

Para disponer de todos estos servicios se necesita montar el hardware adecuado, entre los componentes de hardware se pueden encontrar tarjetas de red, concentradores, puentes, repetidores, routers, y otros, además de los ordenadores existentes en la red tales como los servidores y las estaciones de trabajo.

Es importante destacar que hoy en día las organizaciones y sus sistemas de información están expuestos a un gran número de amenazas debido a la cantidad de recursos que poseen al tener instalada una red informática, por eso la seguridad informática es la que se encarga de desarrollar técnicas y actividades para proteger los equipos informáticos individuales y conectados a la red frente a daños accidentales e intencionados. (MASTERMAGAZINE, 2004)

Por lo planteado anteriormente se puede decir que la Seguridad Informática tiene como principal objetivo proteger los recursos que existen en la red manteniendo 5 aspectos fundamentales, integridad, confidencialidad, disponibilidad, no rechazo y autenticación. Para mantener estos 5 aspectos están los Planes de Seguridad Informática.

1.2 Planes de seguridad informática.

Plan de Seguridad Informática: Documento básico que establece los principios organizativos y funcionales de la actividad de seguridad informática en una entidad. (Resolución 127, 2007)

El Plan de Seguridad Informática constituye el documento para lograr la confidencialidad, integridad y disponibilidad de la información y la protección de los medios y los locales donde se utilice la técnica de computación.

Para tener éxito en la realización de un Plan de Seguridad Informática se debe realizar una gestión de seguridad en la que participen todos los miembros activos de la organización incluyendo la dirección de la organización y teniendo en cuenta además clientes y proveedores de bienes y servicios. El máximo dirigente de cada entidad garantiza, según corresponda a la actividad informática que se desarrolle, que se elabore, ponga en vigor, cumpla y actualice periódicamente.

En el desarrollo de este plan es necesario formular las políticas de seguridad, establecer una estructura de gestión de la seguridad informática, elaborar el sistema de medidas de seguridad informática, implantar el programa de seguridad informática y elaborar el plan de contingencia de la entidad. (Rodríguez Aneiro, 2001)

Para la elaboración del Plan de Seguridad Informática se tendrán en cuenta las consideraciones siguientes:

- ✓ Serán confeccionados tantos ejemplares como se determine en cada lugar, enumerando las páginas consecutivamente.
- ✓ Se clasificarán de acuerdo al contenido e importancia que expresa dicho plan en cada lugar.
- ✓ Contendrán las tablas y gráficos que se consideren necesarios y contribuyan a su mejor interpretación.
- ✓ Tendrán acceso a este documento, o a parte de él, las personas que en cada área requieran de su conocimiento.
- ✓ Se mantendrá permanentemente actualizado sobre la base de los cambios que se produzcan en las condiciones que se consideraron durante su elaboración. (Ministerio del Interior, 2008)

1.3 Metodología para la generación de Planes de Seguridad Informática.

Para llevar a cabo la realización de los planes de seguridad informática cada entidad, organización, empresa u organismo realiza un estudio integral de los procesos e información de la institución y genera un documento en el que se recogen todos los aspectos que conforman un plan de seguridad informática.

En Cuba el diseño de un Plan de Seguridad Informática de cada entidad se realiza en correspondencia con la metodología que establece al respecto la Oficina de Seguridad para las Redes, que esta adscripta al Ministerio de la Informática y las Comunicaciones y es por donde estará regido este trabajo.

En esta metodología se describen los elementos fundamentales que deben ser incluidos en el Plan de Seguridad Informática de una entidad (contenido) y el modo que pueden ser estructurados (formato). Tiene un carácter general, y no esta orientado a un tipo determinado de entidad o sistema informático, sino que debe ser considerada como una guía de trabajo y no es necesario seguirlos al pie de la letra, por lo que el equipo designado para la elaboración desarrolla los aspectos que considere necesarios a partir del Sistema de Seguridad Informática que previamente se haya diseñado para la entidad en cuestión. Serán excluidos los aspectos que no se correspondan a las necesidades de protección identificadas y se adicionará cualquier elemento que se considere importante para los requerimientos de seguridad. (Ministerio del Interior, 2008)

1.4 Etapas en el desarrollo de los Planes de Seguridad Informática.

El Plan de Seguridad Informática es la expresión gráfica del Sistema de Seguridad Informática diseñado y durante el proceso de diseño de un Sistema de Seguridad Informática se distinguen tres etapas.

1. **Determinar las necesidades de protección del sistema informático objeto de análisis**, que incluye:
    - ✓ Caracterización del sistema informático.
    - ✓ Identificación de las amenazas y estimación de los riesgos.
    - ✓ Evaluación del estado actual de la seguridad.

2. **Definir e implementar el sistema de seguridad que garantice minimizar los riesgos identificados en la primera etapa.**
    - ✓ Definir las políticas de seguridad.
    - ✓ Definir las medidas y procedimientos a implementar.
3. **Evaluar el sistema de seguridad diseñado.**

1.5 Estructura y desglose del Plan de Seguridad Informática.

En este epígrafe se detallan cada uno de los aspectos de las etapas mencionadas anteriormente, estructurando así el Plan de Seguridad Informática.

1.5.1 Caracterización del Sistema Informático.

La Caracterización del Sistema Informático describe el resultado de la caracterización realizada al sistema informático de la entidad, con el objetivo de determinar qué se trata de proteger, especificando sus principales componentes y considerando entre otros:

- ✓ Bienes informáticos, su organización e importancia.
- ✓ Redes instaladas, estructura, tipo y plataformas que utilizan.
- ✓ Aplicaciones en explotación.
- ✓ Características del procesamiento, transmisión y conservación de la información.
- ✓ Otros datos de interés.

1.5.2 Identificación de amenazas y estimación de riesgos.

Identificación de las amenazas y estimación de los riesgos, en este paso se identifican las amenazas con posibilidad de afectar los activos de la empresa y las vulnerabilidades, se valora su impacto y la probabilidad de que ocurran.

Objetivo del análisis de riesgo

- ✓ Identificar, evaluar y manejar los riesgos de seguridad.
- ✓ Estimar la exposición de un recurso a una amenaza determinada.

- ✓ Determinar cual combinación de medidas de seguridad proporciona un nivel de seguridad razonable a un costo aceptable.
- ✓ Tomar mejores decisiones en seguridad informática.
- ✓ Enfocar recursos y esfuerzos en la protección de los activos.

Las necesidades de protección se determinan mediante la realización de un análisis de riesgos, que es el proceso dirigido a determinar la probabilidad de que las amenazas se materialicen sobre los bienes informáticos, e implica la identificación de los recursos a proteger, las amenazas que actúan sobre ellos, su probabilidad de ocurrencia y el impacto que puedan causar.

Existen varias metodologías para el análisis de riesgos, entre las cuales se encuentran:

MAGERIT: Es una metodología de análisis de riesgo de carácter público elaborada en España por el Ministerio de Administraciones Públicas.

MAGERIT responde a Metodología de Análisis y GEstión de Riesgos de los sistemas de Información de las Administraciones públicas, y es un método formal orientado a activos, cuya misión es descubrir los riesgos a los que se encuentran expuestos nuestros sistemas de información y recomendar las medidas apropiadas que deberían adoptarse para controlar estos riesgos**.** (Expósito Gutiérrez, 2003)

MAGERIT persigue los siguientes objetivos generales:

- ✓ Concienciar a los responsables de los sistemas de información de la existencia de riesgos y de la necesidad de impedirlos a tiempo.
- ✓ Ofrecer un método sistemático para analizar tales riesgos.
- ✓ Ayudar a descubrir y planificar las salvaguardas oportunas para mantener los riesgos bajo control.
- ✓ Apoyar a la Organización en la preparación de procesos de evaluación, auditoría, certificación o acreditación, según corresponda en cada caso.

CRAMM (CCTA Metodología para la Gestión de Análisis de Riesgo): Es una metodología que aplica sus conceptos de una manera formal, estructurada y disciplinada, esta orientada a proteger la confidencialidad, integridad y disponibilidad de un sistema y de sus activos. Fue creada por la Agencia Central de Computación y Telecomunicaciones del Gobierno del Reino Unido.

El análisis de riesgo en CRAMM se basa en identificar y valorar los activos datos (es decir la información contratada por los sistemas de tecnología de la información), los activos software (del sistema de aplicaciones) y los activos físicos (equipos, instalaciones en general). Identificar las amenazas al conjunto de activos, haciendo además una evaluación de los grados o importancia de estas vulnerabilidades y evaluar los niveles de riesgos calculados a partir de las valoraciones de los activos, de los niveles de amenazas y vulnerabilidades evaluados. (Calle Guglieri, 2005)

Ebios: es una metodología de gestión de los riesgos de seguridad de sistemas de información desarrollada por la Dirección Central de la Seguridad de los Sistemas de Información francesa.

El procedimiento metodológico propuesto por Ebios porta una visión global y coherente de la seguridad de los sistemas de información (SSI). Este método permite determinar objetivos y requerimientos de seguridad. Toma en cuenta todas las entidades técnicas (software, hardware, redes) y no técnicas (organización, aspectos humanos, seguridad física). Permite implicar a todos los actores de Sistema Informático y propone un procedimiento dinámico que estudia todo el ciclo de vida del sistema (diseño, realización, puesta en servicio, mantenimiento, y otros). (Premier Ministre, 2004)

OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation) Es un método de análisis de riesgos orientado a activos, desarrollado por el CERT (Instituto Central de Ingeniería de Software) de la Universidad Carnegie Mellon de Pensilvania, Estados Unidos. El objetivo de OCTAVE es desarrollar una perspectiva de seguridad dentro de una organización, teniendo en cuenta perspectivas de todos los niveles para asegurarse que las soluciones puedan implementarse con facilidad.

En OCTAVE, los activos incluyen personas, hardware y software, información y sistemas. Los activos se ordenan según la importancia que tienen para los objetivos de la organización, y las posibles amenazas y vulnerabilidades asociadas a dichos activos, así como el impacto que causaría un problema en cada activo. (Cert, 2008)

Estas metodologías no son las únicas pues existe una gran variedad de las mismas como ISAMM, MARION, MIGRA y MEHARI. Como metodologías de análisis de riesgo todas se parecen pero no se puede decir que sean compatibles de entrada pues cada una tiene sus particularidades y la selección de las opciones correctas para una organización depende de la gama de leyes y las regulaciones, las metas y los objetivos de la organización.

En nuestro país para desarrollar el análisis de riesgo se utiliza la metodología MAGERIT pero adaptada a nuestras empresas, es por eso que no se utiliza ninguna herramienta existente para desarrollar el análisis de riesgo, de ahí que el asistente para la generación de Planes de Seguridad Informática en la UCI facilite el trabajo de los especialistas análisis de riesgo pues todo este proceso será automatizado lo que contribuirá para que el proceso de desarrollo de los planes de seguridad informática sea más grato.

Algunas de estas herramientas son:

- ✓ EAR/PILAR: basada en la metodología MAGERIT y gratuita.
- ✓ OCTAVE: como su nombre lo indica se basa en la metodología OCTAVE y es gratuita.
- ✓ CRAMM: herramienta no gratuita basada en la metodología CRAMM.

En el resultado del análisis de riesgos se precisan claramente los resultados obtenidos por lo que en este paso deberán relacionarse las principales conclusiones obtenidas en ese proceso, entre las que no deben faltar:

- ✓ Cuáles son los activos y recursos más importantes para la gestión de la entidad y por lo tanto requieren de una atención especial desde el punto de vista de la protección, especificando aquellos considerados de importancia crítica por el peso que tienen dentro del sistema.

- ✓ Qué amenazas actúan sobre los activos y recursos a proteger y entre ellas las de mayor probabilidad de materializarse (riesgo) y su posible impacto sobre la entidad.
- ✓ Cuáles son los activos, recursos y áreas con un mayor peso de riesgo y que amenazas lo motivan.

En la medida en que las conclusiones del análisis de riesgos sean más precisas se logra una visión más acertada de hacia donde pueden ser dirigidos los mayores esfuerzos de seguridad y por supuesto los recursos disponibles para ello, logrando que la misma sea más rentable.

Una vez caracterizado el sistema, identificadas las amenazas y realizada la estimación de riesgos se realiza una evaluación del estado actual de la seguridad y se concluye con la etapa uno en el desarrollo del Plan de Seguridad Informática.

1.5.3 Definir las políticas de seguridad informática.

Culminada la primera etapa se continúa con otro aspecto fundamental: Definir las políticas de seguridad informática.

Las Políticas de Seguridad Informática son normas materializadas en documentos que describen la forma correcta y adecuada del uso de los recursos computacionales, las responsabilidades y derechos de administradores y usuarios, además de cómo encarar las contingencias y la previsión de amenazas.

De acuerdo con el análisis de riesgos realizado se definen los aspectos que conforman la estrategia a seguir por la entidad sobre la base de sus características propias y el sistema de seguridad diseñado. Al definir las políticas de Seguridad Informática se consideran, entre otros, los aspectos siguientes:

- ✓ El empleo conveniente y seguro de las tecnologías instaladas y cada uno de los servicios que éstas pueden ofrecer.
- ✓ La definición de los privilegios y derechos de acceso a los activos de información para garantizar su protección contra modificaciones no autorizadas, pérdidas o revelación.
- ✓ La salva y conservación de la información.

- ✓ La definición de los principios relacionados con el monitoreo del correo electrónico, la gestión de las trazas de auditoría y el acceso a los ficheros de usuario.
- ✓ El mantenimiento, reparación y traslado de las tecnologías y el personal técnico que requiere acceso a las mismas por esos motivos.
- ✓ Los principios generales para el tratamiento de incidentes y violaciones de seguridad.

1.5.4 Establecer el Sistema de Seguridad Informática.

Tomando como otro aspecto fundamental el Sistema de Seguridad Informática es importante decir que aquí se describe cómo se implementan, en las áreas a proteger, las políticas generales que han sido definidas para toda la entidad, en correspondencia con las necesidades de protección en cada una de ellas, atendiendo a sus formas de ejecución, periodicidad, personal participante y medios**.**

Deben describirse por separado los controles de seguridad implementados en correspondencia con su naturaleza, de acuerdo al empleo que se haga de los medios humanos, de los medios técnicos o de las medidas y procedimientos que debe cumplir el personal.

Medios Humanos: En este punto se debe hacer referencia al papel del personal dentro del sistema de seguridad implementado, definiendo sus responsabilidades y funciones respecto al diseño, establecimiento, control, ejecución y actualización del mismo, y además deben especificarse las atribuciones y funciones de las distintas categorías de personal.

Medios técnicos de seguridad: Deben describirse los medios técnicos utilizados en función de garantizar niveles de seguridad adecuados, tanto al nivel de software como de hardware, así como la configuración de los mismos, algunos de los siguientes ejemplos se tienen en cuenta:

- ✓ Sistemas Operativos y nivel de seguridad instalado.
- ✓ Tipo de redes utilizadas y topología de las mismas.
- ✓ Conexiones a redes externas a la entidad.
- ✓ Servidores de uso interno y externo.
- ✓ Configuración de los servicios.

- ✓ Barreras de protección.
- ✓ Alarmas
- ✓ Entre otros.

Definir las Medidas y Procedimientos de Seguridad Informática.

En esta parte del Plan se relacionan las acciones que deben ser realizadas en cada área específica por el personal, en correspondencia con las políticas generales para toda la entidad y con la ayuda, en los casos que lo requieran, de los medios técnicos, adecuando las mismas a las necesidades de protección de cada una de ellas de acuerdo con el peso del riesgo estimado para cada bien informático objeto de protección.

Las medidas y procedimientos de Seguridad Informática que de manera específica (no deben ser confundidas con las políticas que tienen un carácter general) sean requeridas en las distintas áreas, serán definidas de manera suficientemente clara y precisa, evitando interpretaciones ambiguas por parte de quienes tienen que ejecutarlas y son de obligatorio cumplimiento por las partes implicadas.

Una medida de seguridad informática es un conjunto de acciones orientadas al fortalecimiento del Sistema de Seguridad Informática. Un procedimiento de seguridad es una secuencia predeterminada de acciones dirigida a garantizar un objetivo de seguridad, debe especificarse lo más claro y sencillo posible. Detallando en cada caso: **qué** se hace, **cómo** se hace y **quién** lo hace, así como los recursos que sean necesarios para su cumplimiento.

Las Medidas de Seguridad Informática, se clasifican de acuerdo a su origen en:

Medidas de protección física.

A las áreas con tecnologías instaladas: Precisar áreas vitales de acuerdo a la información.

A las tecnologías de información: Identificar las áreas que requieren del empleo de medios técnicos de protección física.

A los soportes de información: Debe describirse el régimen de control establecido sobre los soportes magnéticos de información.

Medidas Técnicas o lógicas.

*Identificación de usuarios**:*** se explica el método empleado para la identificación de los usuarios ante los sistemas, servicios y aplicaciones.

Autenticación de usuarios: se especifica el método de autenticación empleado para comprobar la identificación de los usuarios ante los sistemas, servicios y aplicaciones.

Control de acceso a los activos y recursos: se describen las medidas y procedimientos que aseguran el acceso autorizado a los activos de información y recursos informáticos.

Integridad de los ficheros y datos: se aclaran por escrito las medidas y procedimientos establecidos con el fin de evitar la modificación no autorizada, destrucción y pérdida de los ficheros y datos, así como para impedir que sean accedidos públicamente.

Auditoría y alarmas: se informan las medidas y procedimientos implementados para el registro y análisis de las trazas de auditoría en las redes y sistemas instalados, con el fin de monitorear las acciones que se realicen (acceso a ficheros, dispositivos, empleo de los servicios, etc.), y detectar indicios de hechos relevantes a los efectos de la seguridad que puedan afectar la estabilidad o el funcionamiento del sistema informático.

Medidas de Seguridad de operaciones.

De recuperación ante contingencias: Estas describen las medidas y procedimientos de neutralización y recuperación ante cualquier eventualidad que pueda paralizar total o parcialmente la actividad informática o degraden su funcionamiento.

Medidas Legales

La violación de los Reglamentos Internos y de la Red, constituye una violación administrativa y como tal se le dará el tratamiento legal correspondiente.

Medidas Educativas

Dar a conocer el Plan de Seguridad Informática y los Reglamentos correspondientes a todos los trabajadores, con vistas a su educación.

Medidas Administrativas

Se establece el estricto cumplimiento del Reglamento Interno en todas las dependencias de la Entidad, así como el Reglamento de la Red. Se faculta a todos los Directivos y Activistas de Seguridad Informática para exigir su cumplimiento.

Por su forma de actuar, las medidas de seguridad pueden ser:

Medidas Preventivas

Como medidas preventivas se establece la instalación de los Antivirus en las PC y su obligatoria actualización y escaneo. Actualizar los Sistemas Operativos con todas las herramientas necesarias, monitoreo de la red para conocer como trabajan los usuarios y establecer alertas además de tomar bitácoras del tráfico en la red para posterior análisis y valoraciones.

Medidas Detectivas

Ante un hecho extraordinario ocurrido, informar a las instancias competentes y tomar las medidas correspondientes.

Medidas de Recuperación

Las medidas de recuperación ante un hecho son establecidas a partir de la identificación de los posibles incidentes que pueden causar la interrupción o afectación de los procesos informáticos y garantizan las acciones de respuesta a realizar, la determinación de los responsables de su cumplimiento y los recursos necesarios para ello.

Algunos procedimientos de seguridad que entre otros pueden ser implementados en un Plan de Seguridad Informática son de:

- ✓ Administración de cuentas de usuario.
- ✓ Asignación de derechos y privilegios.
- ✓ Gestión de incidentes.
- ✓ Gestión de salvas.
- ✓ Gestión de auditoría.
- ✓ Acceso a las áreas.
- ✓ Entrada y salida de tecnologías y sus soportes.

Luego de establecer las políticas, medidas y procedimientos que conforman la segunda etapa se procede a evaluar el sistema de seguridad diseñado, culminando así la elaboración del PSI.

2

LENGUAJES, METODOLOGÍAS Y HERRAMIENTAS.

En este capítulo se realiza una fundamentación de varias herramientas y lenguajes existentes en el mundo para realizar aplicaciones web, se determina la metodología de desarrollo a utilizar y el lenguaje de programación apropiado. Además se aborda el tema de la arquitectura cliente servidor y el tratamiento de seguridad en las aplicaciones Web.

2.1 Metodologías de desarrollo.

El desarrollo de software es una tarea difícil, como resultado a este problema ha surgido una alternativa desde hace mucho tiempo: las metodologías de desarrollo de software, estas imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente.

En un proyecto de desarrollo de software la metodología define *quién* debe hacer *qué*, *cuándo* y *cómo* debe hacerlo. Una metodología es un proceso, no existe una metodología de software universal. Las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigen que el proceso sea configurable.

2.1.1 Proceso Unificado de Desarrollo (RUP).

La metodología RUP es más apropiada para proyectos grandes debido a que requiere un equipo de trabajo capaz de administrar un proceso complejo en varias etapas aunque también se utiliza en proyectos pequeños.

RUP provee una aproximación disciplinada para asignar tareas y responsabilidades. Su objetivo es asegurar la producción de software de alta calidad. RUP permite controlar el proceso de desarrollo del proyecto al mismo tiempo que es elaborado, quedando conformada, a su vez, una guía para posteriores mejoras del producto. Posibilita aminorar la aparición de riesgos críticos que perjudiquen la realización del

éxito del proyecto y simultáneamente, ayuda en gran medida a disminuir el tiempo de desarrollo y los costos, así como a elevar la calidad del producto.

Es un proceso que se basa mucho en la documentación, además utiliza UML para preparar todos los esquemas de un sistema de software. Tiene una cantidad considerable de documentos publicados que se pueden consultar para esclarecer dudas. Una característica importante que posee es que permite corregir errores en cada iteración y es flexible a cambios en los requerimientos.

RUP provee un grupo de buenas prácticas que proporcionan un conjunto de beneficios. Entre ellas se debe mencionar el desarrollo iterativo, que nos brinda la posibilidad de que los elementos sean integrados progresivamente, facilita el rehúso y resulta un producto más robusto pues los errores se van corrigiendo en cada iteración. También señalar como una de las buenas prácticas que posee RUP la arquitectura basada en componentes que permite una arquitectura modular , un diseño de componentes reusables y el aprovechamiento de infraestructuras comerciales como COM, CORBA, JavaBeans. Igualmente nos permite la administración de cambios al software permitiendo así una mejor identificación de los recursos básicos en las prioridades y riesgos del proyecto.

Tiene 3 características fundamentales:

- ✓ Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, constituyen la guía fundamental establecida para las actividades a realizar durante todo el proceso de desarrollo del sistema.
- ✓ Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo.
- ✓ Iterativo e incremental: RUP divide el proyecto en fases de desarrollo, propone además que cada una de ellas se desarrolle en iteraciones, las cuales aportan un incremento en el proceso de desarrollo y terminan con el cumplimiento del punto de control trazado en la fase.

La metodología RUP divide en 4 fases el desarrollo del software y cada etapa tiene un objetivo específico.

- ✓ **Inicio**: determinar la visión del proyecto.
- ✓ **Elaboración**: determinar la arquitectura óptima.

- ✓ **Construcción**: lograr la capacidad operacional inicial.
- ✓ **Transición**: obtener el release del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, esto consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes. (IBM, 2001)

Una particularidad de esta metodología es que en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

2.1.2 Programación Extrema (XP)

Programación Extrema fue concebida y desarrollada para atender las necesidades específicas de desarrollo de software llevadas a cabo por equipos pequeños según la evolución de las necesidades. XP reconoce que los proyectos tienen que trabajar para lograr la reducción de costes y aprovechar las economías una vez que han sido obtenidos.

Características de XP, la metodología se basa en:

- ✓ **Pruebas Unitarias:** son las pruebas realizadas a los principales procesos.
- ✓ **Re-fabricación:** se basa en la reutilización de código, creando patrones o modelos estándares, siendo más flexible al cambio.
- ✓ **Programación en pares:** consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. (Mendoza Sánchez, 2004)

XP tiene la flexibilidad necesaria para añadir y eliminar la complejidad innecesaria de los proyectos que con un diseño simple evolucionan constantemente.

2.1.3 SCRUM.

Otra metodología de desarrollo de software que existe en el mundo de hoy es SCRUM. Esta metodología esta basada en un proceso ágil y liviano que sirve para administrar y controlar el desarrollo de software.

El desarrollo se realiza en forma iterativa e incremental (una iteración es un ciclo corto de construcción repetitivo). Cada ciclo o iteración termina con una pieza de software ejecutable que incorpora una nueva funcionalidad. Las iteraciones en general tienen una duración entre 2 y 4 semanas.

Aunque surgió como modelo para el desarrollo de productos tecnológicos, también se emplea en entornos que trabajan con requisitos inestables y que requieren rapidez y flexibilidad; situaciones frecuentes en el desarrollo de determinados sistemas de software. (López, 2008)

Dentro de las principales características que podemos hallar de esta metodología están:

- ✓ Se focaliza en priorizar el trabajo en función del valor que tenga para el negocio, maximizando la utilidad de lo que se construye y el retorno de inversión.
- ✓ Está diseñado especialmente para adaptarse a los cambios en los requerimientos, por ejemplo en un mercado de alta competitividad.
- ✓ El equipo se centra en una única cosa: construir software de calidad.
- ✓ Los requerimientos y las prioridades se revisan y ajustan durante el proyecto en intervalos muy cortos y regulares.
- ✓ Se busca entregar software que realmente resuelvan las necesidades, aumentando la satisfacción del cliente.

2.2 Lenguajes.

2.2.1 JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. (Eguíluz Pérez)

Es un lenguaje que no requiere compilación, lo que se conoce como lenguaje de programación interpretado. Su sintaxis es muy parecida a la del lenguaje Java y el lenguaje C. Actualmente, todos los navegadores interpretan el código JavaScript integrado dentro de las páginas Web.

Por tradición, el lenguaje JavaScript se utilizaba para realizar tareas y operaciones en el marco de la aplicación únicamente del lado del cliente, sin acceso a funciones del servidor. Actualmente, existen aplicaciones en JavaScript para el servidor.

Aunque JavaScript de cliente y de servidor comparten el mismo conjunto base de funciones y características; en algunos casos se utilizan de distinta forma. De manera general, JavaScript permite crear aplicaciones específicamente orientadas a su funcionamiento en la red Internet. Usando JavaScript se pueden crear páginas HTML dinámicas que procesen la entrada del usuario y que sean capaces de gestionar datos persistentes usando objetos especiales, archivos y bases de datos relacionales.

Además, se pueden construir aplicaciones que varían desde la gestión de la información corporativa interna y su publicación en Intranets hasta la gestión masiva de transacciones de comercio electrónico.

2.2.2 PHP

PHP es el acrónimo de Hypertext Preprocessor, es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor pues funciona en un servidor remoto que procesa la página Web antes de que sea abierta por el navegador del usuario. PHP ha sido especialmente creado para el desarrollo de páginas Web dinámicas y puede ser incluido con facilidad dentro del código HTML.

Ha alcanzado gran popularidad y existe una amplia comunidad de desarrolladores y programadores que continuamente implementan mejoras en su código. Se caracteriza por una sencilla integración con múltiples bases de datos y, aunque MySQL es la base de datos que mejor trabaja con PHP, puede conectarse también a PostgreSQL, Oracle entre otras bases de datos.

Está dotado de un gran número de funciones predefinidas que simplifican enormemente tareas habituales como descargar documentos, generar imágenes GIF, enviar correos electrónicos, trabajar con cookies y sesiones, establecer conexiones a otros servicios de red y generar documentos PDF.

La sintaxis de PHP se basa en otros lenguajes de programación, principalmente en C y Perl, o un lenguaje de tipo C como C++ o Java, se distingue por su facilidad de aprendizaje y uso. Entre los competidores principales de PHP se puede citar a Perl, Microsoft Active Server Pages (ASP), Java Server Pages (JSP) y Allaire ColdFusion. PHP también soporta el uso de servicios que usen protocolos como IMAP, SNMP, NNTP, POP3, HTTP y derivados. Y además puede interactuar con otros protocolos. (Thomson y Welling, 2003)

PHP es una lenguaje multiplataforma, permite leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ser ingresados por los usuarios desde formularios HTML, es libre por lo que es una alternativa de fácil acceso para todos, es Orientado a Objetos y además posee una arquitectura extensible. Dentro de las opciones que brinda PHP se encuentra: la simplicidad, la estabilidad y la compatibilidad.

2.2.3 UML para el modelado.

Unified Modeling Language que se traduce al español como Lenguaje de Modelación Unificado, es un lenguaje gráfico para especificar, construir, visualizar y documentar las partes o artefactos (información que se utiliza o produce mediante un proceso de software).

Comprende el desarrollo de software que se basen en el enfoque OO (Orientado a Objetos), utilizándose también en el diseño Web, usa procesos de otras metodologías, aprovechando de esta manera la experiencia de sus creadores, además eliminó los componentes que resultaban de poca utilidad práctica y añadió nuevos elementos. UML se ha convertido en el estándar tan ansiado para representar y modelar la información con la que se trabaja en las fases de análisis y, especialmente, de diseño.

Una de las principales ventajas que brinda UML es que ayuda al usuario a entender la realidad de la tecnología y le posibilidad reflexionar antes de invertir y gastar grandes cantidades en proyectos que no

estén seguros en su desarrollo, reduciendo el coste y el tiempo empleado en la construcción de las piezas que constituyen el modelo.

Algunas de las características de UML como lenguaje de modelado estándar son:

- ✓ Reemplaza a decenas de notaciones empleadas con otros lenguajes.
- ✓ Modela estructuras complejas.
- ✓ Las estructuras más importantes que soportan tienen su fundamento en las tecnologías orientadas a objetos, tales como objetos, clase, componentes y nodos.
- ✓ Comportamiento del sistema: casos de uso, diagramas de secuencia y de colaboraciones, que sirven para evaluar el estado de las máquinas.

2.3 Sistemas de gestión de base de datos.

Un Sistema de Gestión de Bases de Datos (SGBD) es una herramienta que permite, mediante procedimientos o lenguajes, utilizar o actualizar datos almacenados más o menos permanentemente en una computadora, los que organizados y relacionados entre sí, constituyen una base de datos.

Un Sistema Gestor de Bases de Datos debe permitir definir una base de datos: especificar tipos, estructuras y restricciones de datos; construir la base de datos, es decir, guardar los datos en algún medio controlado por el mismo SGBD y manipular la base de datos: realizar consultas, actualizarla y generar informes. (Collector, 2004)

Para realizar la interconexión entre un SGBD y una aplicación Web es necesario utilizar un lenguaje de programación en el lado del servidor y para definir los datos y las estructuras, así como para hacer las consultas sobre los datos, el SQL (Structured Query Language), algo así como un lenguaje estructurado de consultas que su uso es estándar para la comunicación entre las aplicaciones y los SGBD.

Las consultas son una combinación de instrucciones que son transferidas desde el código de la aplicación Web hasta el SGBD utilizado para actualizar y manipular las bases de datos. Entre los sistemas de

gestión de base de datos comúnmente utilizados en el mundo está Oracle, MySQL, Microsoft SQL Server, PostgreSQL, InterBase, entre otros.

2.3.1 PostgreSQL

Es el SGBD de código abierto más avanzado en la actualidad, ofreciendo control de concurrencia multi-versión, soportando casi toda la sintaxis SQL (incluyendo sub-consultas, transacciones, tipos y funciones definidas por el usuario). Cuenta con un amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, C#, Java, Perl, PHP entre otros).

Dentro de las principales características se encuentran:

- ✓ Es un servidor de base de datos relacional, libre. Tiene soporte total para transacciones, disparadores, vistas, procedimientos almacenados, almacenamiento de objetos de gran tamaño.
- ✓ Permite la definición de tipos de datos personalizados e incluye un modelo de seguridad completo.
- ✓ PostgreSQL tiene una gran comunidad de desarrollo en Internet, su código fuente está disponible sin costo alguno y algo muy importante es que esta herramienta es multiplataforma.
- ✓ Soporta transacciones y desde la versión 7.0, claves ajenas con comprobaciones de integridad referencial.
- ✓ Tiene mejor soporte para vistas y procedimientos almacenados en el servidor, además tiene ciertas características orientadas a objetos.

La mayor limitación de PostgreSQL viene dada por su velocidad: es el sistema de bases de datos más lento, sobrecarga bastante al sistema más que MySQL.

2.3.2 MySQL

Es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. El servidor MySQL está diseñado para entornos de producción críticos con alta carga de trabajo así como para integrarse en software para ser distribuido. (MySQL, 2007)

Dentro de las principales ventajas que posee se encuentran:

- ✓ Soporta la mayoría de los comandos del lenguaje SQL (structured query language), el estándar en bases de datos.
- ✓ Es uno de los servidores de bases de datos más rápido y el de menor precio.
- ✓ Acceso a las bases de datos de forma simultánea por varios usuarios y/o aplicaciones.
- ✓ Garantiza la seguridad en forma de permisos y privilegios, determinados usuarios tendrán permiso para consulta o modificación de determinadas tablas, lo que permite compartir datos sin que peligre la integridad de la base de datos.
- ✓ Alta portabilidad, pues SQL es también un lenguaje estandarizado, de modo que las consultas hechas usando SQL son fácilmente portables a otros sistemas y plataformas.
- ✓ MySQL está escrito en C y C++ y probado con multitud de compiladores y esta disponible para muchas plataformas diferentes.
- ✓ Permite conexiones entre diferentes clientes con distintos sistemas operativos.

Este sistema gestor de bases de datos se distribuye bajo los términos de la licencia pública general (GPU), la misma plantea que MySQL es gratis incluso para su uso comercial mientras trabaje como servidor de web, pero si se desea trabajar con otras aplicaciones será entonces necesario obtener una licencia.

2.4 Herramientas.

2.4.1 Zend Studio

Zend Studio es un completo entorno integrado de desarrollo para el lenguaje de programación PHP. Está escrito en Java, y está disponible para las plataformas Microsoft Windows, Mac OS X y GNU/Linux.

Zend Studio consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. Las dos partes se instalan por separado, la del cliente contiene el interfaz de edición y la ayuda. Permite además hacer depuraciones simples de scripts, aunque para disfrutar de toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor, que instala Apache y el módulo PHP o, en caso de que estén instalados, los configura para trabajar juntos en depuración. (Álvarez, 2003)

Dentro de las principales características que provee se encuentran:

- ✓ No requiere la instalación previa de PHP ni del entorno de ejecución de Java.
- ✓ Soporte para PHP 4 y PHP 5.
- ✓ Resaltado de sintaxis, autocompletado de código, ayuda de código y lista de parámetros de funciones y métodos de clase.
- ✓ Inserción automática de paréntesis y corchetes de cierre.
- ✓ Sangrado automático y otras ayudas de formato de código.
- ✓ Detección de errores de sintaxis en tiempo real.
- ✓ Funciones de depuración.
- ✓ Soporte para gestión de grandes proyectos de desarrollo.
- ✓ Manual de PHP integrado.
- ✓ Cliente FTP integrado.
- ✓ Soporte para navegación en bases de datos y ejecución de consultas SQL.

Zend Studio fue diseñado para usarse con el lenguaje PHP; sin embargo ofrece soporte básico para otros lenguajes Web, como HTML.

2.4.2 Dreamweaver

Es la herramienta de diseño de páginas Web más avanzada, tal como se ha afirmado en muchos medios. Aunque sea un experto programador de HTML el usuario que lo maneje, siempre se encontrará en este programa razones para utilizarlo, resaltando todo en lo que a productividad se refiere.

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. El panel de comportamientos también permite crear JavaScript básico sin conocimientos de código.

Cumple perfectamente el objetivo de diseñar páginas con aspecto profesional, y soporta gran cantidad de tecnologías, además muy fáciles de usar:

- ✓ Hojas de estilo y capas
- ✓ Javascript para crear efectos e interactividades
- ✓ Inserción de archivos multimedia

Además es un programa que se puede actualizar con componentes, que fábrica tanto Macromedia como otras compañías, para realizar otras acciones más avanzadas

2.3.4 Rational Rose Enterprise Edition

Rational Rose es una herramienta de diseño orientada a objetos, que da soporte al modelado visual, es decir, que permite representar gráficamente el sistema, permitiendo hacer énfasis en los detalles más importantes, centrándose en los casos de uso y enfocándose hacia un software de mayor calidad, empleando un lenguaje estándar común que facilita la comunicación.

Esta herramienta propone la utilización de cuatro tipos de modelo para realizar el diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software. Esta herramienta tiene la desventaja de ser un software con licencia privativa y solo funciona sobre sistemas operativos Microsoft Windows.

Dentro de las características que ofrece esta herramienta se encuentran:

- ✓ Soporte para análisis de patrones ANSI, C++, Rose J y Visual C++
- ✓ Característica de control por separado de componentes
- ✓ La generación de código Ada, ANSI C ++, C++, CORBA, Java y Visual Basic, con capacidad de sincronización modelo - código configurables.
- ✓ Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos.
- ✓ Capacidad para integrarse con cualquier sistema de control de versiones SCC, incluyendo a Rational ClearCase.
- ✓ Ingeniería Inversa
- ✓ Trabajo en grupo

- ✓ Desarrollo iterativo. (Rational, 2007)

2.5 Arquitectura Cliente-Servidor.

La arquitectura cliente-servidor, llamada modelo cliente - servidor es una forma de dividir y especializar programas y equipos de cómputo a fin de que la tarea que cada uno de ellos realiza se efectué con la mayor eficiencia y permita simplificar las actualizaciones y mantenimiento del sistema. (Usero Martínez, 2007)

Para poder entender mejor esta arquitectura se explica en que consiste el término cliente y servidor.

Cliente

Si se considera que el usuario, a través de una computadora local correspondiente al cliente, es el interesado en interactuar con los programas que existen en Internet, el cliente tiene como función primordial facilitar la interacción del usuario. (Weitzenfeld, 2004)

En otras palabras, el objetivo básico del cliente en la arquitectura cliente-servidor es facilitar la presentación y control de la información administrada por la aplicación. Por tanto, la mayoría de las tecnologías que se procesan en el cliente están dirigidas a facilitar la visualización y control de la información, como es el caso de HTML, Flash, Javascript, VBScript, JScript.

Servidor.

El servidor es un programa que recibe una solicitud, realiza el servicio requerido y devuelve los resultados en forma de una respuesta. Generalmente un servidor puede tratar múltiples peticiones (múltiples clientes) al mismo tiempo. Al proceso servidor se le conoce con el término back-end. (Weitzenfeld, 2004)

Históricamente, el primer modelo de programación para arquitecturas cliente-servidor en Internet fueron los CGI (Common Gateway Interface), que en la actualidad se han extendido con estándares más modernos como ASP, PHP, JSP, etc.

El servidor cumple una serie de funciones como:

- ✓ Acepta peticiones recibidas a través de la red.
- ✓ Realiza el servicio y regresa el resultado al cliente.
- ✓ Gestiona los periféricos compartidos.
- ✓ Controla accesos concurrentes a bases de datos compartidas.

2.6 Tratamiento de seguridad en aplicaciones Web.

La seguridad de las aplicaciones Web es hoy en día es una prioridad. Estas aplicaciones contienen y manejan información que puede ser de gran valor, por eso la necesidad de protegerla, pues otras personas pueden utilizarla para causar daño sin importar las consecuencias.

La seguridad de la información que se almacena en el servidor Web es una de las preocupaciones de los programadores, al realizar un software se debe tratar de asegurar la operación continua del servidor, que los datos no sean modificados sin la debida autorización y que la información solo sea distribuida a las personas autorizadas.

Para un manejo adecuado de los datos se debe contar con navegadores y plataformas seguras, libres de virus y vulnerabilidades, además garantizar que la información que viaja por la red no sea leída, modificada o destruida por terceros y que el enlace entre cliente - servidor no sea interrumpido fácilmente.

2.6.1 Principales ataques a las aplicaciones Web.

Según la clasificación hecha por Open Web Application Security Project (OWASP) una de las organizaciones independientes más importantes en Seguridad de la Información, los principales ataques a una aplicación Web son:

1. *Cross Site Scripting* se refiere a aquellos ataques que permiten al hacker obtener información confidencial del usuario (como cuentas y contraseñas), llevar a cabo ataques de phisshing y hasta llegar a tomar el control del navegador del usuario final. Esta es la vulnerabilidad más común y agresiva de las aplicaciones Web.

2. *Inyección defectuosa*, este tipo de vulnerabilidades tiene lugar cuando un usuario malicioso ejecuta comandos no previstos en la aplicación Web mediante el envió de datos especialmente diseñados para tal fin utilizando como intérprete de estos comandos al sitio, estas instrucciones permiten al atacante leer, crear o borrar información. Algunos tipos de instrucciones que se inyectan en las aplicaciones son de tipo SQL, LDAP, XML o comandos de sistema operativo.

Todas las plataformas son susceptibles a cualquiera de estos tipos de ataque, ya sea C#, .Net, PHP, C, Perl, Ruby on Rails y hasta aplicaciones Java y lo peor de todo es que no importa que tan segura sea la configuración de red, si la aplicación no es segura, la información que contiene está en riesgo. Por eso tomar medidas precautorias para evitar estos tipos de ataques es garantía de que ni las organizaciones ni los usuarios sufran algún tipo de pérdida.

2.6.2 ¿Cómo desarrollar aplicaciones Web seguras?

Para desarrollar una aplicación Web segura se debe tener en cuenta la autenticación de usuarios. La autenticación es un proceso que permite verificar la identidad digital del remitente de comunicación digital.

La autenticación se puede realizar de tres formas.

1. Utilizando autenticación con htaccess de Apache.
2. Utilizando autenticación http con PHP.
3. Manejando sesiones.

Utilizando autenticación con htaccess de Apache.

Se crea un archivo denominado htaccess, pero no es la forma mas adecuada de proteger archivos y carpetas de un servidor.

Utilizando autenticación http con PHP.

La autenticación básica con PHP se basa en el uso de variables de entorno cuyos valores se obtienen del servidor Web. Existen diferentes formar de autenticar usuarios desde un script PHP. Se puede hacer la autenticación directamente en el código utilizando usuarios y contraseñas estáticas, o utilizando un archivo de contraseñas alojados en un servidor, pero la forma más segura es utilizando usuarios y

contraseñas almacenados en una base de datos, este método exige que se aplique la autenticación a cada página a la que el usuario tenga acceso para garantizar que este accediendo el mismo usuario.

Manejando sesiones.

La necesidad de las sesiones surge de la naturaleza del protocolo http, que es un protocolo sin estado, lo que significa que no dispone de un método incorporado para conservar el estado entre dos transacciones. Esto provoca que cuando el usuario solicita una página y luego otra no exista forma que el servidor entienda que las dos solicitudes proviene de un mismo usuario, de modo que todas las variables de un script son restablecidas siempre después de una solicitud.

Definición de sesiones.

Se puede definir una sesión como el tiempo que un usuario permanece conectado a un sitio Web, esto es de forma simple pero de forma más técnica y relacionada con la programación del lado del servidor, una sesión es un bloque de información que almacena todo tipo de variables y valores relacionados con los usuarios y sus visitas a un sitio en particular (Domínguez, 2008)

El control de la sesión consiste en poder realizar un seguimiento al usuario mientras se mantenga interactuando con un sitio Web, permitiendo mostrar contenido de las páginas en función de su nivel de autorización o de sus preferencias personales.

Encriptación de contraseñas

Como la autenticación será mediante usuario y contraseña entonces se debe almacenar las contraseñas en la bases de datos de forma segura. Aunque la información que se almacena en una base de datos no es del todo segura, para asegurar las contraseñas se usa la encriptación de contraseñas de manera que si alguien puede acceder a ellas no pueda ver la contraseña sino su encriptación. Existen muchos algoritmos para la encriptación de contraseñas como por ejemplo el MD2, MD4, SHA, Tigre, RIPEMD -160, WHIRLPOOL, CRC, Hash entre otros pero específicamente en el sistema para generar Planes de Seguridad Informática se usa el MD5.

MD5 (Algoritmo de Resumen del Mensaje 5) es un algoritmo de reducción criptográfico de 128 bits ampliamente usado, que se usa para la encriptación de un solo sentido, es decir que no se puede desencriptar de ninguna manera.

Seguridad en las Bases de Datos.

Las bases de datos son componentes muy importantes en las aplicaciones Web pues estas almacenan contenido que puede ser accedido dinámicamente para su modificación y eliminación, además la información almacenada en las mismas puede ser sensible o secreta y es por eso que se deben tomar medidas extremas para protegerlas. No podemos nunca decir que una base de datos es invulnerable porque la seguridad de esta depende de muchos factores.

Son muchas las vulnerabilidades que se pueden presentar en una base de datos como por ejemplo: acceso no autorizado, corrupción de datos, inyección SQL, en ocasiones la comunicación entre la aplicación y la base de datos no usa cifrado o no existe un control adecuado en el control de acceso de bases de datos, también existen múltiples usuarios con varios permisos de acceso sobre los objetos como tablas, vistas procedimientos almacenados, una forma de combatir todas estas debilidades es aplicar correctamente el conocido principio de mínimo privilegio en la configuración de accesos, es decir, conceder los permisos solo a la persona que lo necesite.

Uno de los ataques más frecuentes en una base de datos es la inyección SQL, se debe aplicar una serie de controles para reducir el riesgo al que conlleva este ataque, se debe validar las entradas de los datos, uso de SQL parametrizado tanto en sentencias SQL como en llamadas a procedimientos almacenados, uso de privilegio mínimo en las cuentas que accedan a la base de datos y eliminación de todas las posibilidades de ejecución de código SQL innecesarias y control correcto de errores que provengan de la base de datos. Todos estos controles deben aplicarse conjuntamente para que la base de datos sea lo más robusta posible. (Rodríguez, 2006)

2.7 Metodología de desarrollo, herramientas y lenguaje seleccionados.

Como metodología de desarrollo a utilizar se seleccionó la metodología RUP, la misma brinda grandes ventajas ya que es un proceso muy documentado, está centrado en la arquitectura y es una de las

metodologías más usadas en el desarrollo de software actual y siguiendo los pasos propuestos por esta metodología se obtiene una buena documentación de la tesis; empleándose como herramienta case el Rational Rose 2003 que es compatible con la metodología utilizada, además se eligió para el modelado del sistema UML por tener gran integración con la metodología RUP y la herramienta Rational Rose Enterprise Edition.

Como lenguaje de programación Web se escogió PHP 5.0, este brinda muchas ventajas, es de fácil acceso, es libre y de mucha utilidad en las aplicaciones Web en el mundo, además para la ejecución de errores se utiliza JavaScript 2.0 evitando que el servidor Web procese la página en vano. Para la inclusión del código PHP y JavaScript se usa el Zend Studio, versión 6.1. El diseño de la arquitectura cliente\servidor de este IDE permite correr múltiples clientes para cada uno de los programadores de un gran proyecto desde un único servidor. Para el diseño total del software se seleccionó la herramienta Dreamweaver 8. El servidor Web será Apache, en su versión 2.0.

Como sistema gestor de bases de datos se emplea MySQL 5.0 ya que tiene gran vinculación con Apache Server y PHP, pues la relación de estos reúne una serie de ventajas como permitir crear aplicaciones con alta portabilidad debido a que son tecnologías multiplataforma, además se crean aplicaciones altamente seguras.

3
CARACTERÍSTICAS DEL SISTEMA.

En el presente capítulo se realiza una descripción del negocio, se definen actores y trabajadores del negocio, se representan los casos de uso del negocio y los diagramas de actividades correspondientes a cada caso de uso. También se detallan los requisitos funcionales y no funcionales que debe cumplir el sistema, lo que permite realizar una concepción general del mismo y representar en el diagrama de casos de uso del sistema, los actores del sistema y sus relaciones además se realiza la descripción textual de los casos de uso del sistema.

3.1 Modelo del negocio.

El primer paso para la automatización de un sistema de software es comprender los procesos que en ella se desarrollan para lograr un mejor entendimiento del problema a resolver. El modelo del negocio es un modelo de casos de uso que describe los procesos en términos de casos de uso y actores.

3.1.1 Descripción de los procesos del negocio y mejoras propuestas.

La realización del Plan de Seguridad Informática tiene como objetivo principal abordar todo lo relacionado con la Seguridad Informática, así como un conjunto de políticas, medidas y procedimientos que son necesarias para garantizar la confiabilidad, integridad y disponibilidad de la información que se procesa, intercambia, produce y conserva en los activos informáticos pertenecientes a la entidad a la cual se realiza el Plan de Seguridad Informática.

El Plan de Seguridad Informática cuenta con dos procesos fundamentales, el primero es el proceso de realizar el Análisis de Riesgo donde el Director de Seguridad Informática le solicita al Especialista de Análisis de Riesgo que realice la estimación de riesgos. Para realizar el análisis de riesgo es necesario que el Responsable de Control de Inventario le entregue al Especialista de Análisis de Riesgo un listado con todos los activos que existen en la organización. Luego se siguen una serie de pasos (ver Anexo 1). Luego de terminado este proceso se le entrega al Director de Seguridad Informática.

Después de tener el resultado del análisis de riesgo donde se identifican las situaciones que hacen vulnerable el sistema, se establecen las Políticas, Medidas y Procedimientos a seguir en la organización para contrarrestar estas vulnerabilidades y se elabora el Plan de Contingencia, este paso es realizado por el Especialista de Seguridad Informática, el cual elabora un documento general donde realiza una caracterización de la organización, recoge el resultado del Análisis de Riesgo, las Políticas, Medidas, Procedimientos y el Plan de Contingencia y de esta forma queda conformado el Plan de Seguridad Informática.

La situación problémica esta dada debido a que estos planes de seguridad son muy extensos por la cantidad de información que poseen y elaborados manualmente lo que atenta contra la calidad del mismo y en ocasiones se obvian pasos que son muy importantes.

El sistema que se propone contiene una serie de mejoras, pues se debe realizar una herramienta que guíe los pasos para la realización de los Planes de Seguridad Informática y esto tiene como ventaja que no se obvie ningún paso para la elaboración del mismo y que se agilice el proceso de realizar el Plan.

Además se le propondrá al usuario listados predefinidos de activos y amenazas para que escoja las que necesite y se le dará la posibilidad de entrar otras si lo desea, también brinda la posibilidad de calcular la importancia de los activos y la estimación de riesgos lo que permite que el proceso sea más exacto y no haya errores de cálculo.

3.1.2 Actores del negocio.

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. El actor del negocio interactúa con el negocio para beneficiarse de sus resultados. Después de hacer el análisis correspondiente se encontró el actor siguiente:

Actor del negocio	**Descripción**
Director de Seguridad Informática	Es el que luego de elaborado el Plan de Seguridad Informática se encarga de revisar y aprobar el documento.

3.1.3 Trabajadores del negocio.

El trabajador del negocio 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. Se han determinado como trabajadores del negocio los siguientes:

Trabajador del negocio	Descripción
Especialista de Seguridad Informática	Son las personas que designa el Director de Seguridad Informática para que elaboren el Plan de Seguridad Informática.
Especialista de Análisis de Riesgo	Es el encargado de realizar el proceso de Análisis de Riesgo del Plan de Seguridad Informática.
Responsable de Control de Activos	Es el que lleva el control de todos los activos existentes en la entidad.

3.1.4 Diagrama de Casos de Uso del Negocio.

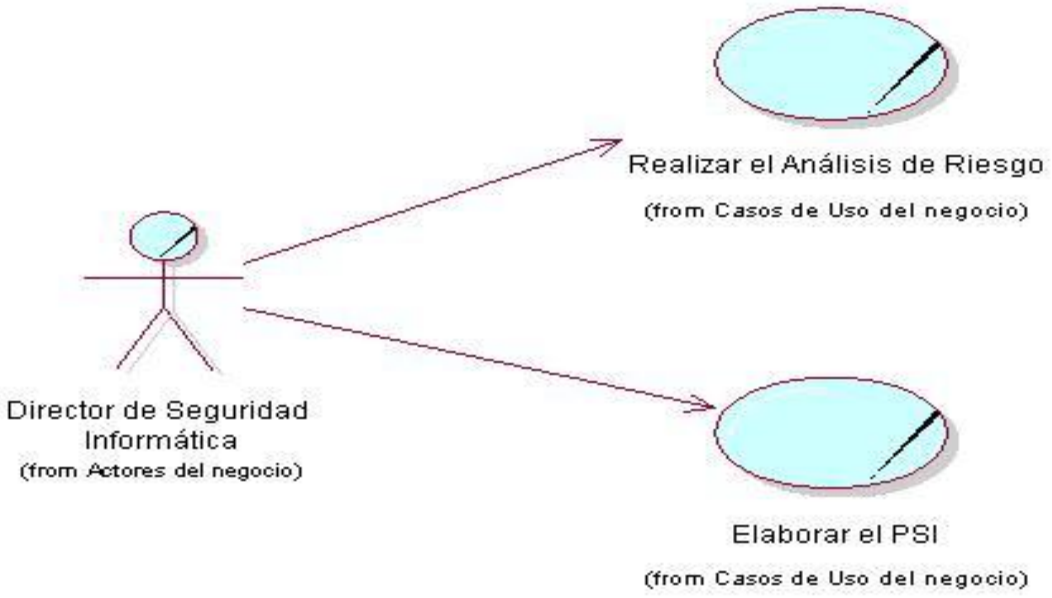


Figura 3.1 Diagrama de Casos de Uso del Negocio.

3.1.5 Descripción textual de los casos de uso del negocio.

La descripción de los caso de uso de negocio constituye un artefacto correspondiente al flujo de trabajo Modelado del Negocio de RUP llamado “Modelo de Casos de Uso del Negocio”. (Ver Anexo 2 y 3)

3.1.6 Diagramas de actividades para cada caso de uso del negocio.

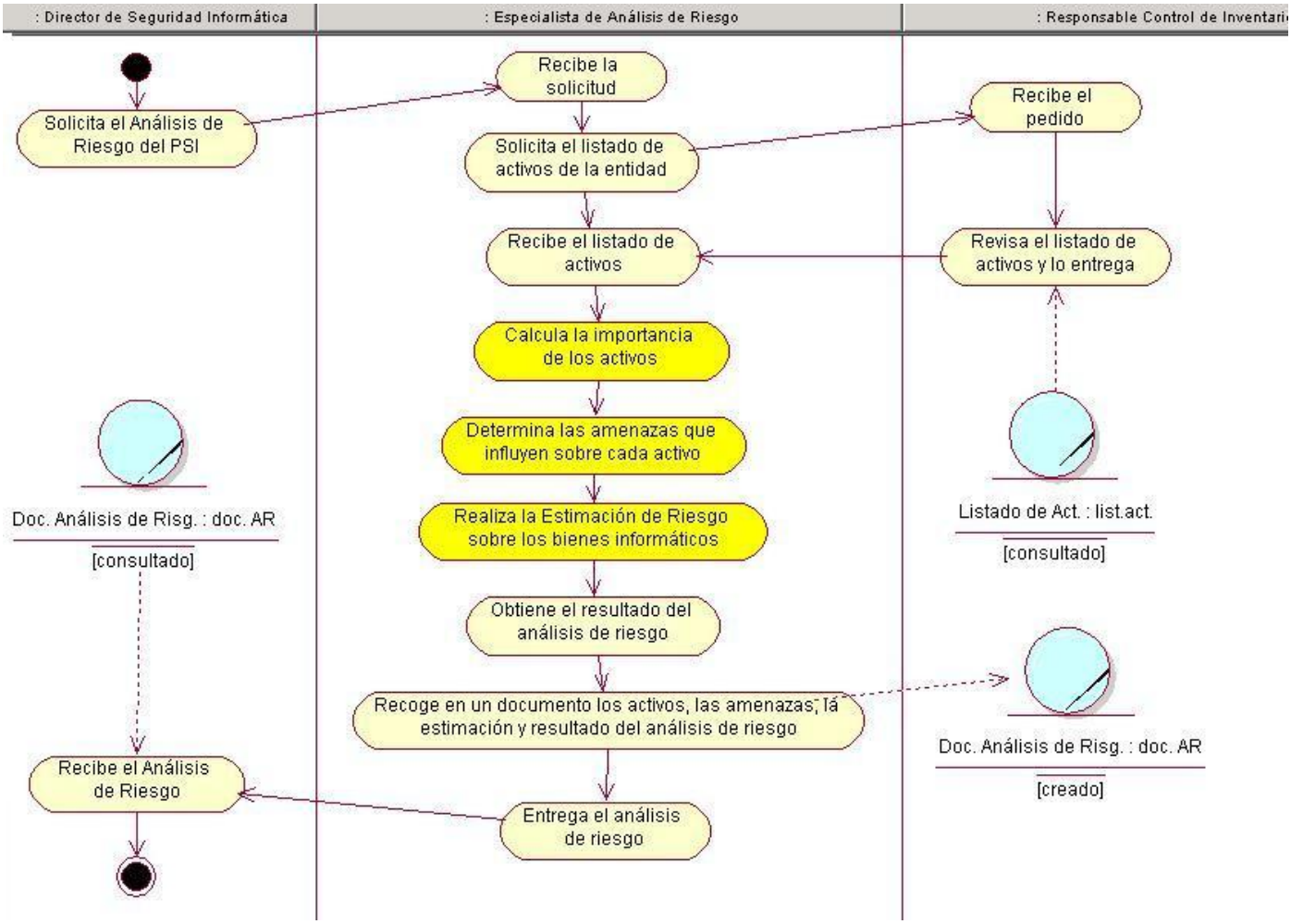


Figura 3.2 Diagrama de Actividades del Caso de uso “Realizar el Análisis de Riesgo”.

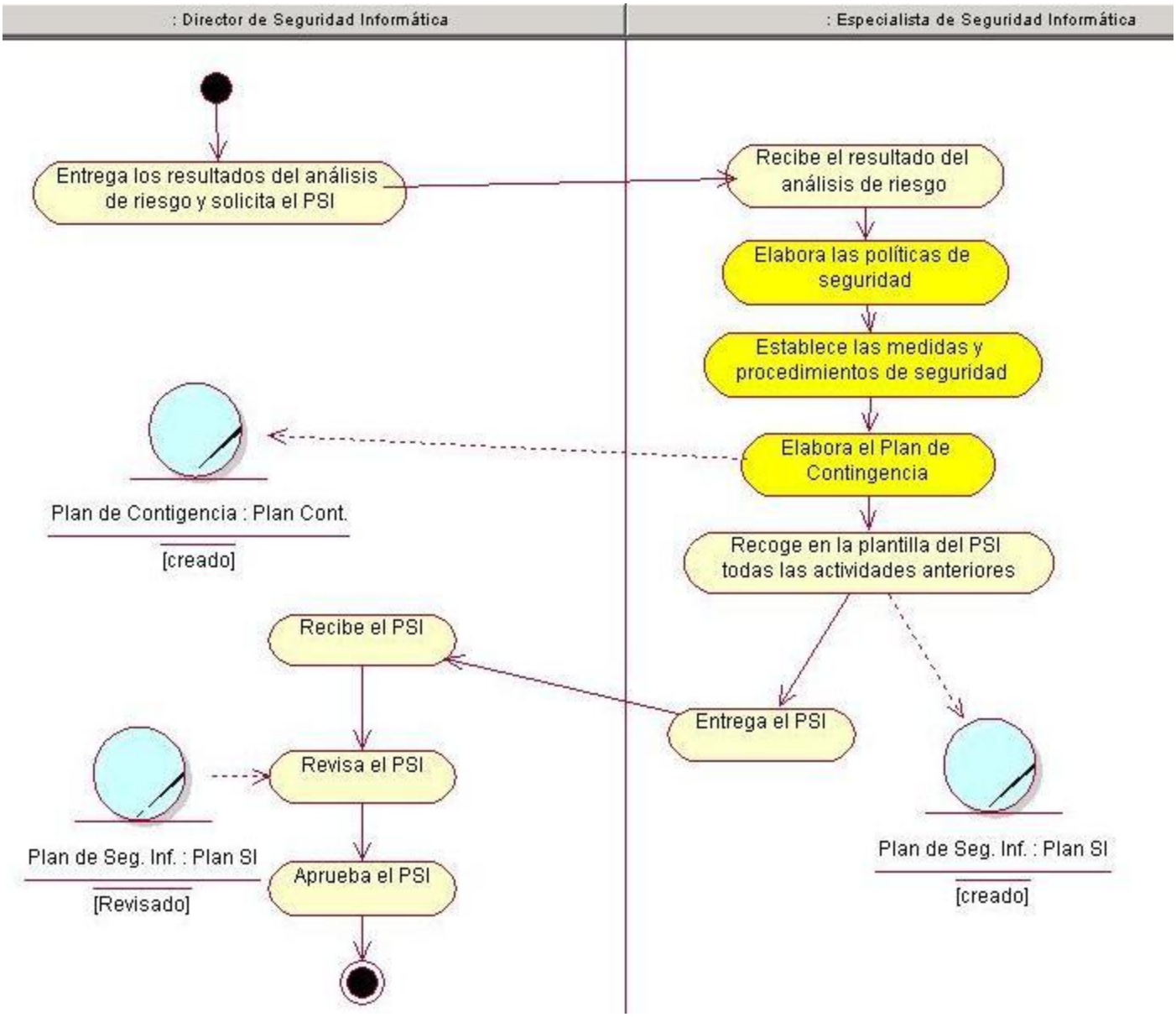


Figura 3.3 Diagrama de Actividades del Caso de uso "Elaborar el Plan de Seguridad Informática".

3.1.7 Diagrama de clases del Modelo Objeto del Negocio.

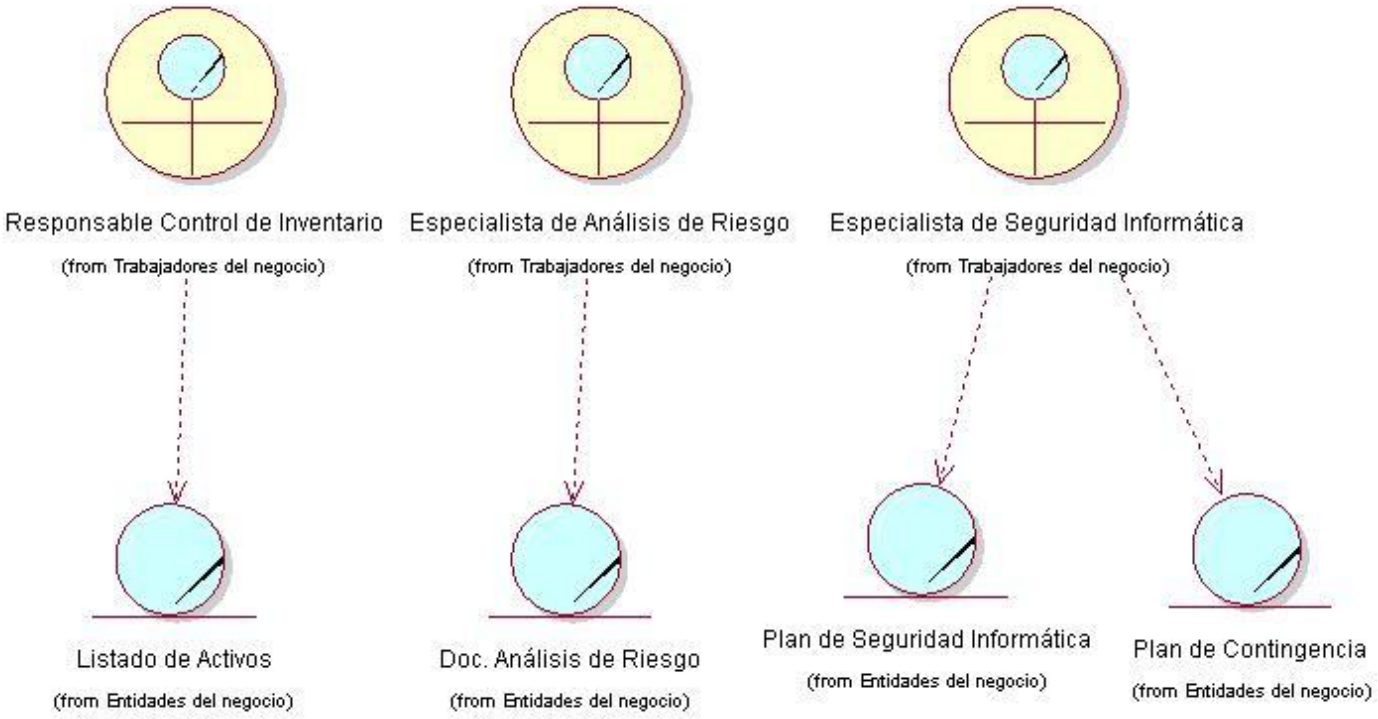


Figura 3.4 Diagrama de clases del Modelo Objeto del Negocio.

3.2 Especificación de los Requisitos Funcionales y No funcionales.

La captura de requisitos se realiza con el objetivo de que el sistema cumpla con las exigencias del cliente y los usuarios finales. El objetivo de la captura de requisitos es guiar el desarrollo del software hacia la correcta solución, definiendo objetivos concretos de manera que tanto el negocio como sus actores se beneficien.

3.2.1 Requisitos Funcionales.

Los requisitos funcionales son condiciones o capacidades que el sistema debe cumplir. En este trabajo se definieron los siguientes requisitos funcionales. El sistema debe permitir:

RF 1. Autenticar Usuario.

RF 2. Gestionar usuarios locales.

RF 2.1 Adicionar un nuevo usuario.

RF 2.2 Modificar datos del usuario.

RF 2.3 Eliminar usuarios.

RF 3. Gestionar usuarios Dominio UCI.

RF 3.1 Adicionar un nuevo usuario.

RF 3.2 Modificar datos del usuario.

RF 3.3 Eliminar usuarios.

RF 4. Gestionar Caracterización.

RF 4.1 Adicionar Caracterización.

RF 4.2 Modificar Caracterización.

RF 5. Gestionar Activos Informáticos.

RF 5.1 Adicionar nuevos activos informáticos.

RF 5.1 Modificar activos informáticos.

RF 5.3 Eliminar activos informáticos.

RF 6. Seleccionar Activos Informáticos.

RF 7 Gestionar Dominio.

RF 7.1 Adicionar nuevo dominio.

RF 7.2 Modificar dominio.

RF 7.3 Eliminar dominio.

RF 8 Importancia de los activos.

RF 8.1 Calcular el valor de la importancia de los activos.

RF 9. Adicionar Amenazas.

RF 10. Determinar amenazas sobre Activos.

RF 11. Realizar Estimación de Riesgo.

RF 11.1 Calcular la valoración del riesgo sobre cada bien informático.

RF 11.2 Calcular el Peso Relativo de Riesgo sobre cada bien informático.

RF 11.3 Calcular el Peso Total de Riesgo del sistema.

RF 12. Gestionar Políticas.

RF 12.1 Adicionar nuevas políticas de seguridad.

RF 12.2 Modificar Políticas.

RF 12.3 Eliminar Políticas.

RF 13. Gestionar Medidas de Seguridad.

RF 13.1 Adicionar nuevas medidas de seguridad.

RF 13.2 Modificar medidas.

RF 13.3 Eliminar medidas.

RF 13. Gestionar Procedimientos de Seguridad.

RF 14.1 Adicionar nuevos procedimientos de seguridad.

RF 14.2 Modificar procedimientos.

RF 14.3 Eliminar procedimientos.

RF 15. Gestionar áreas.

RF 15.1 Adicionar nueva área.

RF 15.2 Modificar área.

RF 15.3 Eliminar área.

RF 16. Revisar y Aprobar plan de seguridad informática.

RF 17. Gestionar Directores.

RF 17.1 Adicionar Directores.

RF 17.2 Modificar Directores.

RF 17.3 Eliminar Directores.

FR 18. Cambiar Contraseña del administrador.

RF 19. Generar Plan de Seguridad Informática.

3.2.2 Requisitos no Funcionales.

Los requisitos no funcionales son cualidades o propiedades que el sistema debe tener. A continuación se han definido los mismos de acuerdo a su clasificación.

Requerimientos de software.

- ✓ Servidor Web Apache versión 2.2 o superior.
- ✓ Servidor de Base de Datos MySQL versión 5.0 o superior.
- ✓ Sistema Operativo Windows XP o superior.
- ✓ Navegadores Web: Mozilla Firefox versión 3.0 o superior. Internet Explorer versión 8.0 o superior.

- ✓ PHP versión 5.0 o superior.

Requerimientos de hardware.

Requerimientos mínimos para la máquina servidora.

- ✓ Procesador de tipo Pentium III con velocidad de 1 GHz o superior.
- ✓ Memoria RAM de 512 MB o superior.
- ✓ Disco duro de 20 GB o superior.
- ✓ Tarjeta de red o módem.

Requerimientos mínimos para la estación de trabajo del usuario.

- ✓ Procesador de tipo Pentium II con velocidad de 450 MHz o superior.
- ✓ Memoria RAM de 256 MB o superior.
- ✓ Disco duro de 4 GB o superior
- ✓ Tarjeta de red o módem.
- ✓ Impresora.

Requerimientos de apariencia o interfaz externa

- ✓ El sistema deberá contar con una interfaz sencilla, amigable y muy intuitiva, que a manera de asistente guíe al usuario por las etapas del proceso de creación del Plan de Seguridad Informática.
- ✓ Elementos organizativos para las funcionalidades como iconos y pestañas**.**
- ✓ Funcionalidades visibles en todo momento que faciliten la navegación.

Requerimientos de seguridad.

- ✓ Sistema de permisos y usuarios para el acceso a la información, para protegerla de accesos no autorizados.
- ✓ El sistema deberá contar con 2 tipos de usuarios: Los usuarios locales del sistema que serán los que estén predefinidos por defecto y los usuarios del dominio UCI que predefinan los usuarios locales.
- ✓ El sistema deberá estar orientado a mantener la integridad, disponibilidad y confidencialidad de los datos.
- ✓ El sistema deberá garantizar que las contraseñas viajen encriptadas por la red.

- ✓ El sistema deberá garantizar la interacción sin afectar la seguridad de la base de datos, mediante su despliegue separadamente del servidor de aplicaciones.

Requerimientos de usabilidad.

- ✓ Diseño orientado a la simplicidad y una clara arquitectura de la información que permita que el sistema pueda interactuar con cualquier tipo de persona que posea conocimientos básicos de aplicaciones Web.

Requerimientos de rendimiento.

- ✓ La herramienta propuesta debe ser eficiente, rápida y el tiempo de respuesta debe ser el mínimo posible y está implementado sobre una tecnología Web, facilitando su uso a través de la red.

3.3 Definición de casos de uso del sistema.

3.3.1 Actores del sistema.

Actor del Sistema	**Descripción.**
Director de Seguridad Informática.	Es el encargado de asignar permisos a los usuarios, revisar y aprobar el PSI, realizar la gestión de usuarios además puede realizar el PSI.
Especialista de análisis de riesgo.	Es el que realiza el análisis de riesgo del PSI, también puede realizar el PSI completo.
Responsable de control de Inventario.	Es el encargado de gestionar los activos informáticos.
Responsable de Políticas	Es el encargado de establecer las políticas de seguridad.
Ingeniero de Sistemas.	Es el que elabora las medidas y procedimientos de seguridad.
Responsable de Áreas.	Se encarga de la gestión de las áreas.
Administrador	Responsable de adicionar los directores a las áreas

	y puede cambiar su contraseña.

3.3.2 Diagrama de casos de uso del sistema.

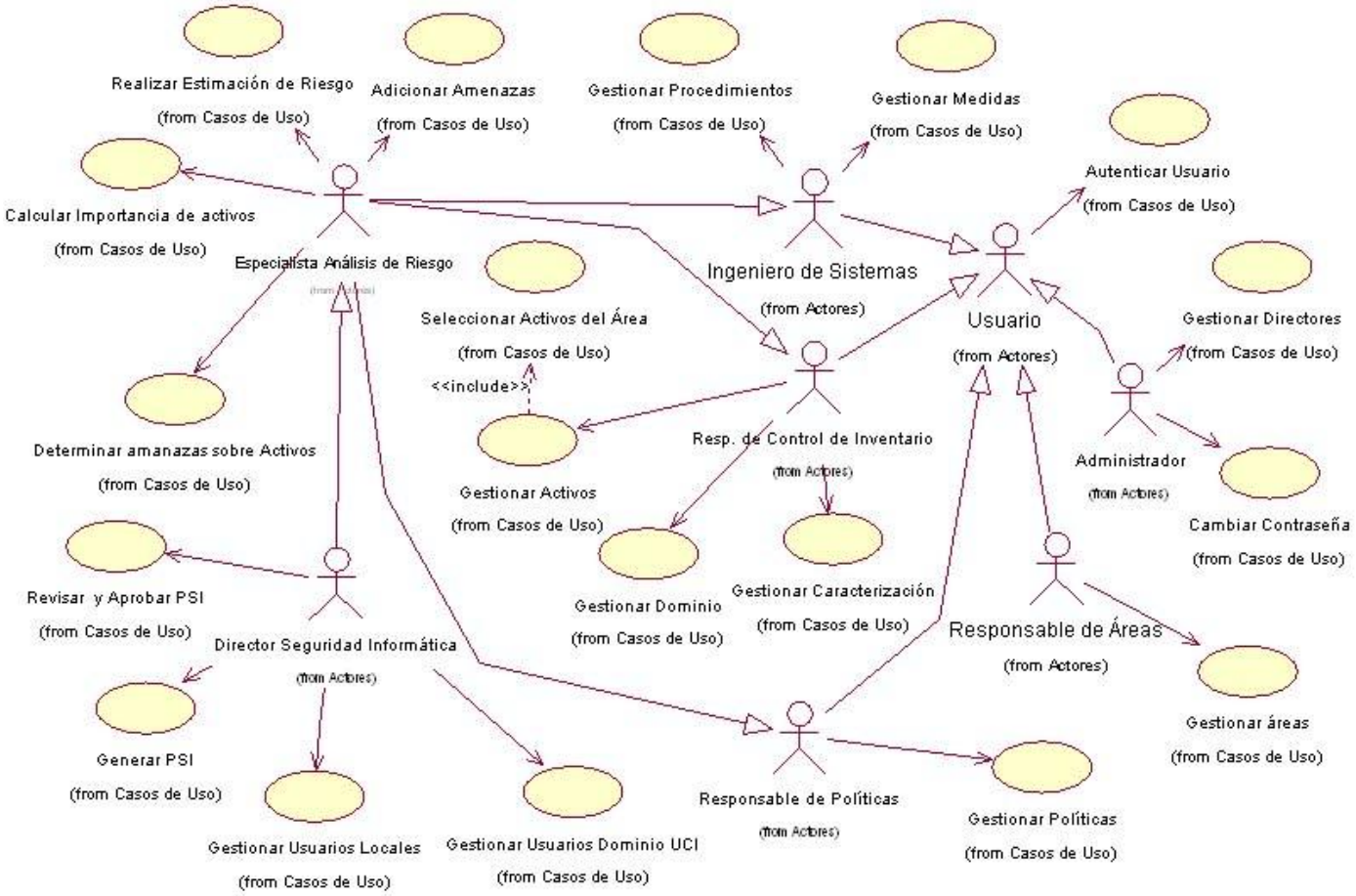


Figura 3.5 Diagrama de Casos de Uso del Sistema.

3.3.3 Descripción textual de los casos de uso del sistema.

Descripción textual CU "Autenticar Usuario".

Caso de Uso:	Autenticar Usuario
Actores:	Responsable de control de Inventario, Especialista de análisis de riesgo, Director de Seguridad Informática, Responsable de Áreas, Ingeniero de Sistemas, Responsable de Políticas

Resumen:	El caso de uso se inicia cuando un Usuario necesita acceder al sistema, se comprueba la validez de los datos del Usuario que se autentica y se le brindan los permisos según el rol que tenga asignado en el sistema.
Precondiciones:	El Usuario debe tener permisos para interactuar con el sistema.
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El Usuario escribe nombre y contraseña en el formulario para acceder al sistema.	2. El sistema verifica la valides de los datos. 3. Muestra la página según el rol que tenga asignado en el sistema.
Flujo Alterno	
Acción del Actor	**Respuesta del Sistema**
	4. Si los datos no son validos muestra un mensaje de error.
Referencias	RF1

Descripción textual CU "Gestionar Usuarios Locales".

Caso de Uso:	Gestionar Usuarios Locales
Actores:	Director de Seguridad Informática
Resumen:	El caso de uso se inicia cuando el Director de Seguridad Informática necesita insertar, modificar o eliminar un usuario local al sistema. Finaliza el caso de uso ejecutadas una de las acciones anteriores.
Precondiciones:	-
Flujo Normal de Eventos	
Sección "Insertar Usuario Local"	
Acción del Actor	**Respuesta del Sistema**
1. Accede a insertar un usuario local. 3. Introduce los datos.	2. Muestra formulario para entrar datos del usuario. 4. Guarda la información en la base de datos. 5. Muestra mensaje de acción completada.

Precondiciones:	El usuario a modificar debe encontrarse en el sistema
Flujo Normal de Eventos	
Sección "Modificar Usuario Local"	
Acción del Actor	**Respuesta del Sistema**
1. Accede a modificar un usuario local. 3. Selecciona el usuario que desea modificar. 5. Realiza los cambios.	2. El sistema muestra un listado con todos los usuarios locales de su área. 4. Muestra formulario con los datos del usuario local. 6. Guarda los cambios realizados. 7. Muestra el usuario actualizado.
Precondiciones:	El usuario a eliminar debe encontrarse en el sistema
Flujo Normal de Eventos	
Sección "Eliminar Usuario Local"	
Acción del Actor	**Respuesta del Sistema**
1. Accede a eliminar un usuario local. 3. Selecciona el usuario que desea eliminar.	2. El sistema muestra un listado con todos los usuarios locales de su área. 4. Elimina el usuario de la base de datos. 5. Muestra el listado de usuarios actualizado.
Referencias	RF2.(RF2.1, RF2.2, RF2.3)

Descripción textual CU "Importancia de Activos".

Caso de Uso:	Importancia de Activos
Actores:	Director de Seguridad Informática, Especialista de Análisis de Riesgo
Resumen:	El caso uso se inicia cuando el Director de Seguridad Informática o el Especialista de Análisis de Riesgo van a calcular la importancia de los activos, el sistema muestra una tabla para introducir los valores de las columnas función, costo, imagen, confidencialidad, integridad, y disponibilidad en dependencia de la estimación que se haga de la importancia de cada uno de estos factores sobre

	los bienes informáticos analizados. El caso de uso termina cuando el sistema guarda los datos de la tabla y muestra el resultado de los cálculos.
Precondiciones:	Los activos deben haber sido seleccionados.
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. Accede a calcular la importancia de los activos. 3. Introduce los valores de las columnas función, costo, imagen, confidencialidad, integridad, y disponibilidad.	2. Muestra una tabla con los campos a llenar. 4. Verifica que no hayan campos vacíos. 5. Guarda los datos de la tabla en la base de datos. 6. Realiza los cálculos. 7. Muestra los resultados de los cálculos.
Flujo Alterno	
Acción del Actor	**Respuesta del Sistema**
	4. Si hay campos vacíos muestra mensaje de error.
Referencias	RF 8 (8.1)

Descripción textual CU "Gestionar Dominios".

Caso de Uso:	Gestionar Dominios
Actores:	Director de Seguridad Informática, Especialista de Análisis de Riesgo
Resumen:	El caso de uso se inicia cuando el Director de Seguridad Informática o Especialista de Análisis de Riesgo necesita insertar, modificar o eliminar un dominio al sistema. Finaliza el caso de uso ejecutadas una de las acciones anteriores.
Precondiciones:	-
Flujo Normal de Eventos	
Sección "Insertar Dominio"	
Acción del Actor	**Respuesta del Sistema**

1. Accede a adicionar un nuevo dominio. 3. Introduce los datos del dominio	2. Muestra formulario para entrar el nuevo dominio. 4. Guarda la información en la base de datos. 5. Muestra mensaje de acción completada.
Precondiciones:	El dominio a modificar debe encontrarse en el sistema.
Flujo Normal de Eventos	
Sección "Modificar Dominio"	
Acción del Actor	**Respuesta del Sistema**
1. Accede a modificar un dominio. 3. Selecciona el dominio que desea modificar. 5. Realiza los cambios	2. El sistema muestra un listado con todos los dominios. 4. Muestra los datos de dominio. 5. Guarda los cambios realizados. 6. Muestra el listado de dominios actualizado.
Precondiciones:	El dominio a eliminar debe encontrarse en el sistema.
Flujo Normal de Eventos	
Sección "Eliminar Dominio"	
Acción del Actor	**Respuesta del Sistema**
1. Accede a eliminar un dominio. 3. Selecciona el dominio que desea eliminar.	2. El sistema muestra un listado con todos los dominios. 4. Elimina el dominio de la base de datos. 5. Muestra el listado de todos los dominios actualizado.
Referencias	RF 7 (7.1, 7.2, 7.3)

Descripción textual CU "Seleccionar Activos del Área".

Caso de Uso:	Seleccionar Activos del Área
Actores:	Director de Seguridad Informática, Especialista de Análisis de Riesgo, Responsable de Control de Inventario.

Resumen:	El caso de uso se inicia cuando el usuario debe seleccionar los activos pertenecientes a su área. El caso de uso termina cuando se introducen los datos de los activos seleccionados y el sistema guarda los cambios.
Precondiciones:	

Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. Accede a seleccionar activos del área.	2. Muestra el listado de activos globales.
3. Selecciona los activos correspondientes a su área.	4. Guarda los activos seleccionados. 5. Muestra formulario para introducir datos.
6. Introduce los datos de los activos.	7. Guarda los datos de los activos.

Referencias	RF 6

Descripción textual del CU "Gestionar Activos".

Caso de Uso:	Gestionar Activos
Actores:	Director de Seguridad Informática, Especialista de Análisis de Riesgo, Responsable de Control de Activos
Resumen:	El caso de uso se inicia cuando el usuario necesita Adicionar, Modificar o Eliminar activos informáticos al sistema. El caso de uso termina cuando el sistema guarda los cambios realizados.
Precondiciones:	**-**
Flujos Normal de Eventos	
Sección "Adicionar Activos"	
Acción del Actor	**Respuesta del Sistema**
1. Accede a adicionar un nuevo activo. 3. Introduce los datos del activo.	2. Muestra un formulario para escribir los datos del activo. 4. Guarda los datos del activo. 5. Muestra mensaje de acción completada.
Precondiciones:	El activo a modificar debe encontrarse en el listado.
Flujos Normal de Eventos	
Sección "Modificar Activos"	
Acción del Actor	**Respuesta del Sistema**
1. Accede a modificar los datos del activo. 3. Selecciona el activo que desea modificar. 5. Realiza los cambios sobre el o los activos a los que desea modificar los datos.	2. Muestra el listado de los activos del área. 4. Muestra los datos del activo. 6. Guarda los cambios realizados sobre los datos del activo. 7. Muestra el listado de activos actualizado.
Precondiciones:	El activo a eliminar debe encontrarse en el listado.
Flujos Normal de Eventos	
Sección "Eliminar Activos"	
Acción del Actor	**Respuesta del Sistema**
1. Accede a eliminar un activo de los que el tiene seleccionado. 3. Selecciona el activo a eliminar.	2. Muestra el listado de activos de su área. 4. Elimina el activo de la base de datos. 5. Muestra el listado de activos actualizado.
Referencias	RF 5(5.1, 5.2, 5.3)

Descripción textual CU "Adicionar Amenazas".

Caso de Uso:	Adicionar Amenazas	
Actores:	Director de Seguridad Informática, Especialista de análisis de riesgo	
Resumen:	El caso de uso se inicia cuando el Director de Seguridad Informática o el Especialista de Análisis de Riesgo necesitan Adicionar una nueva amenaza en el sistema. El caso de uso termina cuando el sistema guarda los cambios realizados en una de las secciones.	
Flujos Normal de Eventos		
Sección "Adicionar Amenazas"		
Acción del Actor		**Respuesta del Sistema**
1. Accede a adicionar una nueva amenaza. 3. Introduce la nueva amenaza.		2. Muestra un formulario para introducir la nueva amenaza. 4. Guarda la amenaza introducida en la base de datos. 5. Muestra mensaje de acción completada.
Referencias	RF 9	

Descripción textual CU "Determinar amenazas sobre activos".

Caso de Uso:	Determinar amenazas sobre activos	
Actores:	Director de Seguridad Informática, Especialista de Análisis de Riesgo	
Resumen:	El caso de uso se inicia cuando el Director de Seguridad Informática o el Especialista de Análisis de Riesgo van a determinar las amenazas que influyen sobre los activos de su área. El caso de uso termina cuando el sistema guarda	
Precondiciones:	-	
Flujo Normal de Eventos		
Acción del Actor		**Respuesta del Sistema**
1. Accede a seleccionar amenazas. 3. Selecciona un activo. 5. Selecciona las amenazas que influyen sobre el		2. Muestra listado de activos del área. 4. Muestra el listado de amenazas existentes. 6. Guarda el activo y las amenazas.

activo.	
Referencias	RF 10

Descripción textual CU "Realizar Estimación de Riesgo".

Caso de Uso:	Realizar Estimación de Riesgo
Actores:	Director de Seguridad Informática, Especialista de Análisis de Riesgo
Resumen:	El caso de uso se inicia cuando el Director de Seguridad Informática o el Especialista de Análisis de Riesgo van a realizar el análisis de riesgo, el sistema muestra la tabla de Estimación de Riesgo y se deben llenar los campos de la tabla para que el sistema realice los cálculos correspondientes. El caso de uso termina cuando el sistema guarda los datos de la tabla.
Precondiciones:	
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. Accede a realizar la estimación de riesgo. 3. Introduce los valores de probabilidad de que se materialicen las amenazas identificadas sobre cada bien informático.	2. Muestra una tabla con los campos a llenar. 4. Calcula el riesgo estimado sobre cada bien informático. 5. Calcula el riesgo de cada activo. 6. Calcula el Peso Relativo del Riesgo sobre cada bien informático. 7. Calcular el Peso Total del Riesgo del Sistema. 8. Guarda los datos de la tabla en la base de datos.
Referencias	RF 11 (11.1, 11.2, 11.3)

Descripción textual CU "Gestionar Políticas".

Caso de Uso:	Gestionar Políticas
Actores:	Director de Seguridad Informática, Especialista de Análisis de Riesgo,

	Responsable de Políticas
Resumen:	El caso de uso se inicia cuando el Director de Seguridad Informática o el Especialista de Análisis de Riesgo o el Responsable de Políticas necesitan Adicionar, Modificar o Eliminar políticas al sistema. El caso de uso termina cuando el sistema guarda los cambios realizados en una de las secciones.
Precondiciones:	**-**
Flujos Normal de Eventos	
Sección "Adicionar Políticas"	
Acción del Actor	**Respuesta del Sistema**
1. Accede a adicionar una nueva política. 3. Introduce la nueva política.	2. Muestra un formulario para introducir la nueva política. 4. Guarda la nueva política introducida. 5. Muestra mensaje de acción completada.
Precondiciones:	La política a modificar debe encontrarse en el listado.
Flujos Normal de Eventos	
Sección "Modificar Políticas"	
Acción del Actor	**Respuesta del Sistema**
1. Accede a modificar una política. 3. Selecciona la política a modificar. 4. Realiza los cambios.	2. Muestra el listado de políticas existentes en su área. 4. Muestra los datos de la política. 5. Guarda los cambios en la base de datos. 6. Muestra la política actualizada.
Precondiciones:	La política a eliminar debe encontrarse en el listado.
Flujos Normal de Eventos	
Sección "Eliminar Políticas"	
Acción del Actor	**Respuesta del Sistema**
1. Accede a eliminar políticas de seguridad. 3. Selecciona la política a eliminar.	2. Muestra el listado de políticas existentes en su área. 4. Elimina la política de la base de datos.

	5. Muestra el listado de políticas actualizado.
Referencias	RF 12 (12.1, 12.2, 12.3)

Descripción textual CU "Gestionar Medidas".

Caso de Uso:	Gestionar Medidas
Actores:	Director de Seguridad Informática, Especialista de Análisis de Riesgo, Ingeniero de Sistemas
Resumen:	El caso de uso se inicia cuando el Director de Seguridad Informática o el Especialista de Análisis de Riesgo o el Ingeniero de Sistemas necesitan Adicionar, Modificar o Eliminar medidas de seguridad al sistema. El caso de uso termina cuando el sistema guarda los cambios realizados en una de las secciones.
Precondiciones:	**-**
Flujos Normal de Eventos	
Sección "Adicionar Medidas"	
Acción del Actor	**Respuesta del Sistema**
1. Accede a adicionar una nueva medida de seguridad. 3. Introduce la nueva medida de seguridad.	2. Muestra un formulario para introducir la nueva medida de seguridad. 4. Guarda la nueva medida de seguridad introducida. 5. Muestra mensaje de acción completada.
Precondiciones:	La medida a modificar debe encontrarse en el listado.
Flujos Normal de Eventos	
Sección "Modificar Medidas"	
Acción del Actor	**Respuesta del Sistema**
1. Accede a modificar medidas de seguridad. 3. Selecciona la medida de seguridad. 4. Realiza los cambios.	2. Muestra el listado de medidas de seguridad de su área. 4. Muestra los datos de la medida. 5. Guarda los cambios en la base de datos.

	6. Muestra la medida de seguridad actualizada.
Precondiciones:	La medida a eliminar debe encontrarse en el listado.
Flujos Normal de Eventos	
Sección "Eliminar Medidas"	
Acción del Actor	**Respuesta del Sistema**
1. Accede a eliminar una medida de seguridad. 3. Selecciona la medida de seguridad a eliminar.	2. Muestra el listado de medidas de seguridad de su área. 4. Elimina la medida de seguridad de la base de datos. 5. Muestra el listado de medidas de seguridad actualizado.
Referencias	RF 13 (13.1, 13.2, 13.3)

Descripción textual CU "Gestionar Procedimientos".

Caso de Uso:	Gestionar Procedimientos
Actores:	Director de Seguridad Informática, Especialista de Análisis de Riesgo, Ingeniero de Sistemas
Resumen:	El caso de uso se inicia cuando el Director de Seguridad Informática o el Especialista de Análisis de Riesgo o el Ingeniero de Sistemas necesitan Adicionar, Modificar o Eliminar procedimientos de seguridad al sistema. El caso de uso termina cuando el sistema guarda los cambios realizados en una de las secciones.
Precondiciones:	**-**
Flujos Normal de Eventos	
Sección "Adicionar Procedimientos"	
Acción del Actor	**Respuesta del Sistema**
1. Accede a adicionar un nuevo procedimiento de seguridad. 3. Introduce el nuevo procedimiento de seguridad.	2. Muestra un formulario para introducir el nuevo procedimiento de seguridad. 4. Guarda el nuevo procedimiento de seguridad

	introducido. 5. Muestra mensaje de acción completada.
Precondiciones:	El procedimiento a modificar debe encontrarse en el listado.
Flujos Normal de Eventos	
Sección "Modificar Procedimientos"	
Acción del Actor	**Respuesta del Sistema**
1. Accede a modificar un procedimiento de seguridad. 3. Selecciona el procedimiento de seguridad a modificar 5. Realiza los cambios.	2. Muestra el listado de procedimientos de seguridad de su área. 4. Muestra los datos del procedimiento. 6. Guarda los cambios en la base de datos. 7. Muestra el procedimiento de seguridad actualizado.
Precondiciones:	El procedimiento a eliminar debe encontrarse en el listado.
Flujos Normal de Eventos	
Sección "Eliminar Procedimientos"	
Acción del Actor	**Respuesta del Sistema**
1. Accede a eliminar un procedimiento de seguridad. 3. Selecciona el procedimiento de seguridad a eliminar.	2. Muestra el listado de procedimientos de seguridad de su área. 4. Elimina el procedimiento de seguridad de la base de datos. 5. Muestra el listado de procedimientos de seguridad actualizado.
Referencias	RF 14 (14.1, 14.2, 14.3)

Descripción textual caso de uso "Gestionar Caracterización."

Caso de Uso:	Gestionar Caracterización
Actores:	Director de Seguridad Informática, Especialista de Análisis de Riesgo,

	Responsable de Control de Inventario
Resumen:	El caso de uso se inicia cuando el Director de Seguridad Informática o el Especialista de Análisis de Riesgo o el Ingeniero de Sistemas necesita Adicionar o Modificar la caracterización de su área. El caso de uso termina cuando el sistema guarda los cambios realizados en una de las secciones.
Precondiciones:	-
Flujo Normal de Eventos	
Sección "Adicionar Caracterización"	
Acción del Actor	**Respuesta del Sistema**
1. Accede a insertar la caracterización del área. 3. Introduce la caracterización.	2. Muestra formulario. 4. Guarda la información en la base de datos. 5. Muestra mensaje de acción completada.
Precondiciones:	-
Flujo Normal de Eventos	
Sección "Modificar Caracterización"	
Acción del Actor	**Respuesta del Sistema**
1. Accede a modificar la caracterización. 3. Realiza los cambios.	2. Muestra la caracterización. 4. Guarda los cambios en la base de datos. 5. Muestra mensaje de acción completada.
Referencias	RF 4 (4.1, 4.2)

Descripción textual caso de uso "Gestionar usuarios Dom. UCI"

Caso de Uso:	Gestionar Usuarios Dom. UCI
Actores:	Director de Seguridad Informática
Resumen:	El caso de uso se inicia cuando el Director de Seguridad Informática necesita insertar, modificar o eliminar un usuario del dominio UCI al sistema. Finaliza el caso de uso ejecutadas una de las acciones anteriores.
Precondiciones:	-
Flujo Normal de Eventos	

Sección "Insertar Usuario Dominio UCI"	
Acción del Actor	**Respuesta del Sistema**
1. Accede a insertar un usuario del dominio UCI. 3. Introduce los datos.	2. Muestra formulario para entrar datos del usuario. 4. Guarda la información en la base de datos. 5. Muestra mensaje de acción completada.
Precondiciones:	El usuario a modificar debe encontrarse en el sistema
Flujo Normal de Eventos	
Sección "Modificar Usuario Dominio UCI"	
Acción del Actor	**Respuesta del Sistema**
1. Accede a modificar un usuario dominio UCI. 3. Selecciona el usuario que desea modificar. 4. Realiza los cambios.	2. El sistema muestra un listado con todos los usuarios del dominio existentes en el área. 4. Muestra los datos del usuario UCI. 5. Guarda los cambios realizados. 6. Muestra el usuario actualizado.
Precondiciones:	El usuario a eliminar debe encontrarse en el sistema
Flujo Normal de Eventos	
Sección "Eliminar Usuario Dominio UCI"	
Acción del Actor	**Respuesta del Sistema**
1. Accede a eliminar un usuario dominio UCI. 3. Selecciona el usuario que desea eliminar.	2. El sistema muestra un listado con todos los usuarios del dominio existentes el área. 4. Elimina el usuario de la base de datos. 5. Muestra el listado de usuarios actualizado.
Referencias	RF 3 (3.1, 3.2, 3.3)

Descripción textual caso de uso "Gestionar áreas".

Caso de Uso:	Gestionar áreas
Actores:	Responsable de áreas

Resumen:	El caso de uso se inicia cuando el Responsable de áreas necesita insertar, modificar o eliminar un área al sistema. Finaliza el caso de uso ejecutadas una de las acciones anteriores.
Precondiciones:	-
Flujo Normal de Eventos	
Sección "Insertar área"	
Acción del Actor	**Respuesta del Sistema**
1. Accede a insertar una nueva área. 3. Introduce los datos.	2. Muestra formulario para entrar datos del área. 4. Guarda la información en la base de datos. 5. Muestra mensaje de acción completada.
Precondiciones:	El área a modificar debe encontrarse en el sistema.
Flujo Normal de Eventos	
Sección "Modificar área"	
Acción del Actor	**Respuesta del Sistema**
1. Accede a modificar un área. 3. Selecciona el área que desea modificar. 4. Realiza los cambios.	2. El sistema muestra un listado con todas las áreas. 4. Muestra los datos del área a modificar. 5. Guarda los cambios realizados. 6. Muestra listado con el área actualizada.
Precondiciones:	El área a eliminar debe encontrarse en el sistema.
Flujo Normal de Eventos	
Sección "Eliminar área"	
Acción del Actor	**Respuesta del Sistema**
1. Accede a eliminar un área. 3. Selecciona el área que desea eliminar.	2. El sistema muestra un listado con todas las áreas. 4. Elimina el área de la base de datos. 5. Muestra el listado actualizado.
Referencias	RF 15 (15.1, 15.2, 15.3)

Descripción textual caso de uso "Generar PSI".

Caso de Uso:	Generar PSI	
Actores:	Director de Seguridad Informática	
Resumen:	El caso de uso se inicia cuando después que los responsables de las actividades a ser realizadas en plan han concluido su parte, el Director de Seguridad Informática debe generar el PSI. El caso de uso termina cuando genera el PSI.	
Flujo Normal de Eventos		
Sección "Generar PSI"		
Acción del Actor		**Respuesta del Sistema**
1. Accede a generar el PSI. 3. Si todas las partes que conforman el PSI están aprobadas.		2. Muestra la opción de generar el PSI. 4. Se genera el PSI.
Flujo Alterno		
Acción del actor		**Respuesta del Sistema**
		3. Si aun no se ha aprobado el PSI, muestra un mensaje de error.
Referencias	RF 18	

Descripción textual caso de uso "Revisar y Aprobar PSI".

Caso de Uso:	Revisar y Aprobar PSI	
Actores:	Director de Seguridad Informática	
Resumen:	El caso de uso se inicia cuando después que los responsables de las actividades a ser realizadas en el PSI han concluido su parte, el Director de Seguridad Informática debe revisar y aprobar cada una de ellas. El caso de uso termina cuando guarda la revisión o la aprobación del PSI.	
Flujo Normal de Eventos		
Acción del Actor		**Respuesta del Sistema**

1. Selecciona lo que desea revisar. 3. Revisa lo que seleccionó. 7. Aprueba si lo desea.	2. Muestra la opción seleccionada. 4. Guarda la revisión. 5. Muestra mensaje de acción completada. 6. Habilita el botón Aprobar. 8. Guarda la aprobación. 9. Muestra mensaje de acción completada.
Referencias	RF 16

Descripción textual caso de uso "Gestionar Directores".

Caso de Uso:	Gestionar Directores
Actores:	Administrador
Resumen:	El caso de uso se inicia cuando el administrador debe insertar los directores de las áreas existentes en el sistema. El caso de uso termina cuando el sistema guardan los directores en la base de datos.
Flujo Normal de Eventos	
Sección "Insertar Directores"	
Acción del Actor	**Respuesta del Sistema**
1. Accede a insertar un director. 3. Introduce los datos.	2. Muestra formulario para la entrada de los datos. 4. Guarda la información en la base de datos. 5. Muestra mensaje de acción completada.
Flujo Normal de Eventos	
Sección "Modificar Directores"	
Acción del Actor	**Respuesta del Sistema**
1. Accede a modificar un director. 3. Selecciona el director que desea modificar. 5. Realiza los cambios.	2. Muestra listado de directores existentes 4. Muestra los datos del director. 5. Guarda los cambios realizados. 6. Muestra listado de directores actualizado.

Flujo Normal de Eventos	
Sección "Eliminar Directores"	
Acción del Actor	**Respuesta del Sistema**
1. Accede a eliminar un director. 3. Selecciona el área que desea eliminar.	2. Muestra listado de directores. 4. Elimina el director de la base de datos. 5. Muestra listado de directores actualizado.
Referencias	RF 17 (17.1, 17.2, 17.3)

Descripción textual caso de uso "Cambiar Contraseña".

Caso de Uso:	Cambiar contraseña
Actores:	Administrador
Resumen:	El caso de uso se inicia cuando el administrador desea cambiar su contraseña. El caso de uso termina cuando el sistema guarda los cambios realizados en la base de datos.
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. Accede a cambiar la contraseña. 3. Introduce la nueva contraseña.	2. Muestra formulario para la entrada de los datos. 4. Guarda la nueva contraseña en la base de datos. 5. Muestra mensaje de acción completada.
Referencias	RF 18

4

ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

En este capítulo se exponen los elementos del análisis y diseño e implementación de la solución. Primeramente se presentan los diagramas de clases del análisis, los diagramas de interacción y los diagramas de clases del diseño con extensiones web. También se añade una descripción de la base de datos mediante el diagrama de clases persistentes y el modelo de datos. Posteriormente en la etapa de implementación se muestra la estructura del sistema en términos de componentes a través del diagrama de componentes.

4.1 Modelo de Análisis.

Durante el análisis, se analizan los requerimientos funcionales refinándolos y estructurándolos, esto tiene como objetivo conseguir una comprensión más precisa de los requisitos. A pesar de que en el análisis hay un refinamiento de los requisitos, no se tiene en cuenta el lenguaje de programación a usar en la construcción, ni la plataforma en la que se ejecuta la aplicación, pues solo se quiere comprender perfectamente los requisitos del software y no precisar cómo se implementa la solución.

4.1.1 Diagramas de clases del análisis.

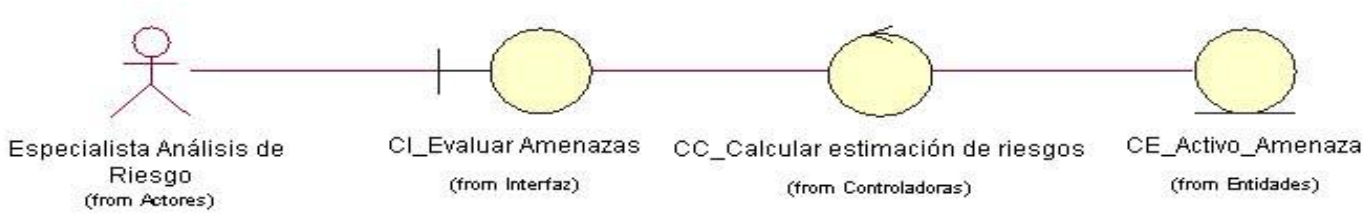


Figura 4.1 Diagrama de clases del análisis para el caso de uso "Realizar Estimación de Riesgo"

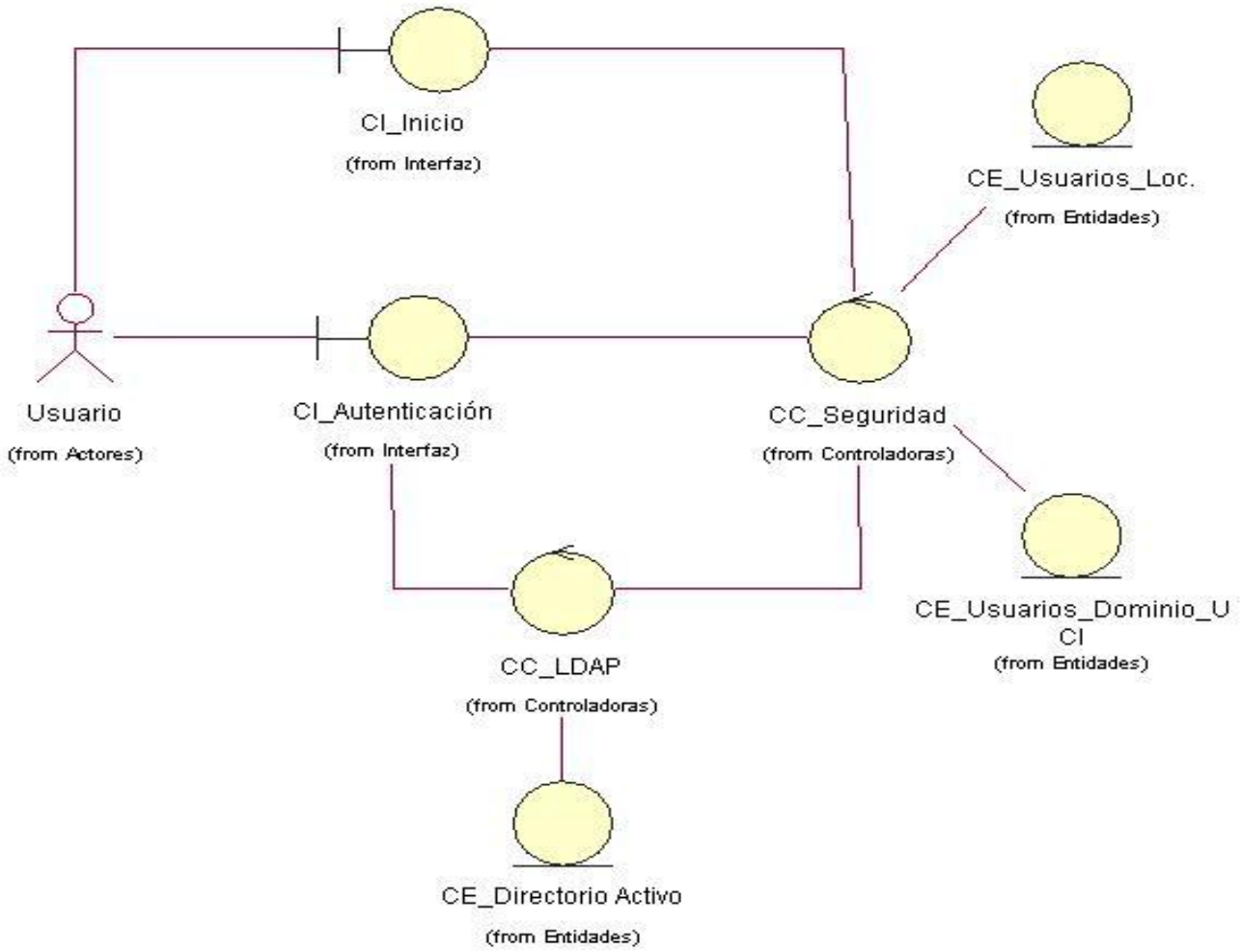


Figura 4.2 Diagrama de clases del análisis para el caso de uso "Autenticar Usuario".

Figura 4.3 Diagrama de clases del análisis para el caso de uso "Importancia de los Activos".

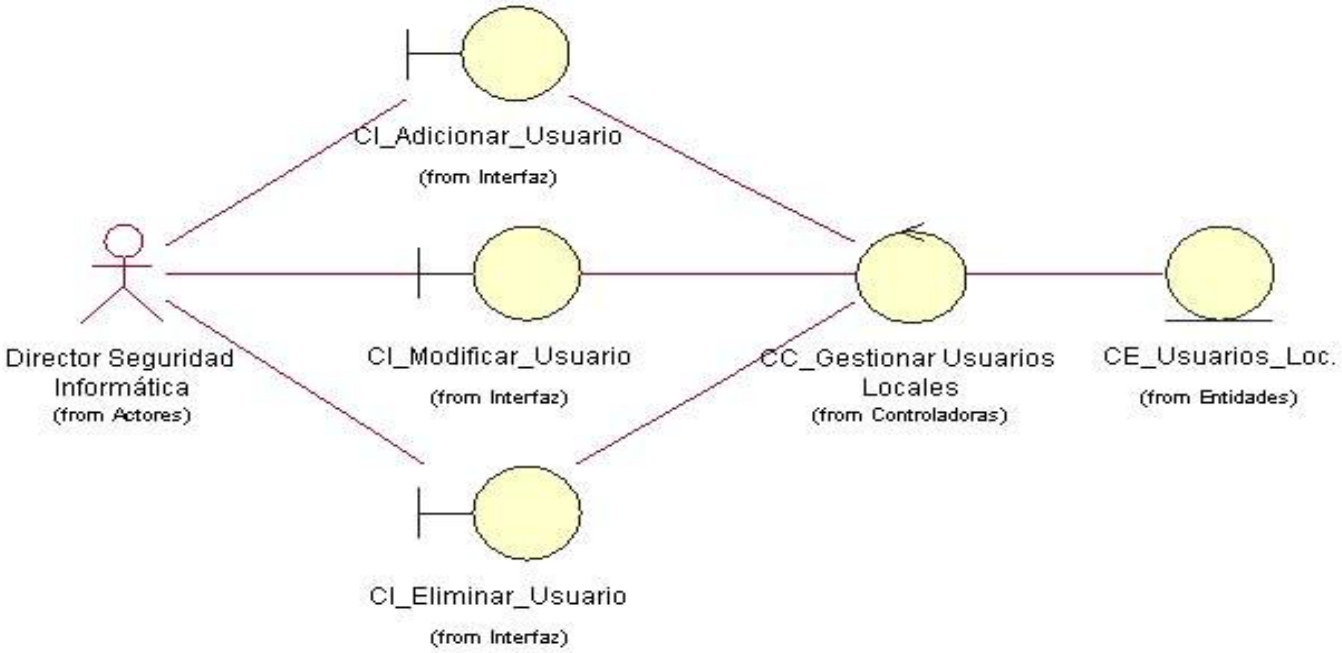


Figura 4.4 Diagrama de clases del análisis para el caso de uso "Gestionar Usuarios Locales".

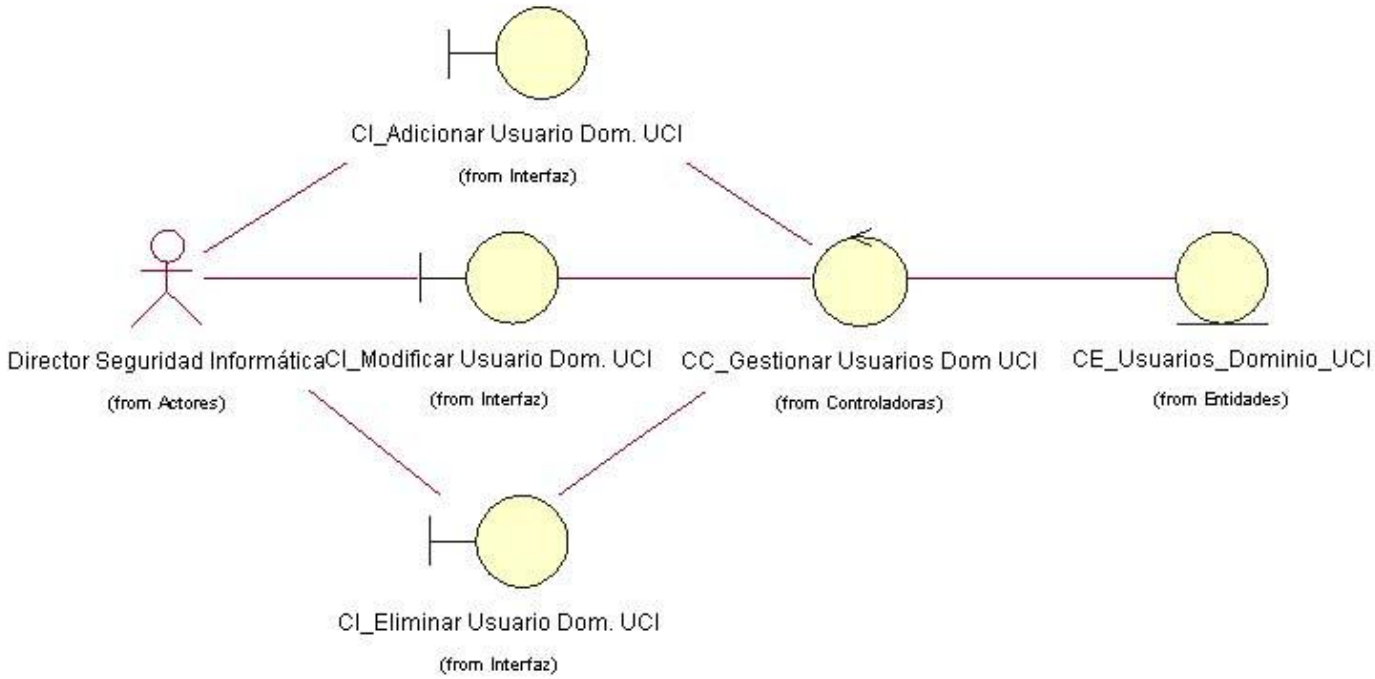


Figura 4.5 Diagrama de clases del análisis para el caso de uso "Gestionar Usuarios Dom. UCI".

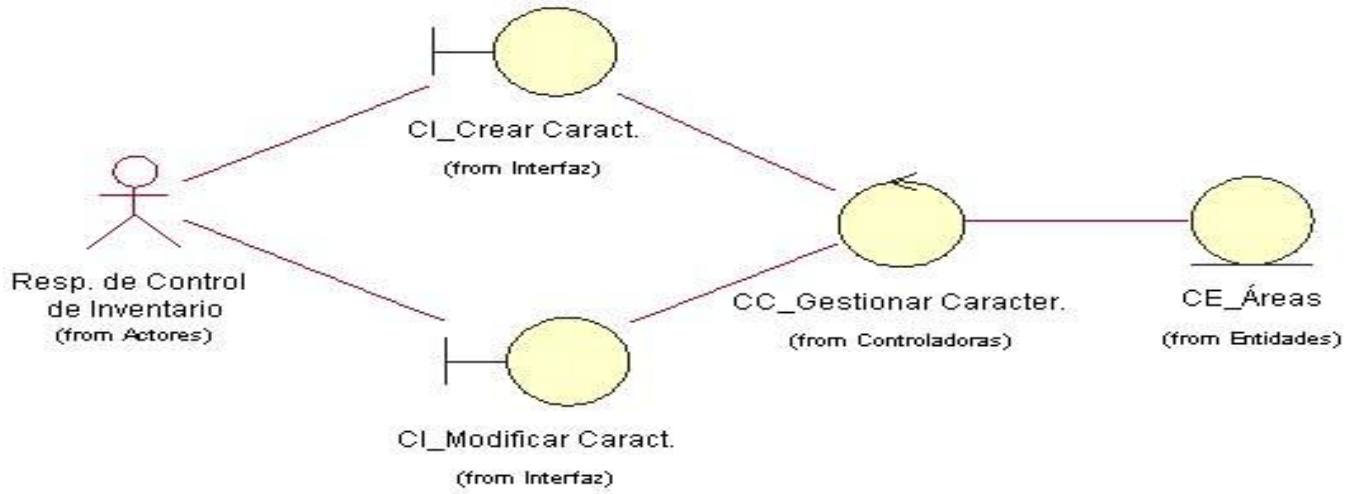


Figura 4.6 Diagrama de clases del análisis para el caso de uso "Gestionar Caracterización".

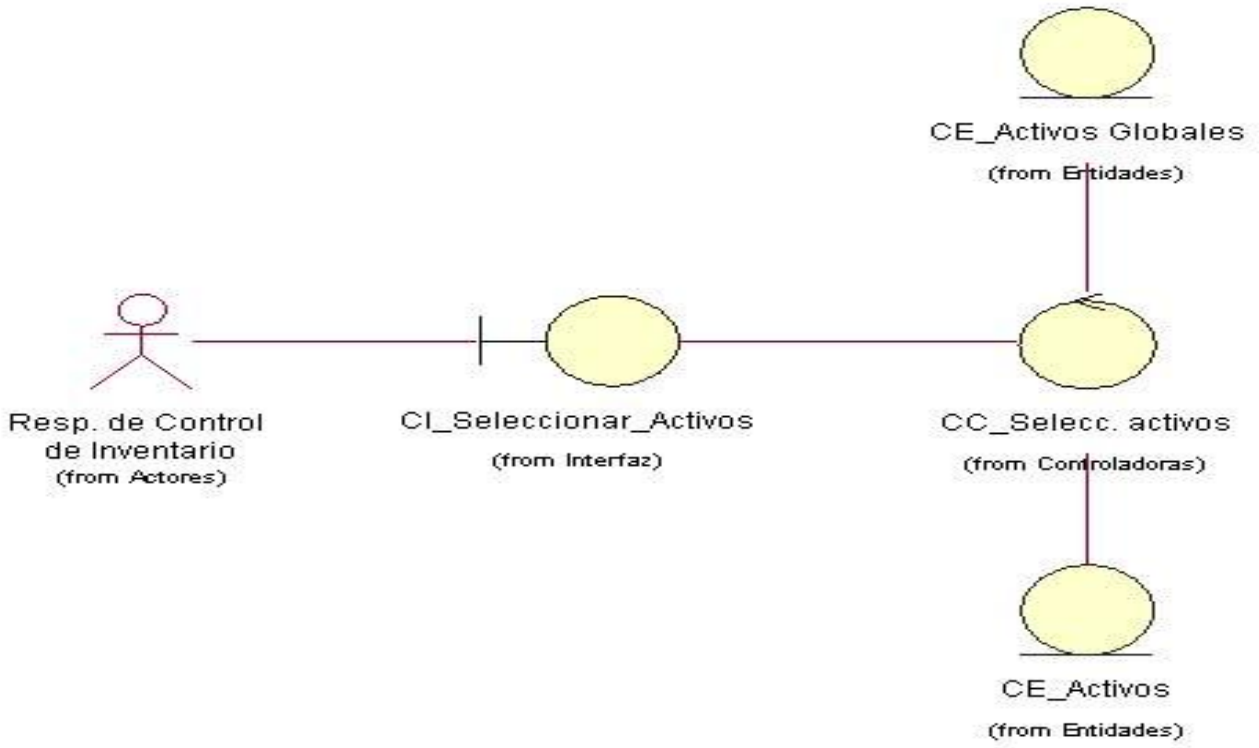


Figura 4.7 Diagrama de clases del análisis para el caso de uso "Seleccionar Activos del Área".

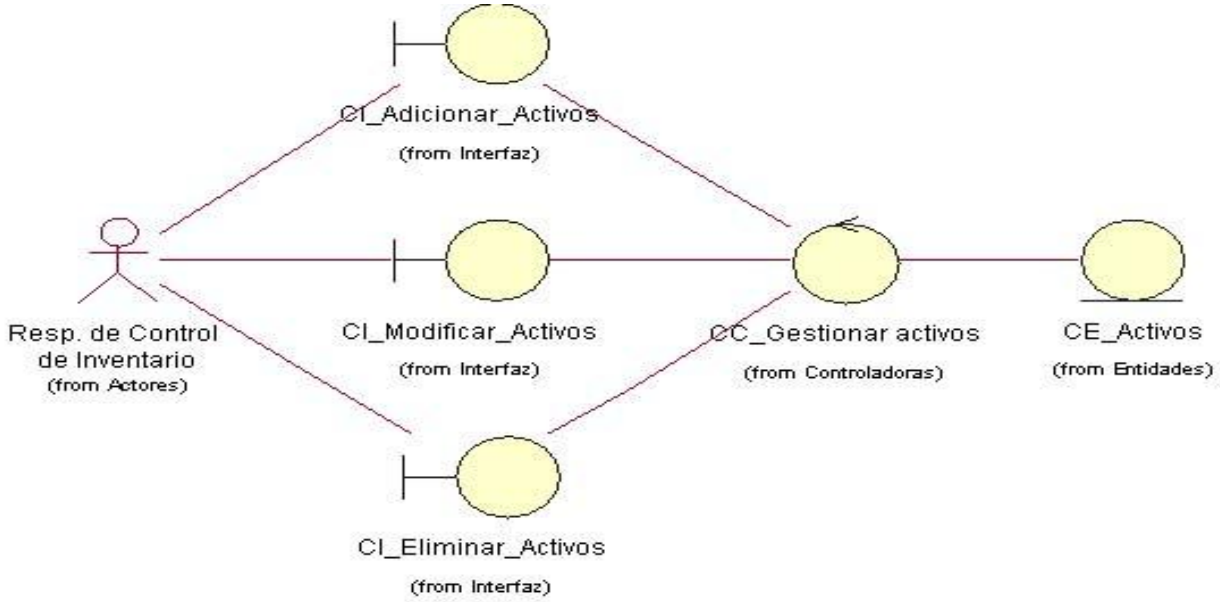


Figura 4.8 Diagrama de clases del análisis para el caso de uso "Gestionar Activos".

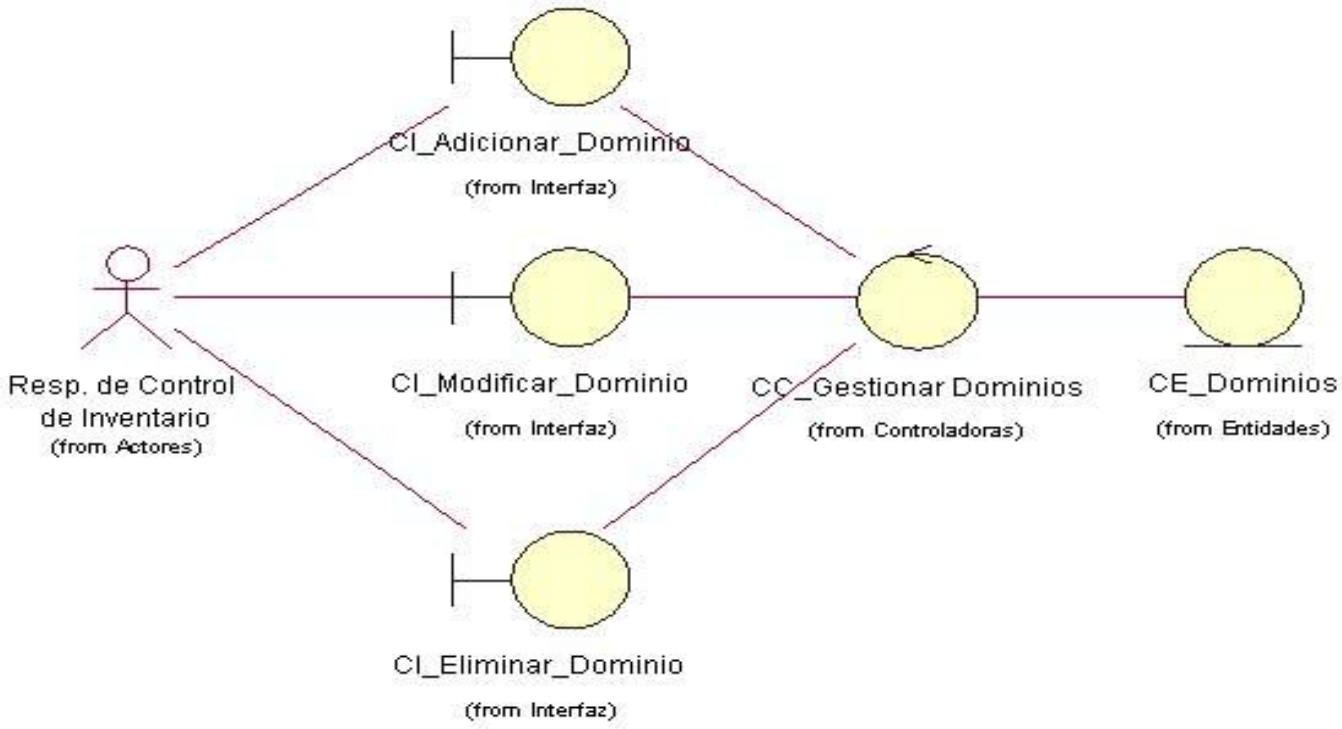


Figura 4.9 Diagrama de clases del análisis para el caso de uso "Gestionar Dominios".

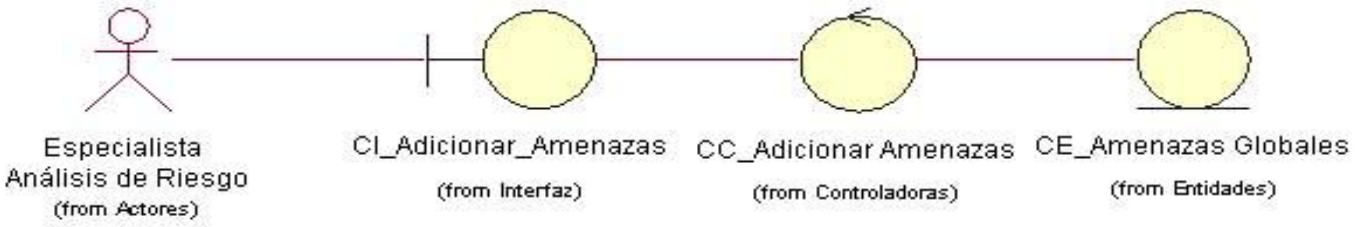


Figura 4.10 Diagrama de clases del análisis para el caso de uso "Adicionar Amenazas".

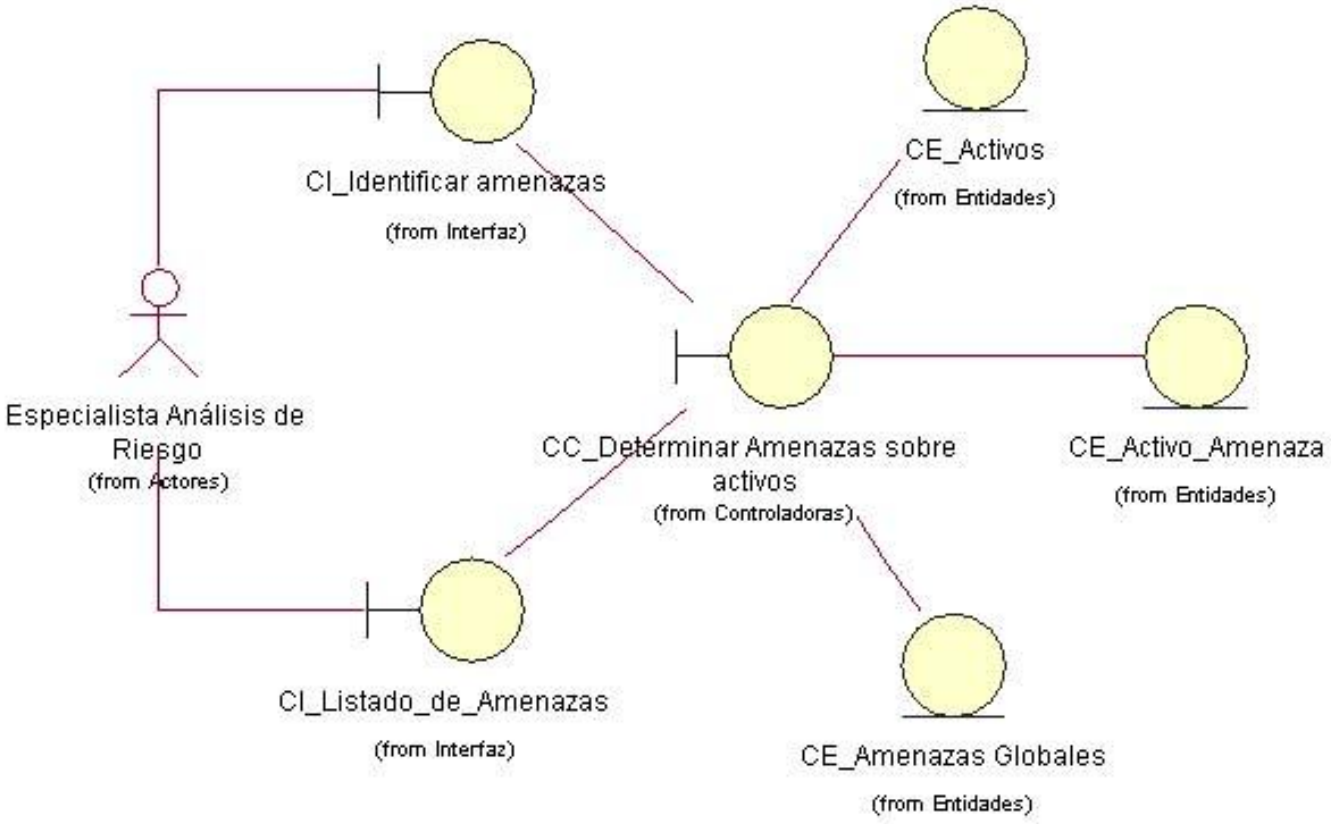


Figura 4.11 Diagrama de clases del análisis para el caso de uso "Determinar amenazas sobre activos".

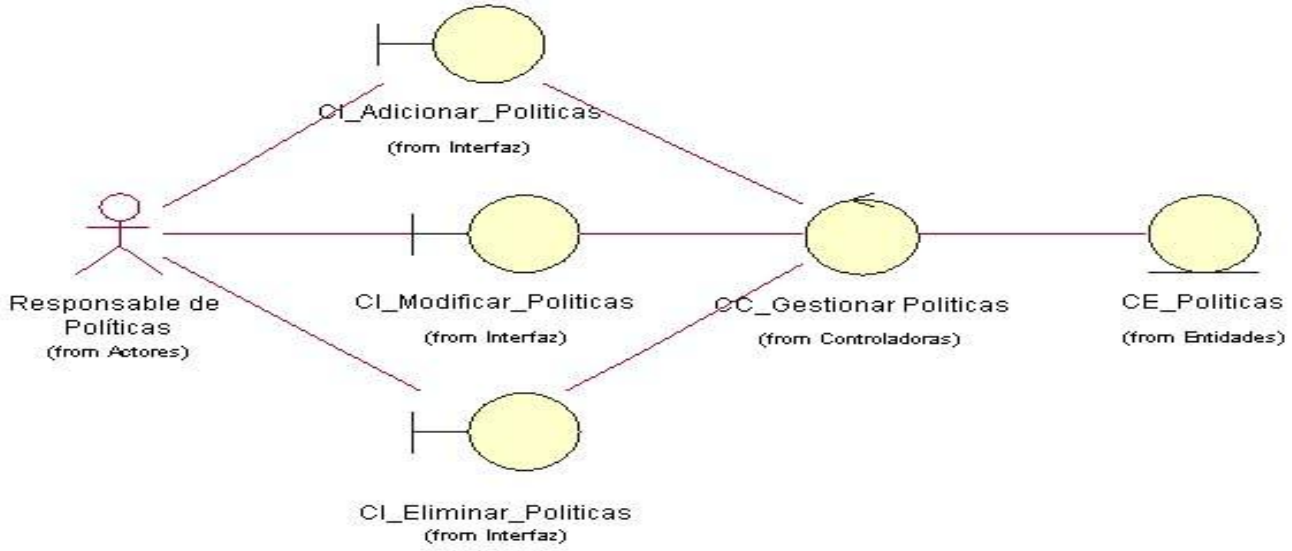


Figura 4.12 Diagrama de clases del análisis para el caso de uso "Gestionar Políticas".

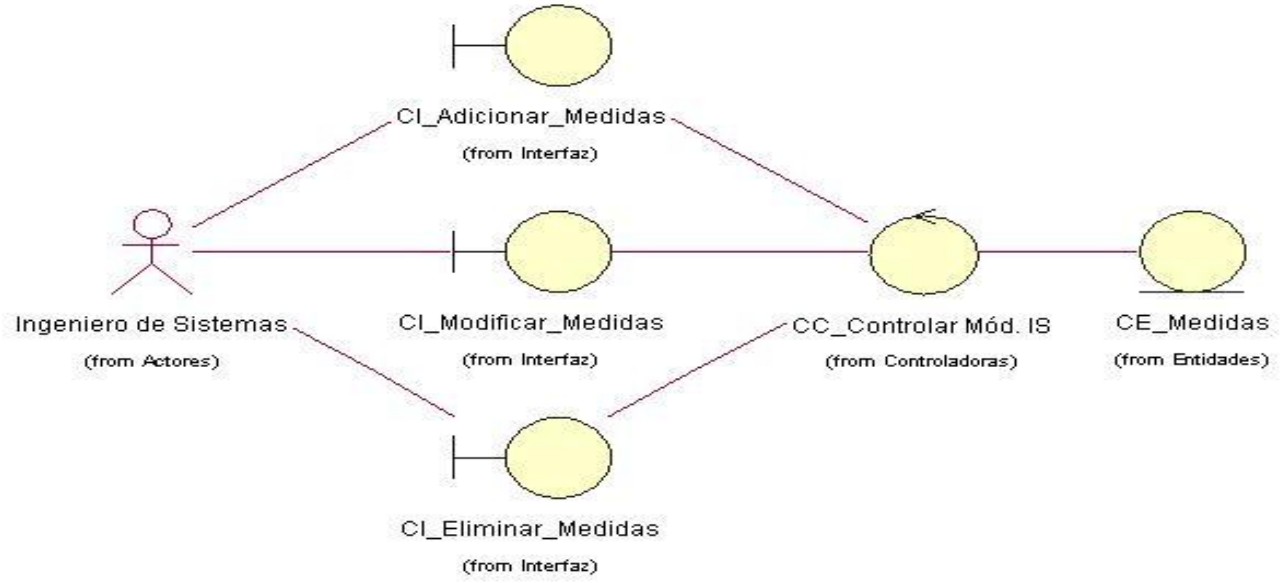


Figura 4.13 Diagrama de clases del análisis para el caso de uso "Gestionar Medidas".

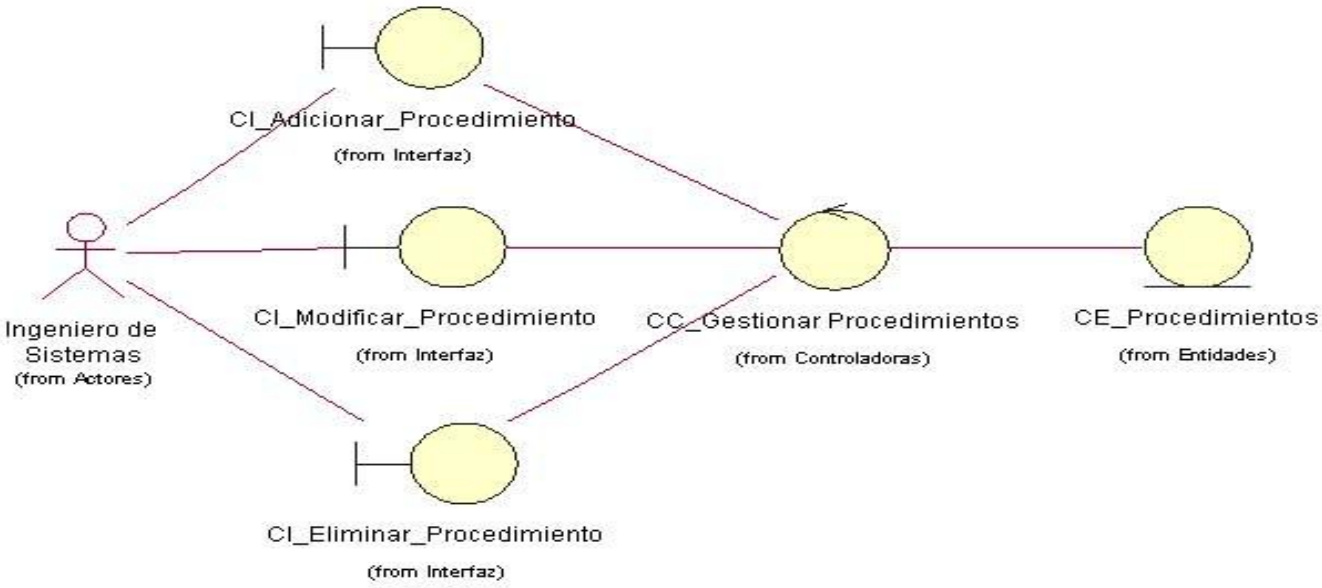


Figura 4.14 Diagrama de clases del análisis para el caso de uso "Gestionar Procedimientos".

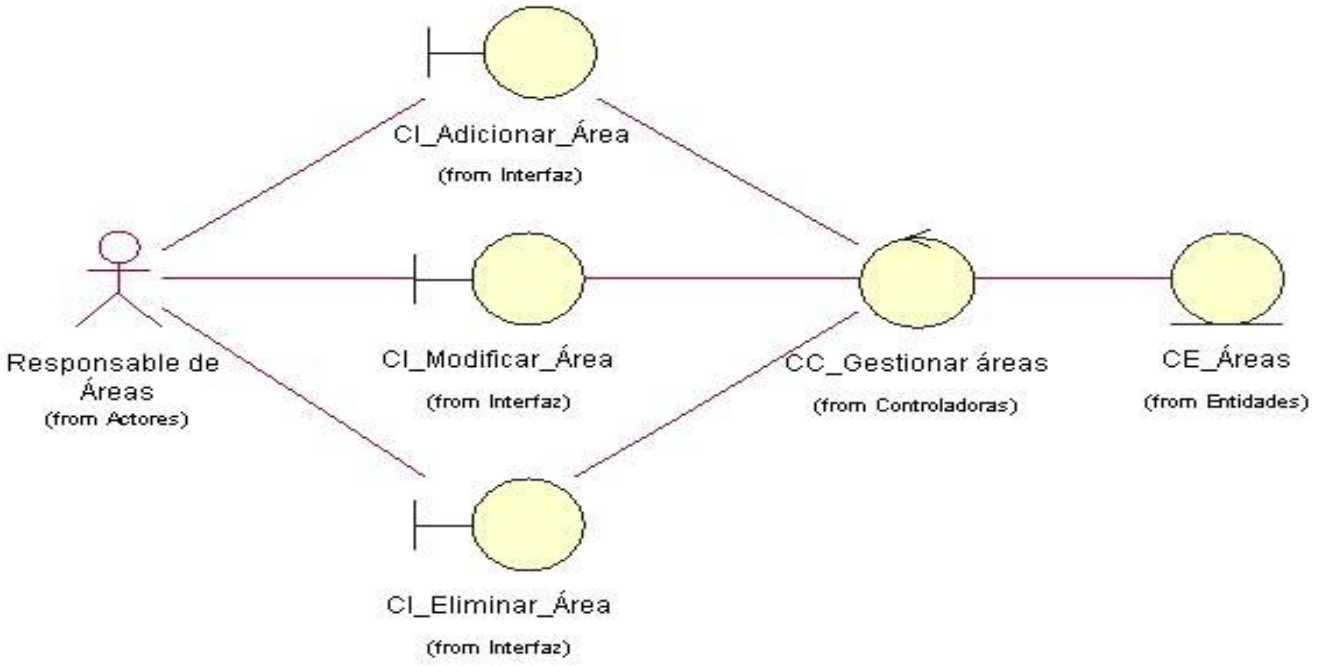


Figura 4.15 Diagrama de clases del análisis para el caso de uso "Gestionar Áreas".

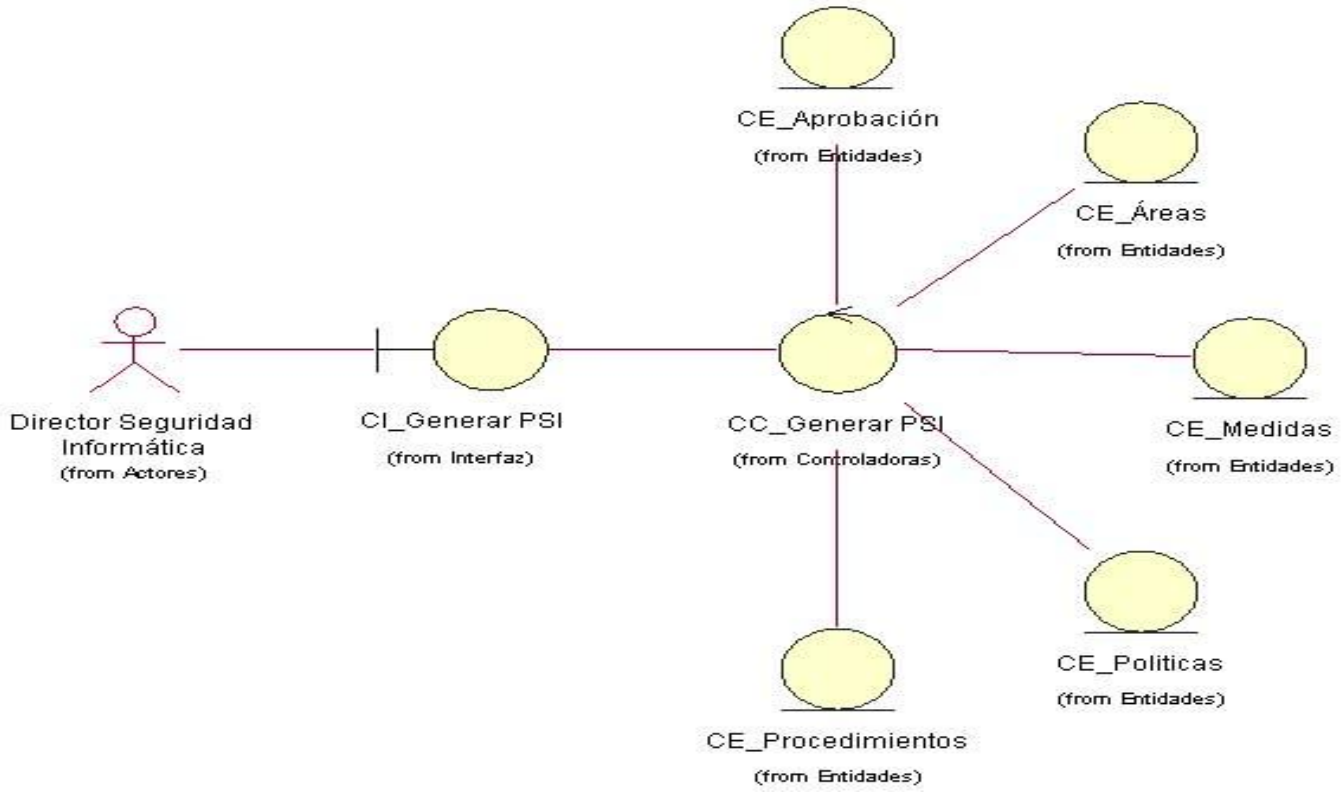


Figura 4.16 Diagrama de clases del análisis para el caso de uso "Generar PSI".

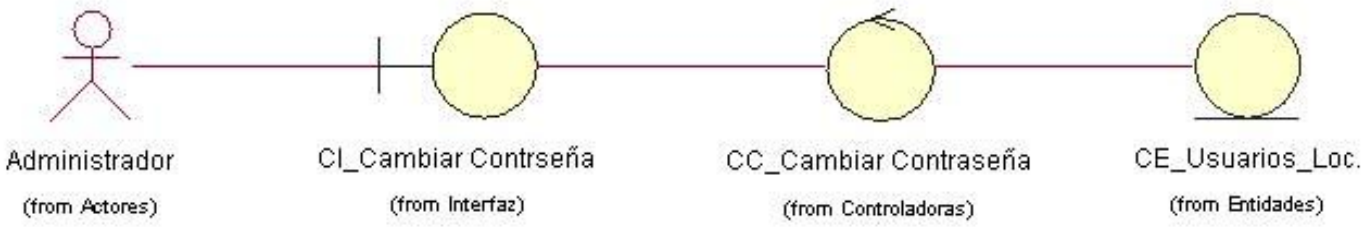


Figura 4.17 Diagrama de clases del análisis para el caso de uso "Cambiar Contraseña".

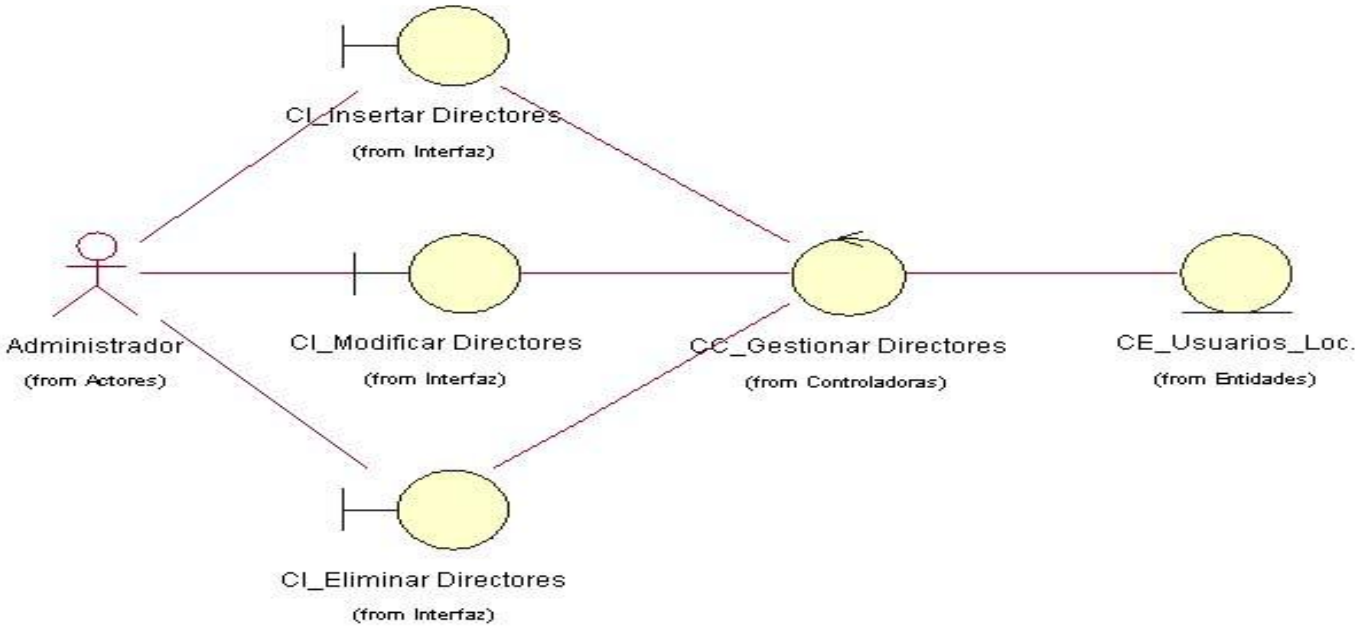


Figura 4.18 Diagrama de clases del análisis para el caso de uso "Gestionar Directores".

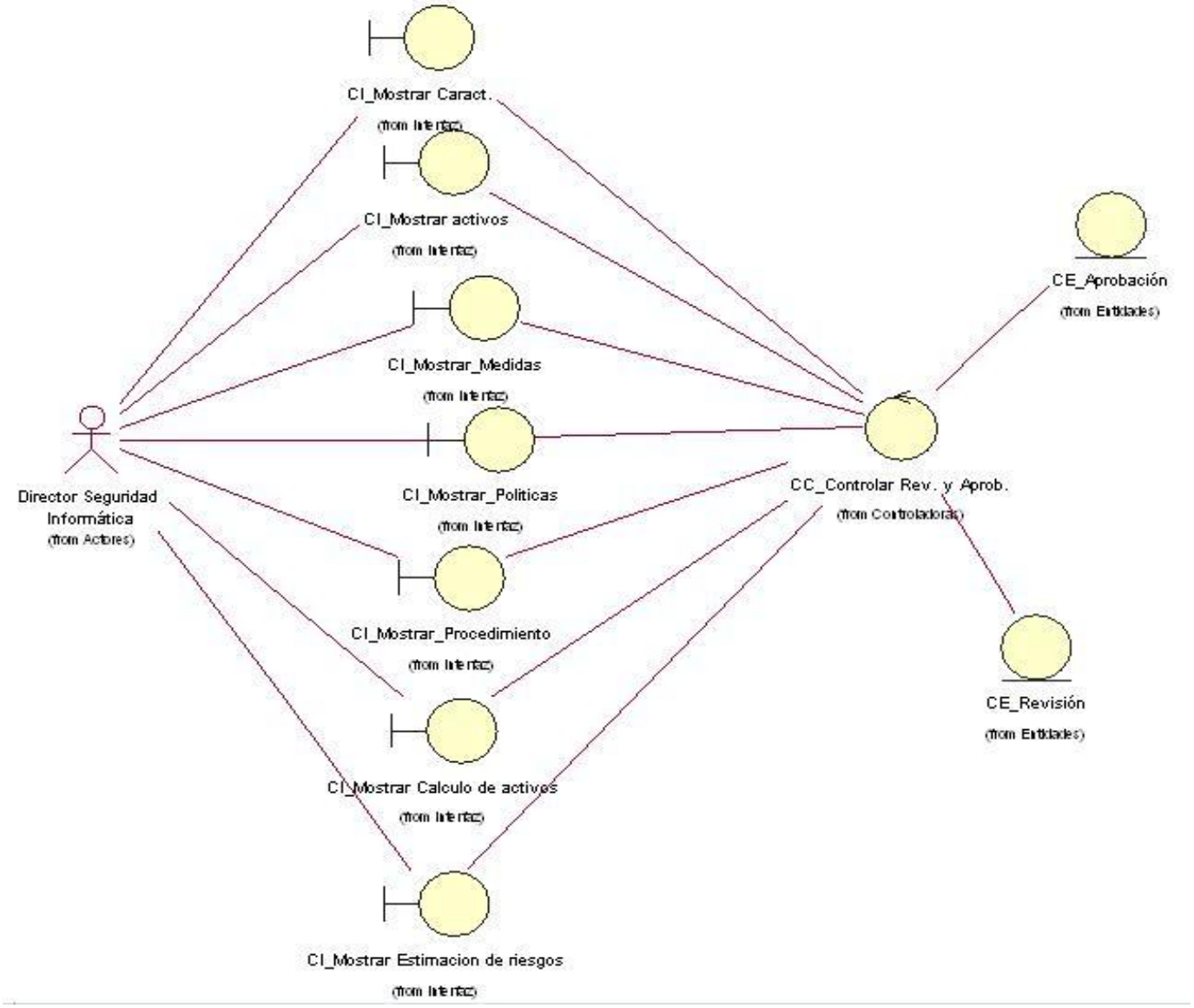


Figura 4.19 Diagrama de clases del análisis para el caso de uso "Revisar y Aprobar PSI".

4.1.2 Diagramas de interacción.

Los diagramas de interacción son los diagramas que permiten representar la interacción que tiene lugar entre los objetos mediante transferencia de mensajes entre los mismos. Son diagramas de interacción el diagrama de secuencia y el diagrama de colaboración.

Un diagrama de secuencia muestra una interacción que está organizada como una secuencia temporal. En particular, muestra los objetos que participan en la interacción mediante sus líneas de vida y mediante los mensajes que intercambian, organizados en forma de una secuencia temporal. Un diagrama de secuencia no muestra los enlaces existentes entre objetos. Los diagramas de secuencia tienen distintos formatos, adecuados para propósitos diferentes.

Un diagrama de colaboración muestra una interacción organizada en tomo a los objetos que efectúan operaciones. Es parecido a un diagrama de objetos que muestra los objetos y los enlaces existentes entre ellos que se necesitan para implementar una operación de nivel más elevado.

Para ver la representación de los diagramas de interacción del análisis, específicamente los diagramas de colaboración para cada caso de uso remitirse a los anexos del 4 al 46.

4.2 Modelo de diseño.

El diseño es el centro de atención final de la fase de elaboración y el comienzo de la fase de construcción. Esto contribuye a una arquitectura estable y sólida, y crear un plano del modelo de implementación. En el diseño se modela el sistema y se encuentra su forma, incluyendo la arquitectura para que soporte todos los requisitos, incluyendo los no funcionales. Una entrada esencial del diseño es el modelo de análisis.

4.2.1 Diagramas de clases con extensiones Web.

El diagrama de clases del diseño es el que permite mostrar las clases con sus atributos y métodos, e incluso; representar la colaboración que tiene lugar entre ellas. Se modelan además las páginas con los enlaces que las interrelacionan.

Con vistas a simplificar el diagrama de clases con extensiones web correspondiente al sistema que se propone, el mismo se ha dividido en 7 módulos de acuerdo a los 7 roles que interactúan. A continuación se describe cada uno de estos módulos.

- ✓ Módulo para el Responsable de áreas.
  Caso de uso: Gestionar Áreas.
- ✓ Módulo para el Responsable de Políticas.
  Caso de Uso: Gestionar Políticas.
- ✓ Módulo para el Responsable de Control de Inventario.
  Casos de Uso: Gestionar Dominios, Seleccionar Activos del Área, Gestionar Caracterización, Gestionar Activos.
- ✓ Módulo para el Ingeniero de Sistemas.
  Casos de Uso: Gestionar Medidas, Gestionar Procedimientos.
- ✓ Módulo para el Especialista de Análisis de Riesgo.
  Casos de Usos: Adicionar Amenazas, Determinar Amenazas sobre activos, Importancia de Activos, Realizar Estimación de Riegos y las funcionalidades correspondientes a los módulos Responsable de Control de Inventario, Responsable de Políticas e Ingeniero de Sistemas.
- ✓ Módulo para el Director de Seguridad Informática.
  Casos de Usos: Gestionar Usuarios Locales, Gestionar Usuarios Dominio UCI, Generar PSI y las funcionalidades correspondientes a los módulos Responsable de Control de Inventario, Responsable de Políticas, Ingeniero de Sistemas y Especialista de Análisis de Riesgo.
- ✓ Modulo para el Administrador
  Casos de Uso: Insertar Directores, Cambiar Contraseña.

Autenticar Usuarios

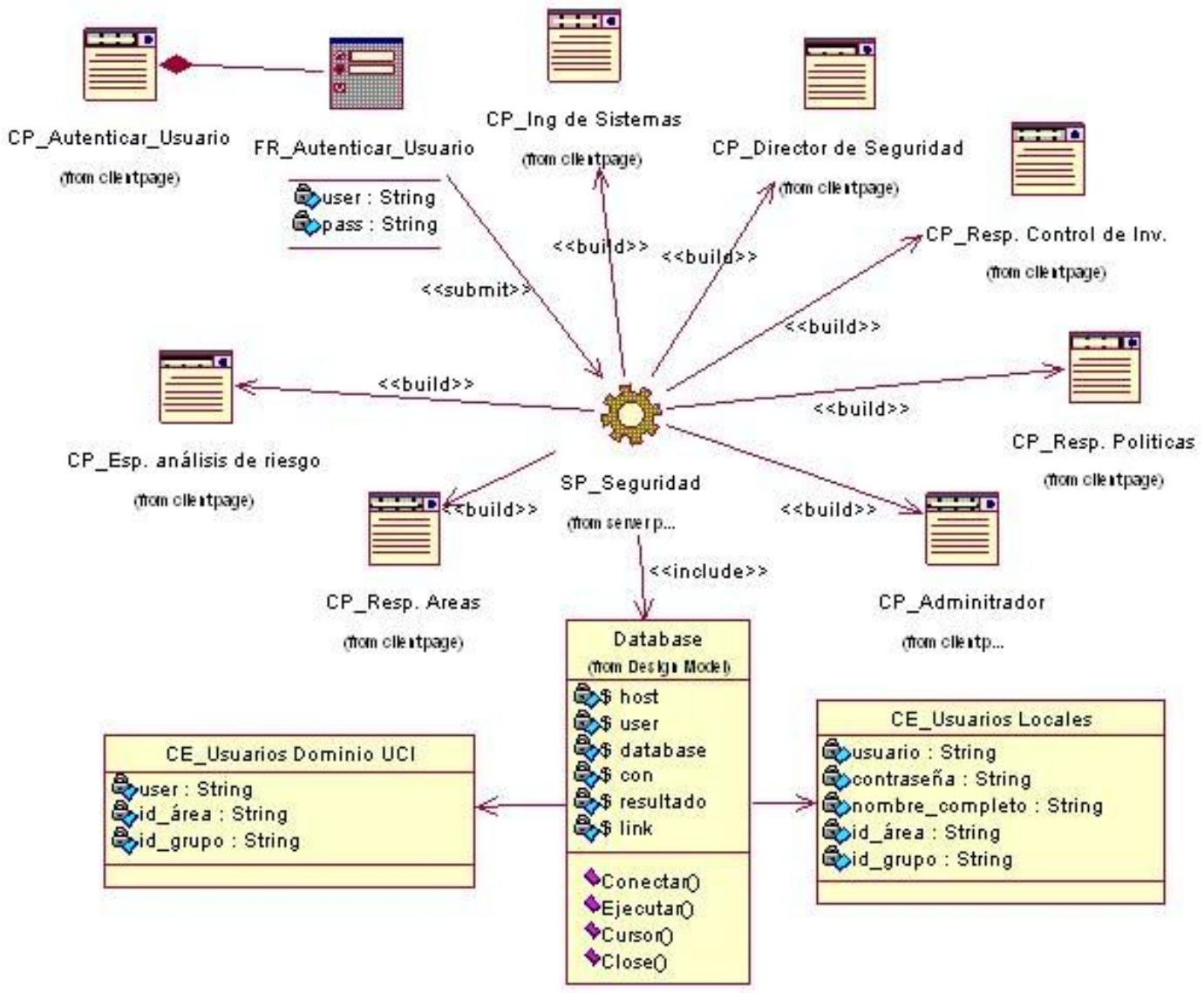


Figura 4.20 Diagrama de clases del diseño para el caso de uso "Autenticar Usuario".

Módulo Responsable de Áreas.

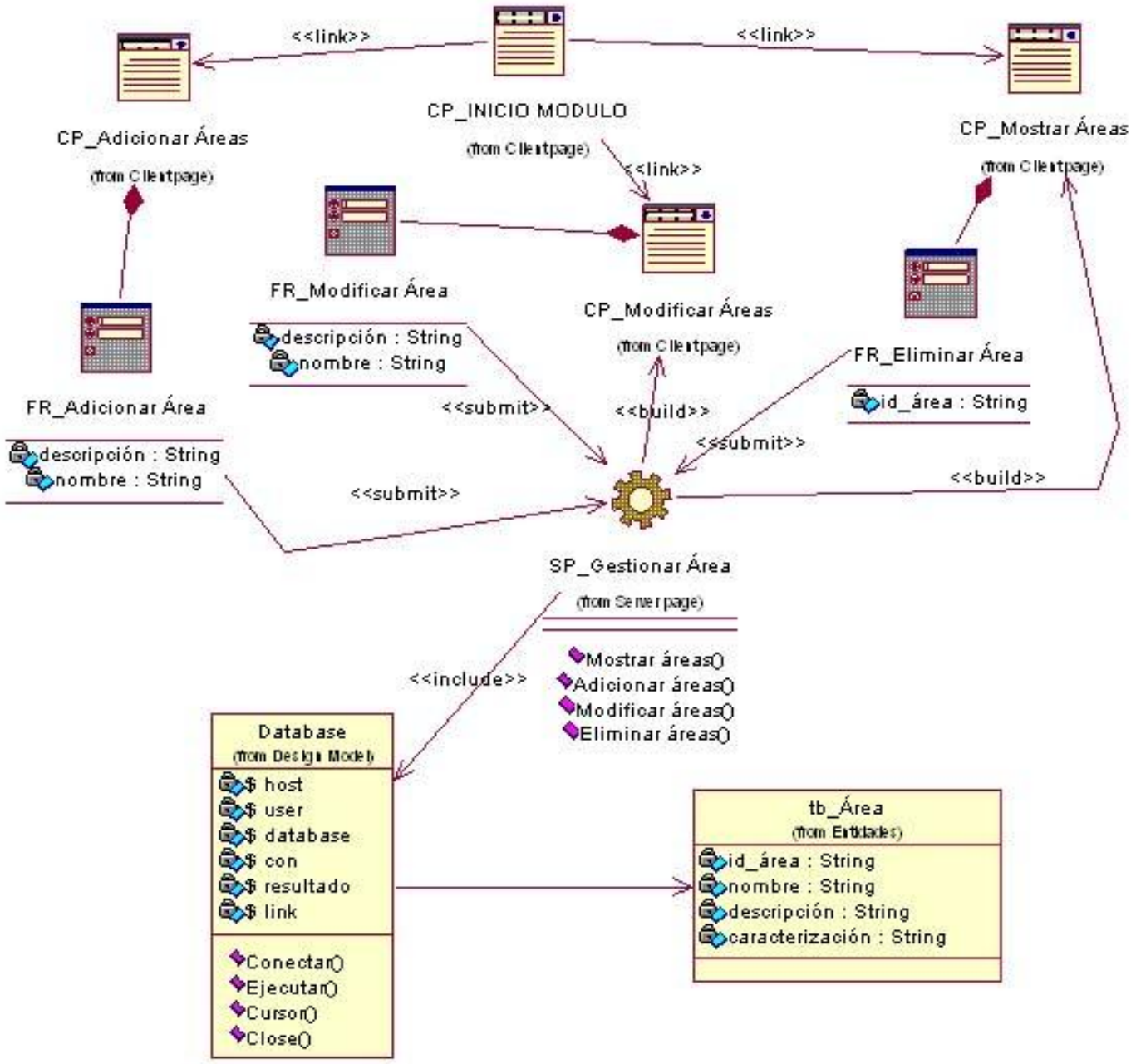


Figura 4.21 Diagrama de clases del diseño para el Módulo "Responsable de Áreas".

Módulo Responsable de Políticas

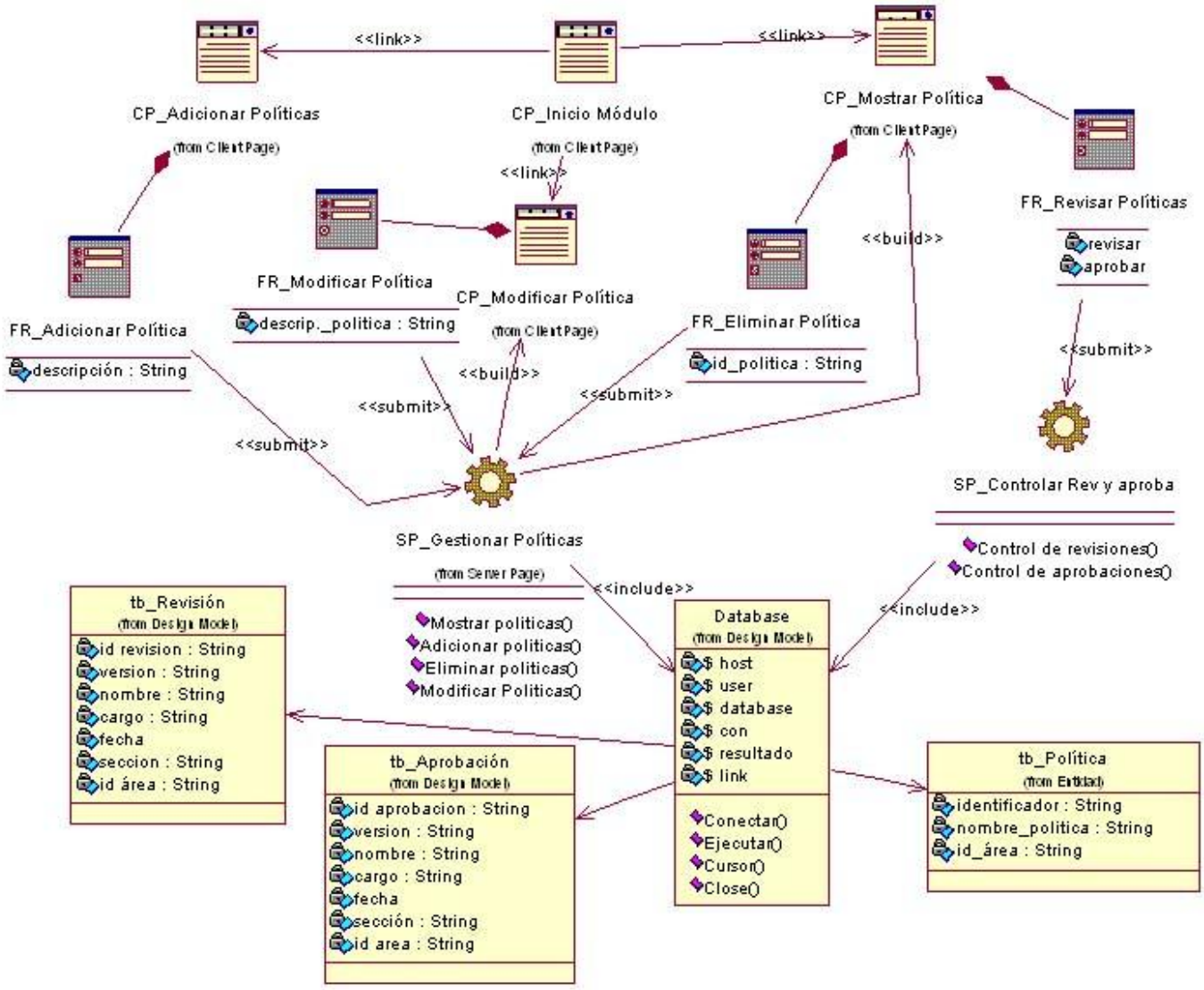


Figura 4.22 Diagrama de clases del diseño para el Módulo "Responsable de Políticas".

Módulo Responsable de Control de Inventario

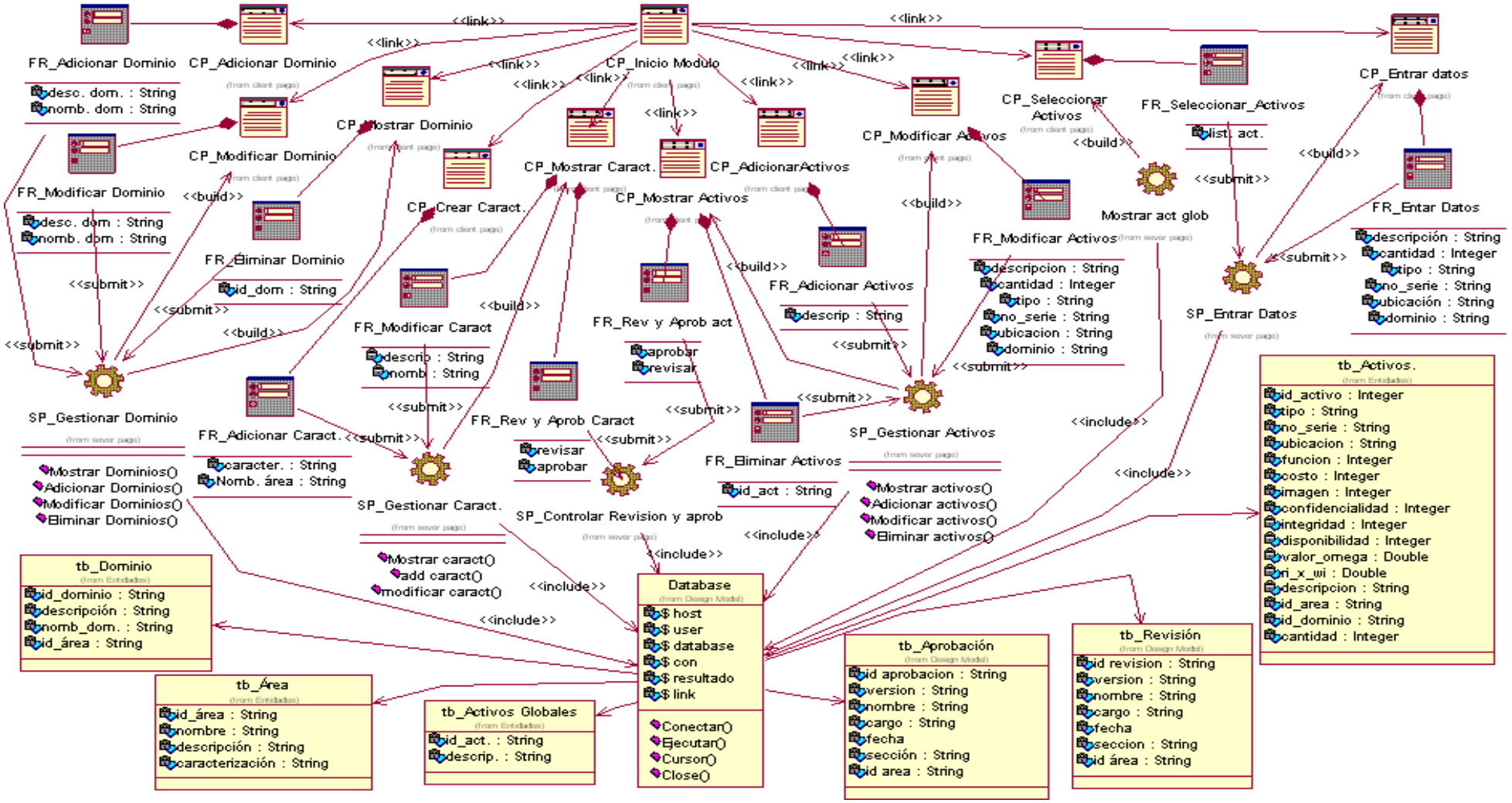


Figura 4.23 Diagrama de clases del diseño para el Módulo "Responsable de Control de Inventario".

Módulo Ingeniero de Sistemas

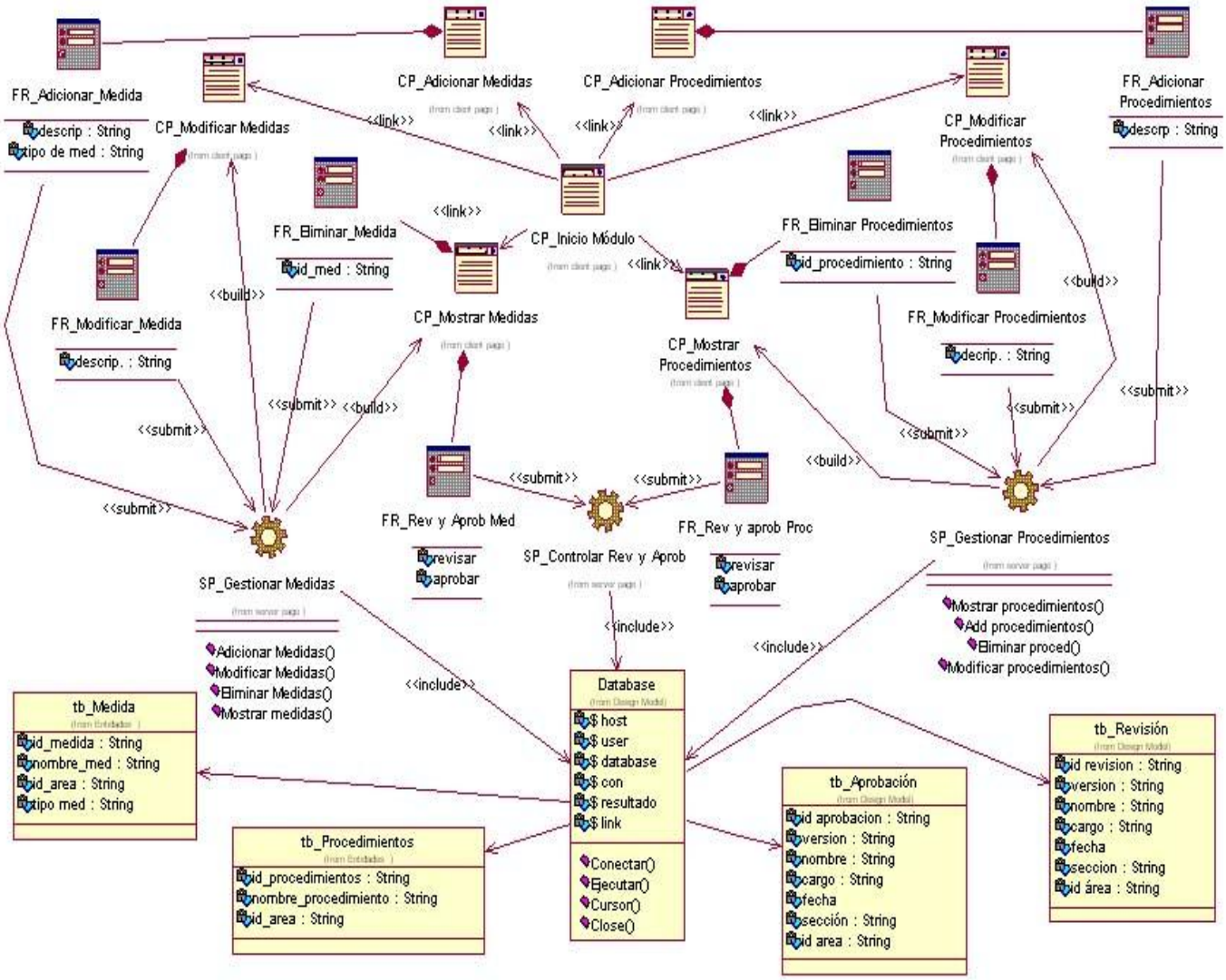


Figura 4.24 Diagrama de clases del diseño para el Módulo "Ingeniero de Sistemas".

Módulo Especialista de Análisis de Riesgo

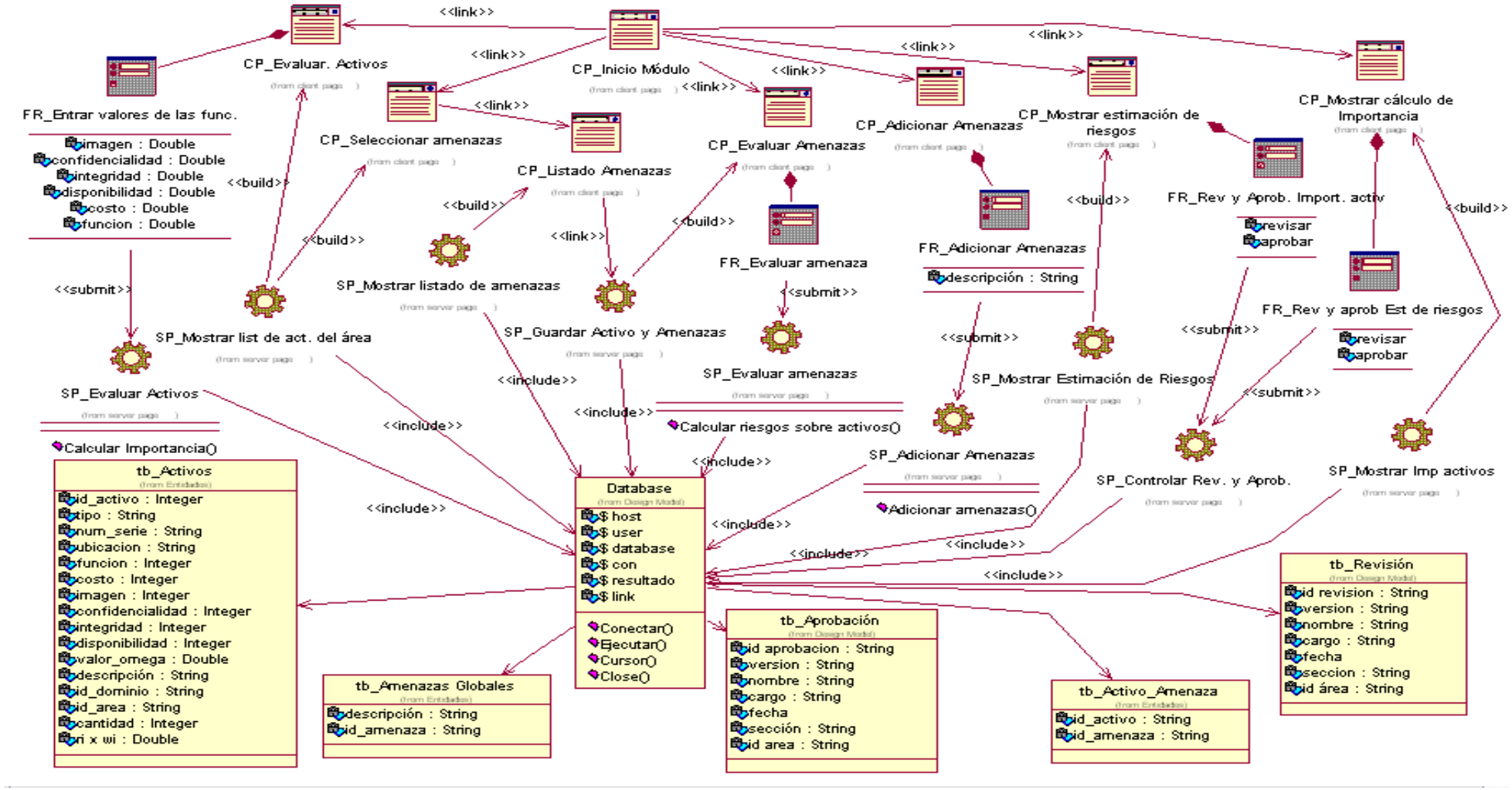


Figura 4.25 Diagrama de clases del diseño para el Módulo "Especialista de Análisis de Riesgo".

Módulo Director de Seguridad Informática

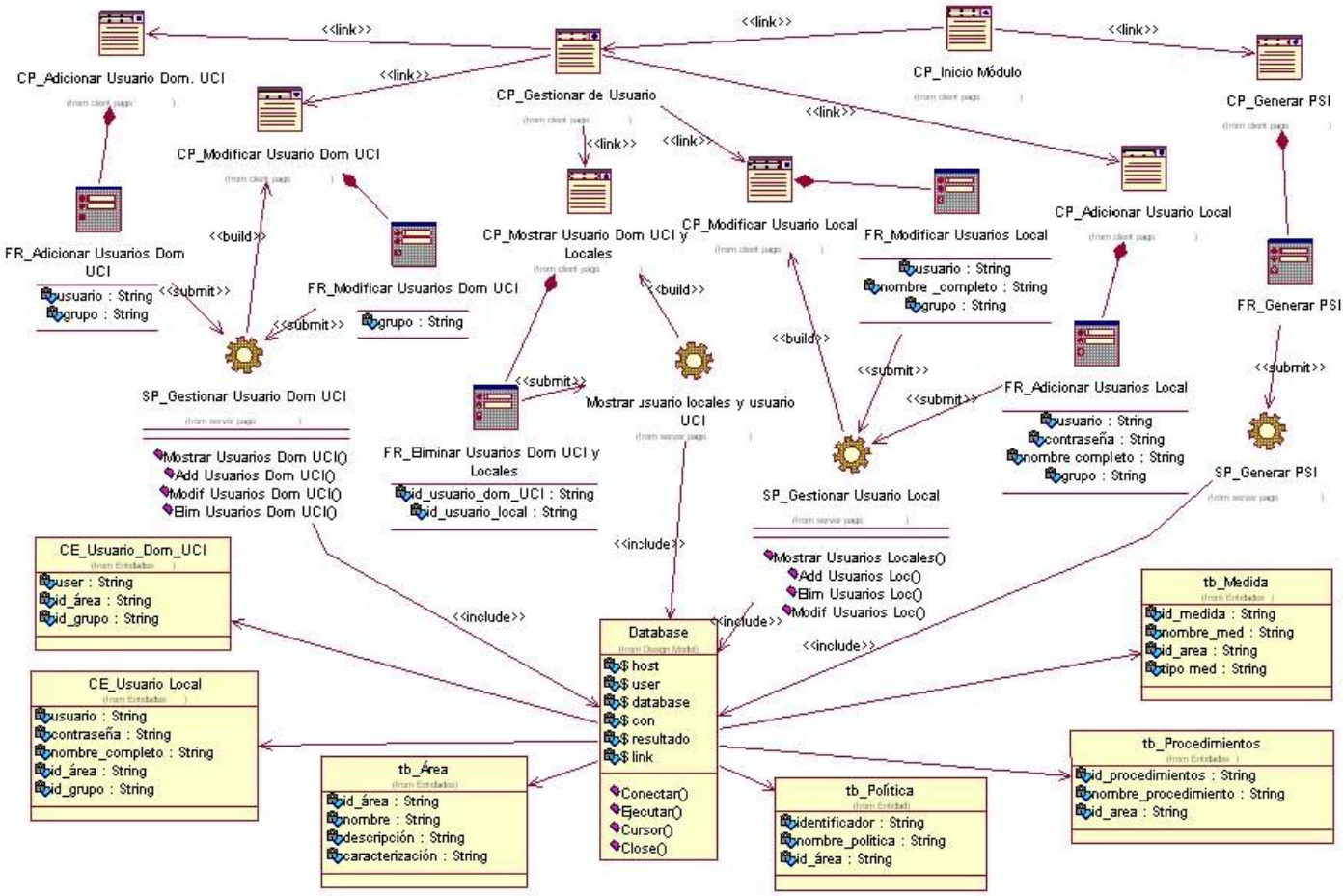


Figura 4.26 Diagrama de clases del diseño para el Módulo "Director de Seguridad Informática".

Módulo Administrador

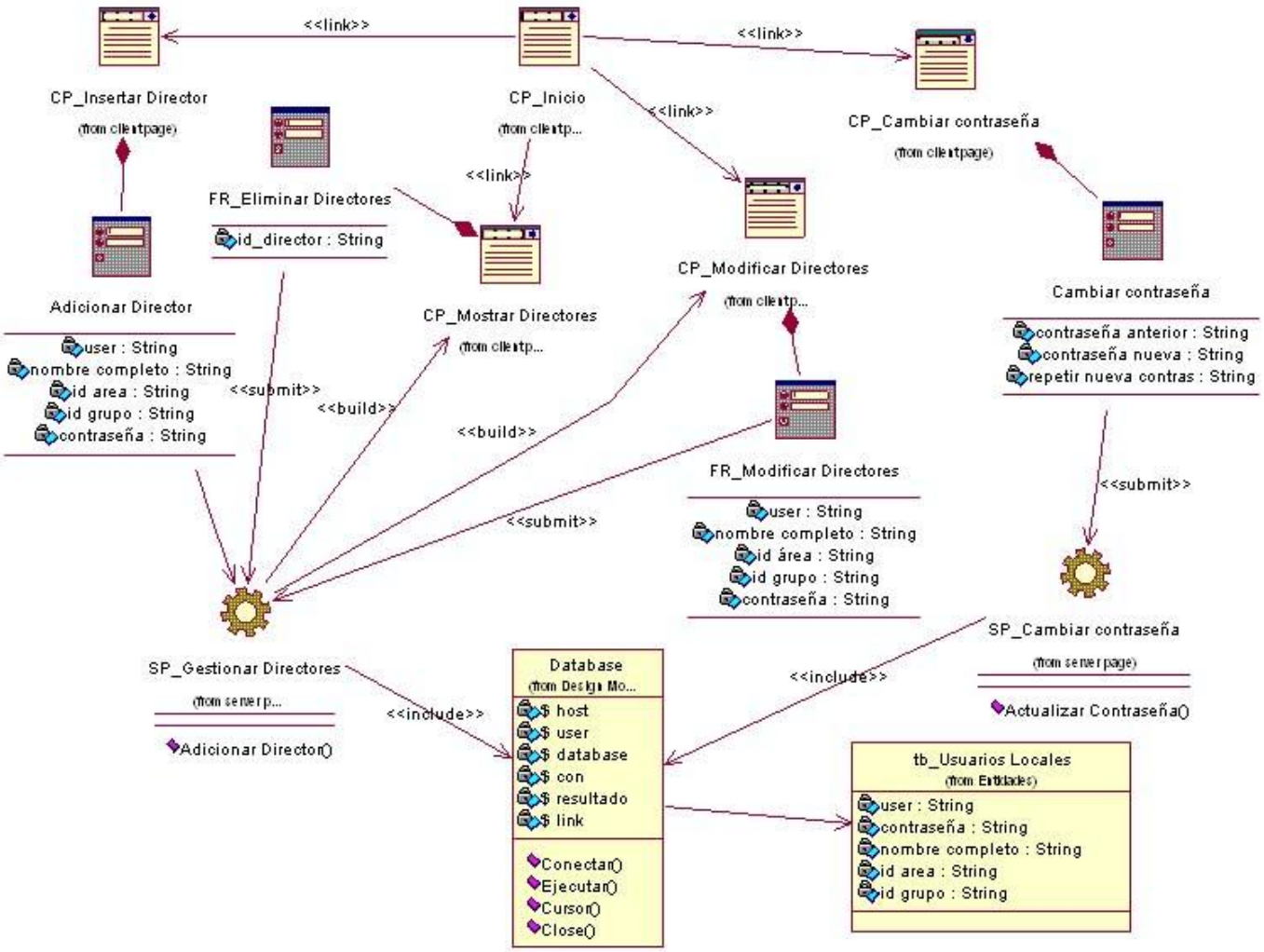


Figura 4.27 Diagrama de clases del diseño para el Módulo "Director de Seguridad Informática".

4.4 Patrones arquitectónico y de diseño empleados.

4.4.1 Patrón de diseño arquitectónico. Modelo - Vista - Controlador.

"Modelo-Vista-Controlador" es un patrón de diseño de arquitectura que está asociado a la idea de 3 capas (3 layers). El mismo se centra en la secuencia de ejecución, desde que se produce un evento en la capa de presentación hasta que el mismo es atendido en forma completa. (Ercoli, 2007)

Las partes que lo componen son:

Vista (View): Muestra la información al usuario. Pueden existir múltiples vistas del modelo.

Modelo (Model): Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada.

Controlador (Controller): componente asociado a la lógica de procesos del negocio. Reciben las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón, pulsaciones de teclas, etc. Los eventos son traducidos a solicitudes de servicio (“service requests”) para el modelo o la vista.

Desde la presentación de este patrón a la comunidad científica, se han desarrollado a lo largo de los años 3 variantes fundamentales, que se presentan brevemente a continuación.

Variante I: Variante en la cual no existe ninguna comunicación entre el Modelo y la Vista y esta última recibe los datos a mostrar a través del Controlador.

Variante II: Variante en la cual se desarrolla una comunicación entre el Modelo y la Vista, donde esta última al mostrar los datos la busca directamente en el Modelo, dada una indicación del Controlador, disminuyendo el conjunto de responsabilidades de este último.

Variante III: Variante en la que se diversifica las funcionalidades del Modelo teniendo en cuanta las características de las aplicaciones multimedia, donde tienen un gran peso las medias utilizadas en estas.

Para la realización de la arquitectura del software se empleó la Variante I del patrón MVC que se presenta en la siguiente figura.

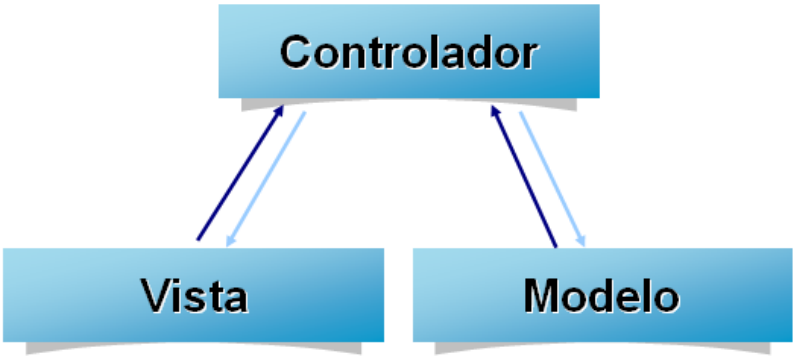


Figura 4.28 Variante I del MVC.

4.4.2 Patrones de diseño (GRASP).

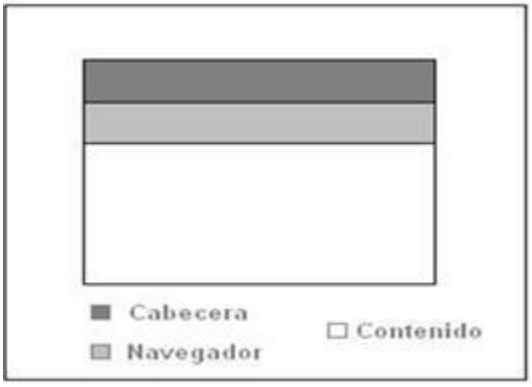
En el diseño del Asistente para la generación de Planes de Seguridad Informática se tuvieron en cuenta los patrones de asignación de responsabilidades conocidos como patrones GRASP (General Responsibility Asignment Software Patterns). Para el diseño de la aplicación se tuvieron en cuenta fundamentalmente el patrón Experto y el Controlador. El primero plantea que se debe asignar responsabilidades a la clase que cuenta con la información necesaria para cumplir responsabilidades, de esta forma el comportamiento se distribuye entre las clases que cuentan con la información requerida alentando con ellos clases sencillas y más fáciles de comprender. El segundo expresa asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase controladora.

4.5 Principios del diseño.

4.5.1 Estándares en la interfaz de la aplicación.

El diseño de interfaces de usuario es una tarea que ha adquirido relevancia en el desarrollo de un sistema. La calidad de la interfaz de usuario puede ser uno de los motivos que conduzca a un sistema al éxito o al fracaso, es por eso que uno de los aspectos más relevantes de la usabilidad de un sistema es la consistencia de su interfaz de usuario.

Para el diseño de la interfaz de la aplicación se utiliza en todas las páginas el esquema Cabecera-Navegador-Contenido. La cabecera contiene el nombre de la aplicación en la parte izquierda superior a modo de logo. En el navegador se incluyen los enlaces, en forma de pestañas, a las distintas secciones. En el área del contenido se muestran los formularios de entrada, las informaciones que se deben mostrar, los reportes, etc.



Se utilizan para el diseño las tablas y plantillas, dado que son cien por ciento compatibles con todos los navegadores, también se emplea hojas de estilos para guardar la configuración del diseño de todas las páginas. Estas hojas de estilos establece el tipo, tamaño de fuente de los distintos elementos de cada página, el color de los vínculos, el color de fondo, el formato de los controles de formulario y las tablas, entre otros.

4.5.2 Formato de los reportes.

Los reportes se muestran en formato de documento .pdf. Exponen la información organizada en un documento que incluye presentación, caracterización del sistema informático, las políticas de seguridad que se establecen en el área, así como las medidas y procedimientos de seguridad.

4.5.3 Tratamiento de excepciones.

Para reducir la ejecución de errores se utilizó JavaScript, evitando así que el servidor Web procese la página en vano. Este es el caso de los formularios de inserción/actualización, y las eliminaciones.

4.5.4 Estándares de codificación.

Para lograr un mejor entendimiento del código en la implementación de la aplicación es necesario establecer un estándar de codificación a usar. En la política seguida al respecto todas las variables y nombres de funciones a utilizar se definieron en idioma español. Los inicios ({) y cierre (}) de ámbito se encuentran alineados debajo de la declaración a la que pertenecen. Se usa una línea propia para {. Los signos lógicos y de operación se separan por un espacio antes y después de los mismos. Los nombres de las variables utilizadas comienzan en minúscula y son cortos, claros y describen su propósito. Los comentarios se definen comenzando con los caracteres ***/**** y terminando con *****/*** para los comentarios de varias líneas, y comenzando con los caracteres ***//*** para los de una sola línea.

4.6 Diseño de la base de datos.

4.6.1 Diagrama de clases persistentes.

La persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo. El diagrama de clases persistentes permite modelar aspectos relacionados con el almacenado de datos del sistema. La figura 4.29 muestra el diagrama de clases persistentes del sistema que se propone.

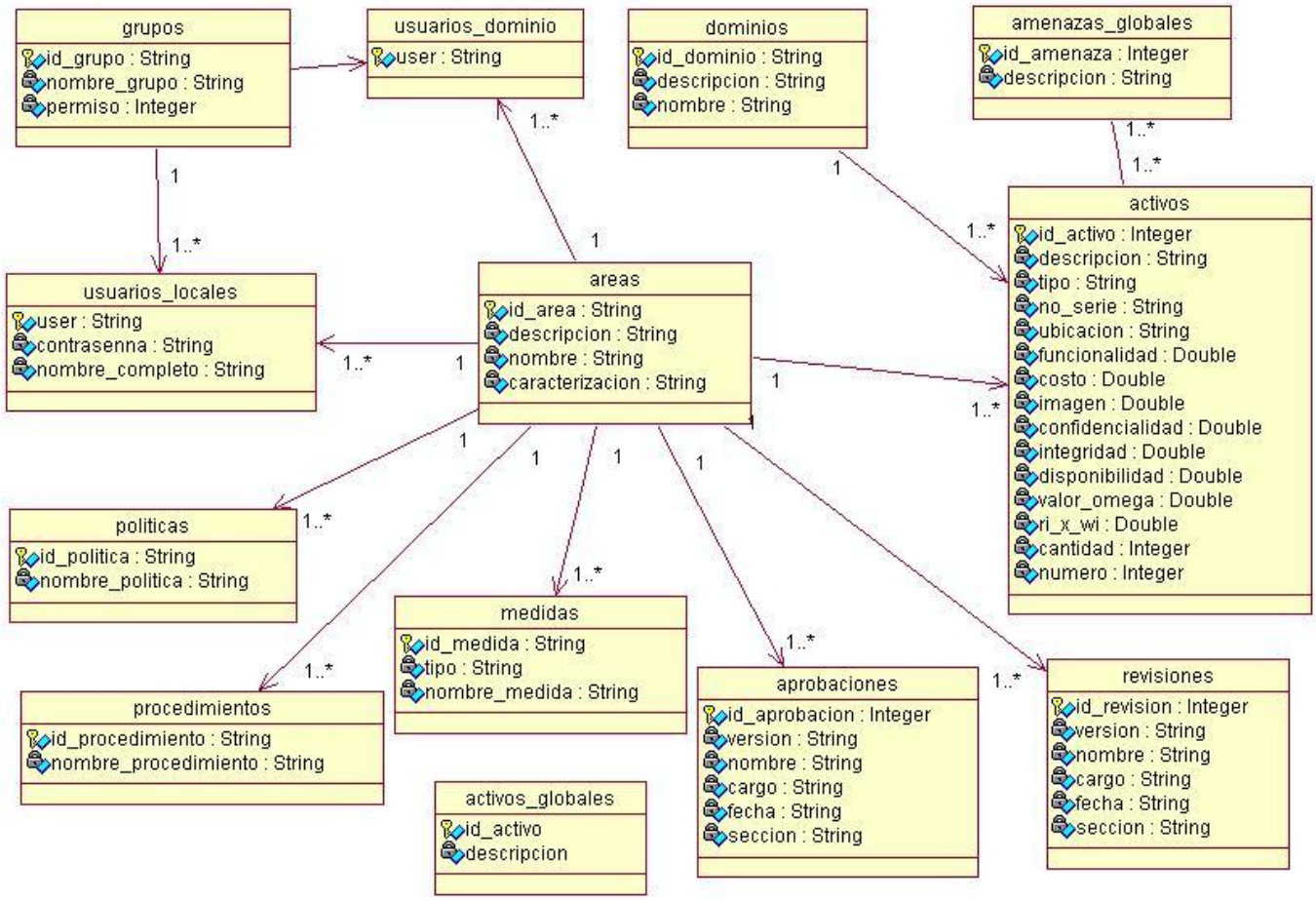


Figura 4.29 Diagrama de clases persistentes.

4.6.2 Modelo de Datos.

El modelo de datos es un conjunto de conceptos que permiten describir la estructura de una base de datos. Estos conceptos no son otros sino: los datos, las relaciones entre ellos y las restricciones que deben cumplirse sobre los mismos. La figura 4.30 ilustra el modelo físico de la base de datos.

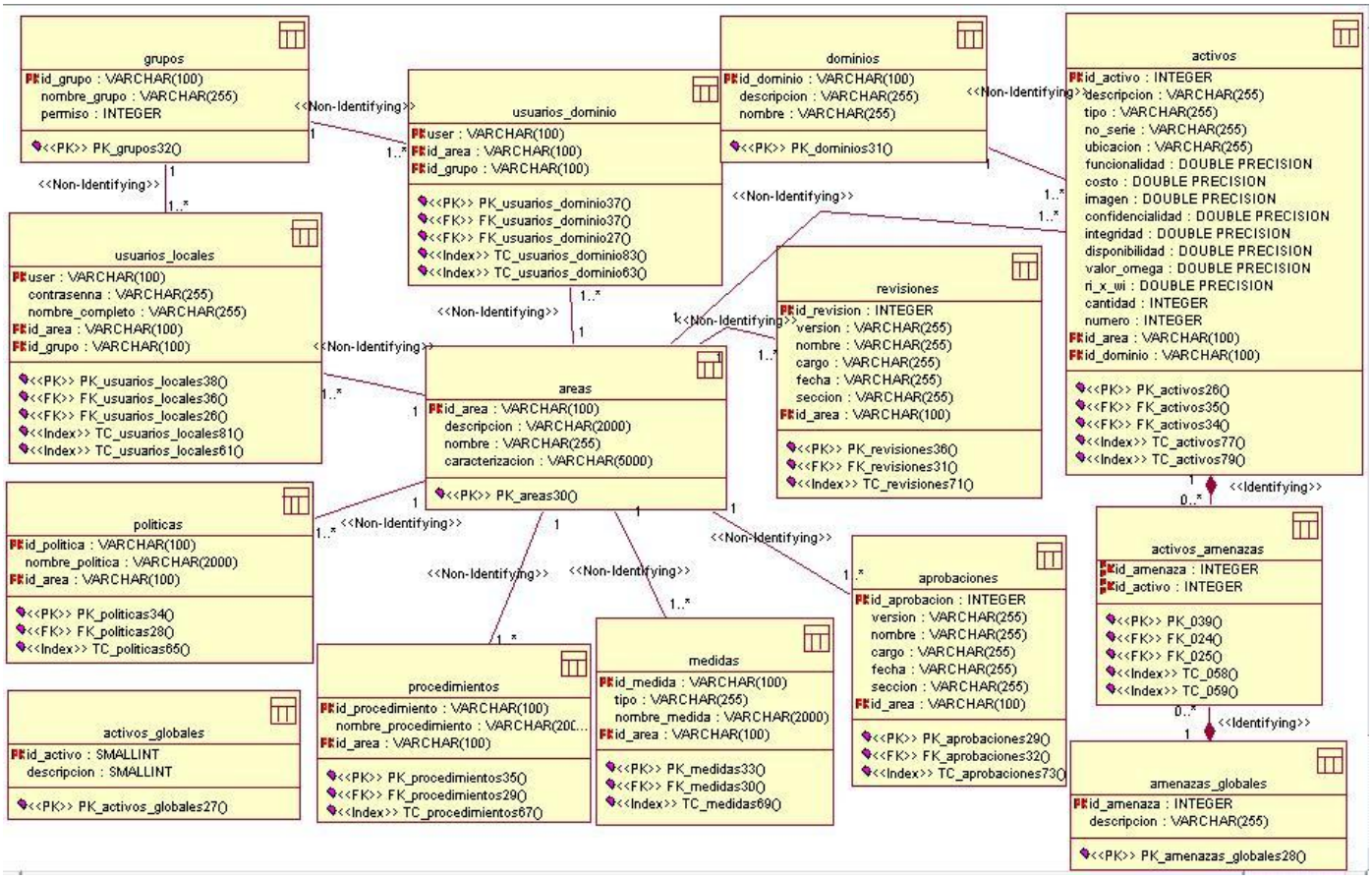


Figura 4.30 Modelo de Datos.

4.7 Modelo de Implementación.

El modelo de implementación describe cómo los elementos del modelo de diseño, como las clases, se implementan en términos de componentes, como ficheros de código fuente, ejecutables, etc. El modelo de implementación describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados y cómo dependen los componentes unos de otros. (Jacobson, Booch y Rumbaugh, 2004)

4.7.1 Diagrama de despliegue.

El diagrama de despliegue describe la arquitectura física del sistema durante la ejecución, en términos de: procesadores, dispositivos y componentes de software. Además ilustra la topología del sistema, es decir, la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos. La figura 4.27 muestra el diagrama de despliegue de la aplicación que se propone.

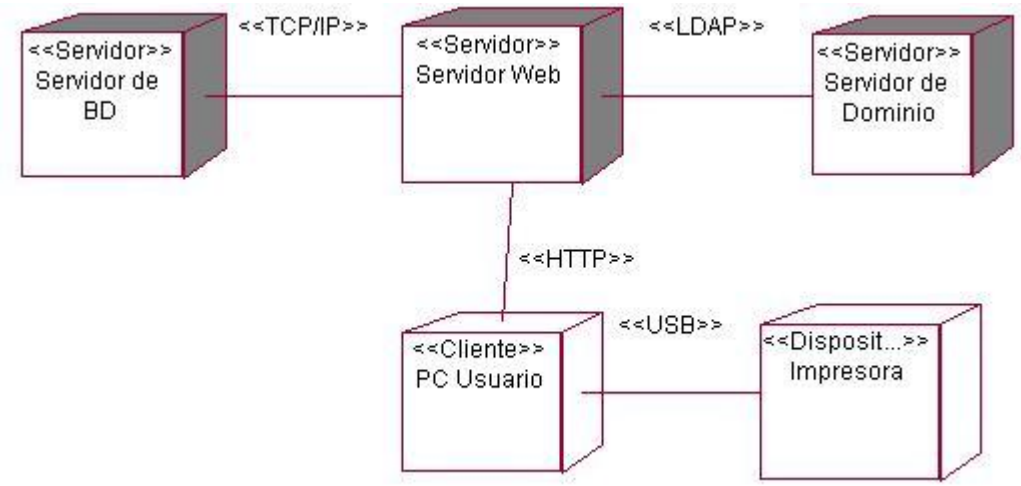


Figura 4.31 Diagrama de despliegue.

4.7.2 Diagrama de componentes.

Estos diagramas son los encargados de describir los elementos físicos del sistema y sus relaciones. Los componentes representan todos los tipos de software que tiene la aplicación. Es donde se presentan las opciones de ejecución conteniendo código fuente, binario y ejecutable. Se utilizan las relaciones de dependencia en dichos diagramas para especificar que un componente utiliza los servicios por otro componente.

A continuación se presentan los diagramas de componentes para los diferentes módulos del sistema que se propone y para la autenticación de los usuarios.

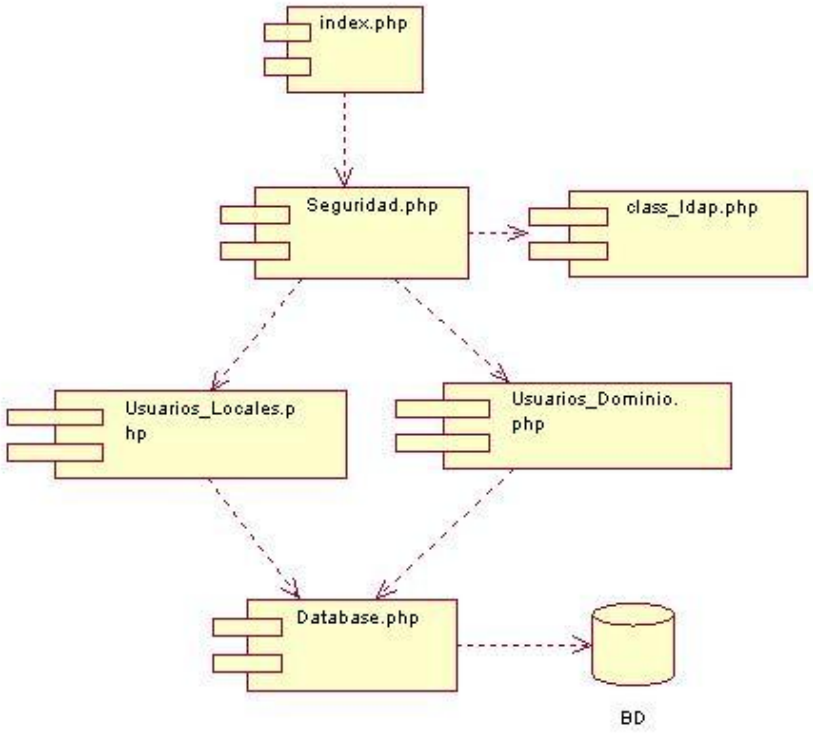


Figura 4.32 Diagrama de componentes para el caso de uso "Autenticar Usuario".

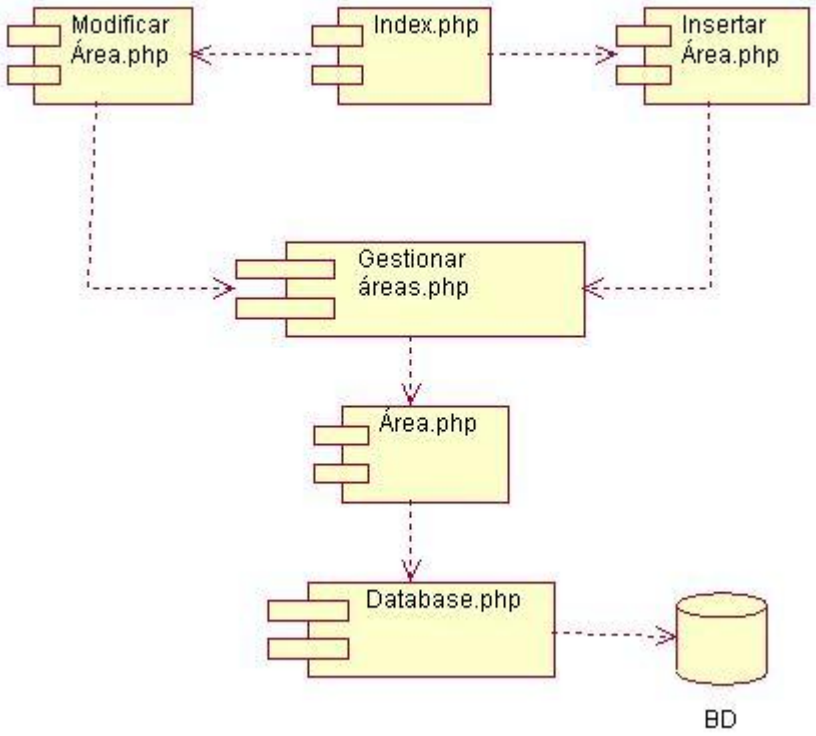


Figura 4.33 Diagrama de componentes para el Módulo "Responsable de Áreas".

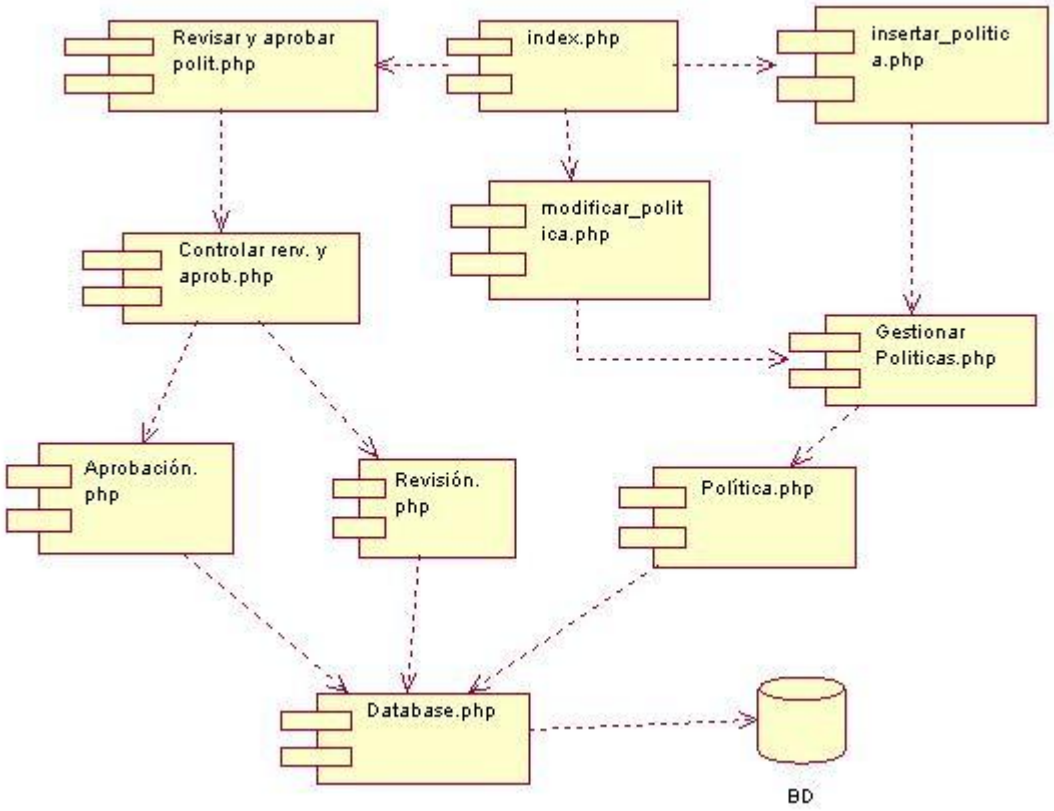


Figura 4.34 Diagrama de componentes para el Módulo "Responsable de Políticas".

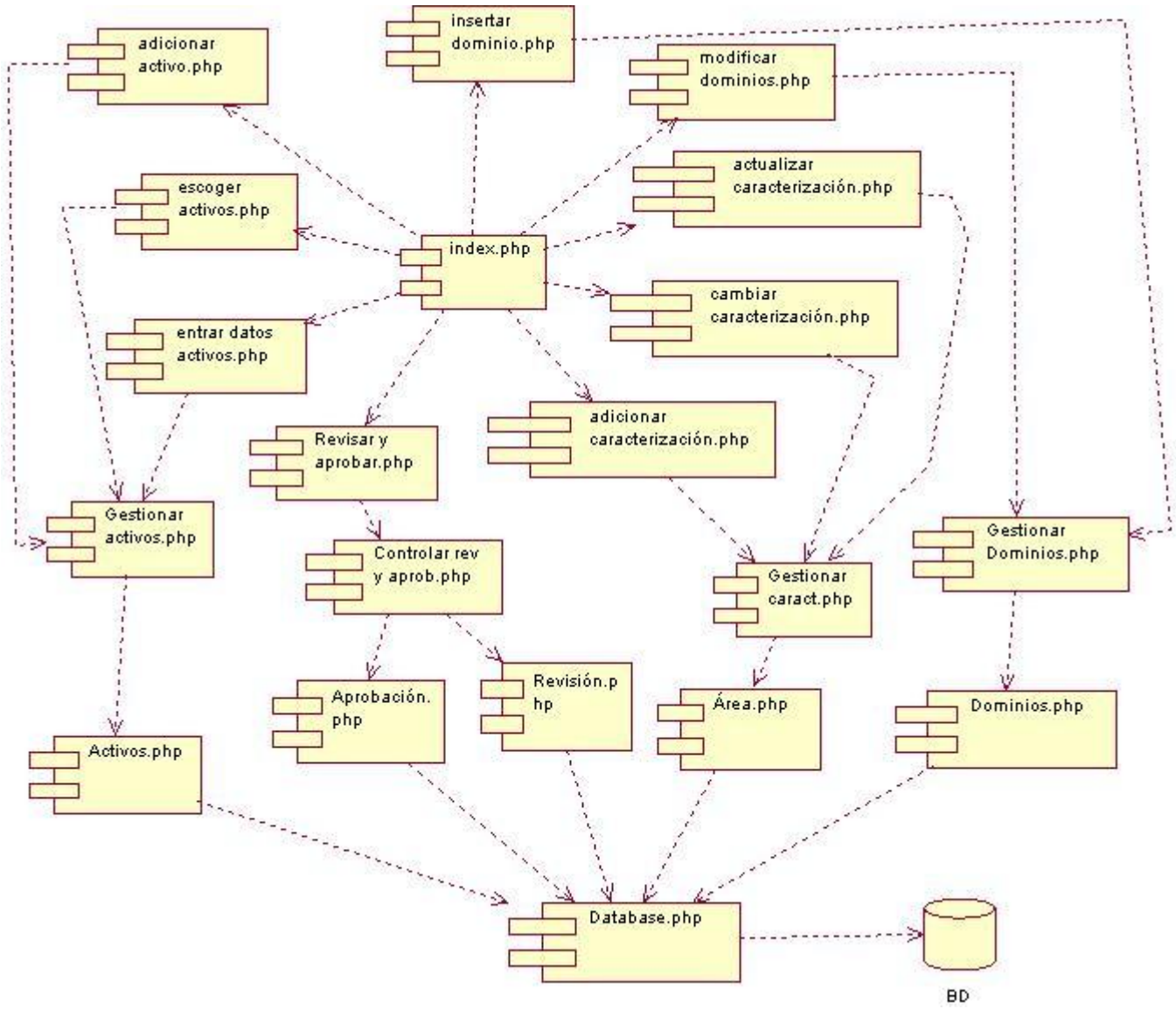


Figura 4.35 Diagrama de componentes para el Módulo "Responsable de Control de Inventario".

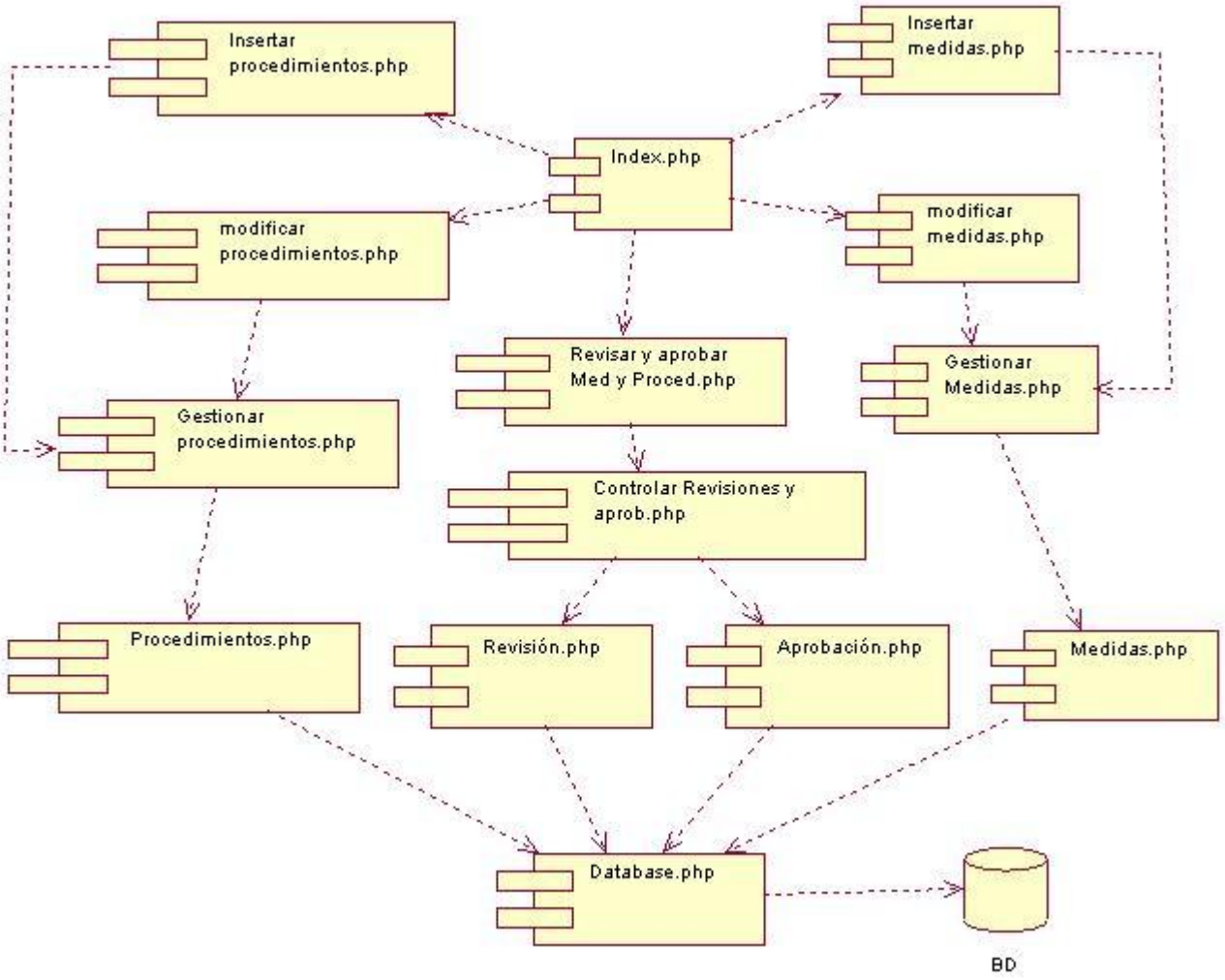


Figura 4.36 Diagrama de componentes para el Módulo "Ingeniero de Sistemas".

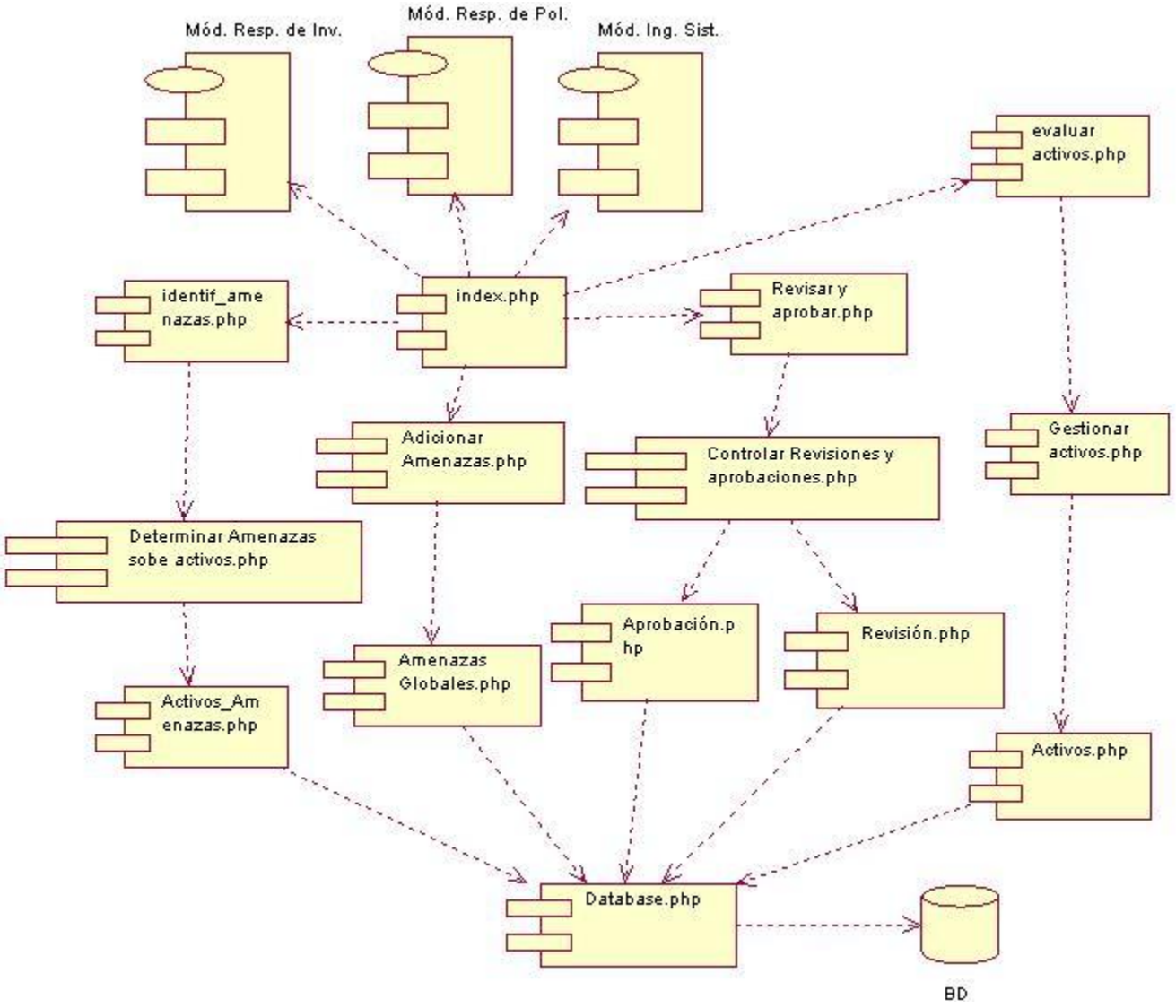


Figura 4.37 Diagrama de componentes para el Módulo "Especialista de Análisis de Riesgo".

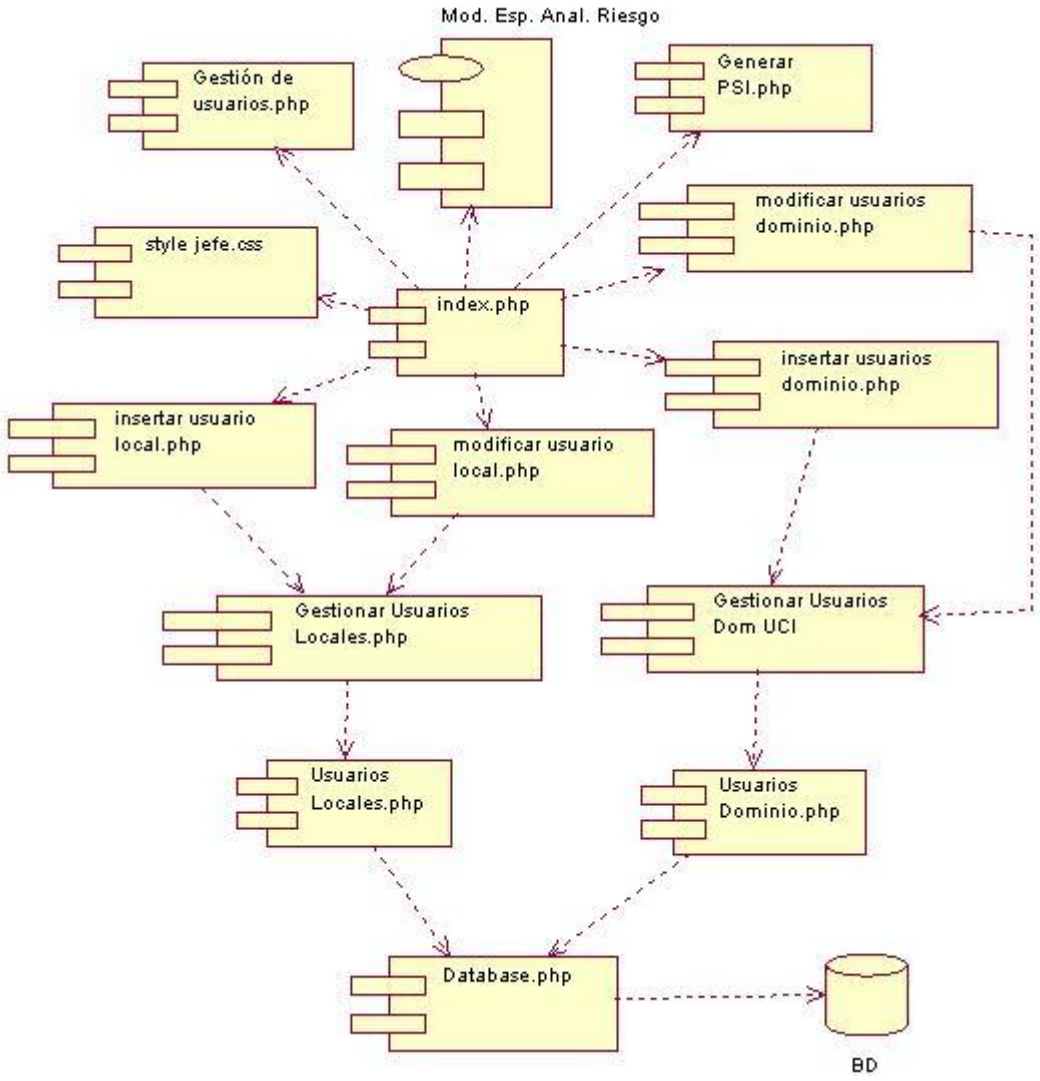


Figura 4.38 Diagrama de componentes para el Módulo "Director de Seguridad Informática".

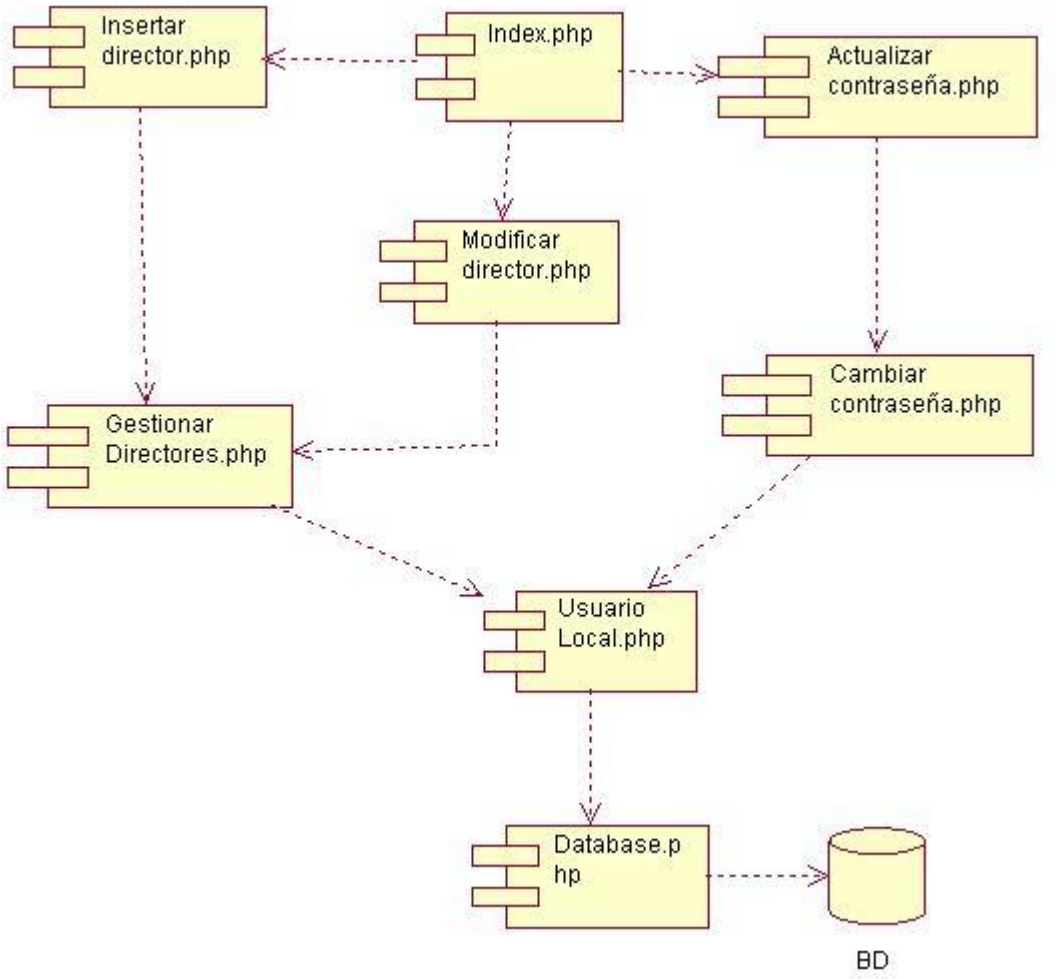


Figura 4.39 Diagrama de componentes para el Módulo "Administrador".

CONCLUSIONES

Después de culminar la investigación realizada para la implementación del Asistente para la generación de Planes de Seguridad Informática, teniendo en cuenta que el desarrollo estuvo guiado por las tareas de investigación, se llegó a las siguientes conclusiones:

- ✓ Se dio solución al objetivo general trazado a partir del cumplimiento de todas las tareas de investigación previstas.

- ✓ La utilización de este sistema contribuye al fortalecimiento del proceso de desarrollo de los Planes de Seguridad Informática y su puesta en práctica elevará la calidad de los mismos.

- ✓ Se obtuvo un sistema multiplataforma y adaptable para otras entidades.

RECOMENDACIONES

- ✓ Implementar la funcionalidad Gestionar Plan de Contingencia en el módulo para el Ingeniero de Sistemas desde la modelación, el Plan de Contingencia se anexa al Plan de Seguridad Informática pero se genera de forma independiente.
- ✓ Optimizar la gestión del control de versiones según el modelo propuesto, pues en esta primera versión del sistema no se lleva a cabo un proceso de gestión de las versiones del plan de seguridad informática.
- ✓ Extender el uso del software desarrollado a otras entidades cubanas.
- ✓ Utilizar el software desarrollado como un medio de apoyo a la docencia en el 4to año de la carrera para la asignatura Seguridad Informática.

REFERENCIAS BIBLIOGRÁFICAS

1. Álvarez, Miguel Ángel. 2003. Desarrollo Web. *Zend Studio. Editor web orientado a la programación de páginas PHP, con ayudas en la gestión de proyectos y depuración de código.* [En línea] 2003. http://www.desarrolloweb.com/articulos/1178.php.

2. Calle Guglieri, José A. 2005. Reingienería y Seguridad en el Ciberespacio . [En línea] 2005. http://books.google.com.cu/books?id=qB3P2GuD3EsC&pg=PA58&lpg=PA58&dq=Cramm+%2B+analisis+de+riesgo&source=bl&ots=uOo6zIkaEz&sig=k8R5kLDiFzTfRWVMUaePKZzdwzk&hl=es&ei=PfkXSt-hNM_gtgeVi5z_DA&sa=X&oi=book_result&ct=result&resnum=3#PPP1,M1.

3. CERT. 2008. . *OCTAVE.* [En línea] 2008. http://www.cert.org/octave/.

4. Collector, Garbage. 2004. *Sistema Gestor de base de datos SGBD.* [En línea] 2004. http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php.

5. Domínguez, Germán. 2008. *Aplicaciones Web Seguras ¿Mito o Realidad?* [En línea] 2008. http://www.sg.com.mx/content/view/793.

6. Eguíluz Pérez, Javier. Libros Web. *Introducción a JavaScript.* [En línea] http://www.librosweb.es/javascript/.

7. Ercoli, Jorge. 2007. Metodologías de Sistemas. *Arquitectura de Sistemas Informáticos, Diseño en 3 capas? físicas ó lógicas? es igual a patrón MVC?* [En línea] 2007. http://metodologiasdesistemas.blogspot.com/2007/05/diseo-en-3-capas-fisicas-lgicas-es.html.

8. Expósito Gutiérrez, Francisco. 2003. *Metodología de análisis y gestión de riesgos: MAGERIT. Seguridad en Redes Telemáticas*. 2003.

9. IBM. 2001. *Rational Unified Process, Best Practices for Software Development Teams.* [En línea] 2001. http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf.

10. Jacobson, Ivar; Booch, Grady y Rumbaugh, James. 2004. *El proceso unificado de desarrollo de software.* Ciudad de La Habana : Editorial Félix Varela, 2004.

11. López, Ángel. 2008. *Metodología de Desarrollo SCRUM.* [En línea] 2008. http://www.fcad.uner.edu.ar/jai/7JAI/confe_scrum.htm.

12. MASTERMAGAZINE. 2004.. *Seguridad Informática.* [En línea] 2004. http://www.mastermagazine.info/termino/6638.php.

13. Mendoza Sánchez, María A. 2004. Informatízate. *Metodologías De Desarrollo De Software.* [En línea] 2004. http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.

14. Ministerio del Interior. 2008. *Metodología para la elaboración del Plan de Seguridad Informática.* [En línea] 2008. http://files.sld.cu/gau/files/2008/10/metodologiapsi.doc.

15. MySQL. 2007. *MySQL 5.0 Reference Manual.* [En línea] 2007. http://dev.mysql.com/doc/refman/5.0/es/index.html.

16. Otrera, David. 2008. Educar. *Redes Informáticas.* [En línea] 2008. http://portal.educ.ar/debates/eid/informatica/para-trabajar-clase/web-quest-redes-informaticas.php.

17. PREMIER MINISTRE. 2004. *EBIOS.* [En línea] 2004. http://www.ssi.gouv.fr/es/confianza/documents/methods/ebiosv2-memento-2004-02-04_es.pdf.

18. Rational. 2007. *Rational Rose Enterprise.* [En línea] 2007. http://www.rational.com.ar/herramientas/roseenterprise.html.

19. Resolución 127.2007. Artículo 6 de la resolución, La Habana, a los 24 días del mes de Julio de 2007, Ramiro Valdés Menéndez, Ministro de la Informática y las Comunicaciones.

20. Rodríguez Aneiro, Luis Orlando. 2001. *Elementos de arquitectura y seguridad informática* . s.l. : Pueblo y Educación., 2001.

21. Rodríguez, Luis. 2006. *Seguridad en la Base de Datos: Ficciones y fricciones.* [En línea] 2006. http://www.als-es.com/home.php?location=recursos/articulos/seguridad-en-bases-de-datos.

22. Thomson, Laura y Welling, Luke. 2003. *Desarrollo web con PHP y MySQL.* s.l. : Ediciones Anaya-Multimedia, 2003.

23. Usero Martínez, José Ángel. 2007. *Nuevas tecnologías para nuevas bibliotecas. Desarrollo de servicios de información electrónica.* s.l. : Alfragama Ediciones, 2007.

24. Weitzenfeld, Alfredo. 2004. *Ingeniería de Software Orientada a Objetos con UML, Java e Internet.* 2004.

BIBLIOGRAFÍA

Álvarez, Miguel Ángel. Desarrollo Web. *Dreamweaver, Probablemente el mejor editor de páginas web para diseñadores que busquen resultados profesionales.* [En línea] http://www.desarrolloweb.com/articulos/332.php.

Boggs, Wendy y Boggs, Michael. 2002. *UML with Rational Rose*. 2002.

Caralli, Richard A., y otros. 2007. *Introducing OCTAVE Allegro: Improving the Information Security Risk*. 2007.

Córdova Rodríguez, Norma Edith. 2007. SISBIB (Sistema de Biblioteca). *Plan de Seguridad Informática para una entidad financiera.* [En línea] 2007. http://sisbib.unmsm.edu.pe/bibvirtual/tesis/Basic/cordova_rn/contenido.htm.

Escobar, Luis y Sánchez., Mar. 2007. Isms. [En línea] 2007. https://www.ismsforum.es/noticia.php?noticia=16.

Espinoza, Humberto. 2005. *PostgreSQL Una Alternativa de DBMS Open Source*. [En línea] 2005. http://www.lgs.com.ve/pres/PresentacionES_PSQL.pd.

European Network and Information Security Agency (ENISA).2007. *Risk Management & IT Security*. 2007.

Hernán Ruiz, Marcelo. 2007. *Programación Web avanzada, Soluciones rápidas y efectivas para desarrolladores de sitios*. La Habana: Félix Varela, 2007.

InfoCitel. 2005. *¿Qué es la seguridad informática?* [En línea] 2005. http://www.citel.oas.org/newsletter/2005/septiembre/seguridad_e.asp Boletín electrónico /

JavaSript. 2007. *JavaScript. Manual de JavaScript*. [En línea] 2007. http://manualdejavascript.com/manualjavascript/introduccion.html.

Maestros del Web. 2004. *Encriptación de contraseña con MD5*. [En línea] 2004. http://www.maestrosdelweb.com/editorial/md5/.

Molpeceres, Alberto. 2002. *Proceso de desarrollo: RUP, XP y FDD*. [En línea] 2002. http://www.willydev.net/descargas/Articulos/General/cualxpfddrup.PDF.

MySQL.2008. *Why MySQL?* [En línea] 2008. http://www.mysql.com/why-mysq.

MySQL. 2007. *The world's most popular open source database*. [En línea] 2007. http://dev.mysql.com/doc/refman/5.0/es/index.html.

Quiñones, Ernesto. *PostgreSQL. Introducción a PostgreSQL*. [En línea] http://www.postgresql.org.pe/articles/introduccion_a_postgresql.pdf.

Schwartz, Jonathan. 2008. *Ya es oficial: MySQL forma parte de Sun*. [En línea] 2008. http://blogs.sun.com/jonathan_es/.

Segurmática, 2009. *Consultoría y Gestión de la Calidad en Seguridad Informática. Planes de Seguridad y Contingencia Informática.* [En línea] 2009. http://www.segurmatica.co.cu/consultoria/.

ENISA Study. 2007. *Emerging-risks-related Information Collection and Dissemination*. 2007.

Wisegeek. *What is Information Security?* [En línea] http://www.wisegeek.com/what-is-information-security.html.

Woody, Carol. 2006. *Applying OCTAVE: Practitioners Report.* 2006.

ANEXOS

Anexo 1. Pasos para realizar Análisis de Riesgo manualmente.

Paso 1. Confeccionar el listado de activos.

No	Descripción	Tipo	Ubicación

Tabla.1 Listados de los activos.

Donde cada columna significa lo siguiente:

No: Número en orden consecutivo de los bienes informáticos.

Descripción: Descripción de los bienes informáticos.

Tipo: Tipo de bienes informáticos:

RD: Redes de diferentes tipos.

GD: Sistemas de gestión de datos.

CP: Sistemas de control de procesos.

OT: Otros tipos de aplicaciones o sistemas.

HW: Hardware.

SW: Software.

Ubicación: Local donde se encuentran los bienes informáticos.

Paso 2. Realizar el cálculo de la importancia de cada activo.

La determinación de la importancia de los bienes informáticos puede ser realizada de forma descriptiva (por ejemplo, valor alto, medio, bajo) o de forma numérica asignando valores entre cero y diez (0 si no tiene importancia y 10 sí es máxima). La forma numérica tiene la ventaja de que permite estimar el nivel de riesgo con mayor rigor, así como la valoración por áreas o grupos de elementos más fácilmente.

Relación entre los métodos descriptivos y numéricos:

0 – 3.5: Importancia baja

3.6 – 5.9: Importancia media

6.0 – 7.9: Importancia alta

8.0 – 10: Importancia muy alta.

No	Dom	Valoración por aspectos						Importancia (Wi)
		Función	Costo	Imagen	Confiden.	Integrid.	Disponib.	

Tabla.2 Evaluación de los bienes informáticos.

En cada una de las filas de esta tabla se relacionan los bienes informáticos identificados en la tabla 1 a fin de facilitar la evaluación de cada uno de ellos.

El significado de cada una de las columnas es el siguiente:

No: Número de bienes informáticos obtenidos de la tabla 1.

Dominio: Identificación para agrupar bienes informáticos afines por las funciones que realizan y/o por la administración sobre ellos. (D1, D2... Dn, según la cantidad que se cree).

Valoración por aspectos:

- ✓ **Función:** Importancia de la tarea que cumplen los bienes informáticos.
- ✓ **Costo:** Precio y valor de uso de los bienes informáticos.
- ✓ **Imagen:** Repercusión interna y/o externa que ocasionaría la pérdida de los bienes informáticos.
- ✓ **Confidencialidad:** Necesidad de proteger la información que de los bienes informáticos pueda obtener.
- ✓ **Integridad:** Necesidad de que la información no se modifique o destruya.

- ✓ **Disponibilidad:** Que los servicios que de bienes informáticos se esperan puedan ser obtenidos en todo momento de forma autorizada.
- ✓ **Importancia (Wi):** Importancia de los bienes informáticos.

A las columnas función, costo, imagen, confidencialidad, integridad, y disponibilidad se le asignan valores entre 0 y 10, en dependencia de la estimación que se haga de la importancia de cada uno de estos factores sobre los bienes informáticos analizados (0 sino tiene importancia y 10 si es máxima).

El valor de la columna **(Wi)** se obtiene por el promedio de los valores estimados de la valoración por aspectos (columnas 3 – 8), es decir, el resultado de la suma de éstas dividido por 6. La suma total **(Wt)** de los valores (Wi) obtenidos representa la importancia total de los bienes informáticos que componen el sistema:

Wt = W1 + W2 +..... + Wn

Paso 3. - Realizar un análisis de los bienes informáticos críticos.

Como resultado de la evaluación anterior se deben determinar los bienes informáticos que se consideran críticos para el Organismo, sin los cuales el trabajo no puede ser ejecutado o es afectado de forma sensible.

Lo anterior implica un análisis complementario de los datos obtenidos de la Tabla.2, que podría realizarse de la forma siguiente:

• Señale adecuadamente aquellos bienes informáticos que fueron valorados de importancia significativa.

• Señale aquellos bienes informáticos, que no habiendo sido valorados de importancia significativa, tienen una incidencia directa con algún otro crítico.

• Señale, después de un estudio riguroso y detallado, aquellos bienes informáticos que no teniendo una valoración significativa, ni incidencia directa en el trabajo de activos y recursos críticos, resulta necesario que sean marcados como tales, por razones prácticas.

Se debe actualizar la tabla No.2 a partir de las consideraciones anteriores.

Paso 4. Identificación de amenazas y estimación de riesgos.

Una vez que los bienes informáticos que requieren protección son identificados y valorados según su importancia es necesario identificar las amenazas sobre éstos y estimar la pérdida potencial (impacto) que puede producir su materialización.

Para esto se debe llenar la tabla siguiente:

No	Amenazas	Bienes Informáticos													
		1	2	3	4	5	6	.	.	..	.	..	..	..	N

Tabla 3 Identificación de amenazas

La realización de un análisis de riesgos implica el examen de cada una de las amenazas sobre los bienes informáticos y su clasificación por niveles, a partir de la probabilidad de su ocurrencia y la severidad del impacto que puedan producir.

A partir de las amenazas identificadas se cuantifica el riesgo de que cada una de ellas se materialice sobre cada uno de los bienes informáticos, esto puede ser realizado de forma descriptiva (por ejemplo, riesgo alto, medio, bajo) o de forma numérica asignando valores entre cero y uno (0 sí la probabilidad de que se materialice la amenaza es nula y 1 sí es máxima).

Relación entre los métodos descriptivos y numéricos:

0 – 0.35: Riesgo bajo

0.36 – 0.59: Riesgo medio

0.60 – 0.79: Riesgo alto

0.80 – 1.0: Riesgo muy alto

A partir de las amenazas identificadas en la Tabla 3 se cuantifica el riesgo de que cada una de ellas se materialice sobre cada uno de los bienes informáticos, con ayuda de la Tabla 4.

No	Dom	Amenazas												Riesgo Ri	Impot. Wi	Peso Ri * Wi
		R 1	R2	R3	..	..	..	..	..	..	..	….	……	RN			

1	2	2	31	32	..	..	…	..	…	..	….	…	…..	3n	4	5	6
2																	
3																	
.																	
.																	
N																	

Tabla. 4 Estimación de los riesgos sobre los bienes informáticos

• Las columnas 1 y 2 (Número de orden y Dominio) corresponden con las de la Tabla 2.

• Las columnas 3, 31, 32,....., 3n reflejan la probabilidad de que se materialicen las amenazas identificadas en la Tabla 3 sobre cada bien informático, asignando valores entre 0 y 1.

• La columna 4 es la valoración del riesgo sobre cada bien informático. Se calcula a partir del promedio de las columnas 3, 31, 32,….., 3n que tomaron valor, es decir, la suma de los valores de esas columnas entre la cantidad de columnas.

• La columna 5, Importancia del bien informático, se obtiene de los valores estimados en la columna 9 de la Tabla 2.

• La columna 6, Peso del Riesgo sobre cada bien informático, se obtiene como resultado de la multiplicación de los valores de las columnas 4 y 5.

La suma de los Pesos Relativos de Riesgos sobre todos los bienes informáticos caracteriza el Peso Total del Riesgo del Sistema (Rt).

De manera que el Peso Total del Riesgo del Sistema (Rt) se puede obtener dividiendo la suma total de los valores de la columna 6 (Ri * Wi) por los de la columna 5 (Wi).

Anexo 2 Descripción textual del caso de uso del negocio "Realizar el Análisis de Riesgo".

Caso de Uso	Realizar el Análisis de Riesgo.
Actores	Director de Seguridad Informática.
Trabajadores:	Especialista de Análisis de Riesgo
Resumen:	El caso de uso se inicia cuando el Director de Seguridad Informática solicita al Especialista de Análisis de Riesgo el documento con el Análisis de Riesgo de la entidad. El Especialista de Análisis de Riesgo solicita al Responsable de

	Control de Inventario el listado de activos de la entidad. El Especialista de Análisis de Riesgo recibe la información de los activos y comienza a realizar la estimación de riesgo. El caso de uso termina cuando el Director de Seguridad Informática recibe el documento con el Análisis de Riesgo.
Precondiciones:	-

Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Negocio**
1. Solicita el Análisis de Riesgo de la entidad.	2. El Especialista de Análisis de Riesgo solicita el listado de los activos informáticos al Responsable de Control de Inventario. 3. El Responsable de Control de Inventario revisa el listado de los activos y lo entrega. 4. El Especialista de Análisis de Riesgo recibe la información. 5. El Especialista de Análisis de Riesgo calcula la importancia de los activos. 6. El Especialista de Análisis de Riesgo determina las amenazas que influyen sobre cada activo. 7. El Especialista de Análisis de Riesgo realiza la estimación de riesgo sobre los bienes informáticos. 8. El Especialista de Análisis de Riesgo obtiene los resultados del análisis de riesgo. 9. El Especialista de Análisis de Riesgo recoge toda la información en el documento de Análisis de Riesgo.

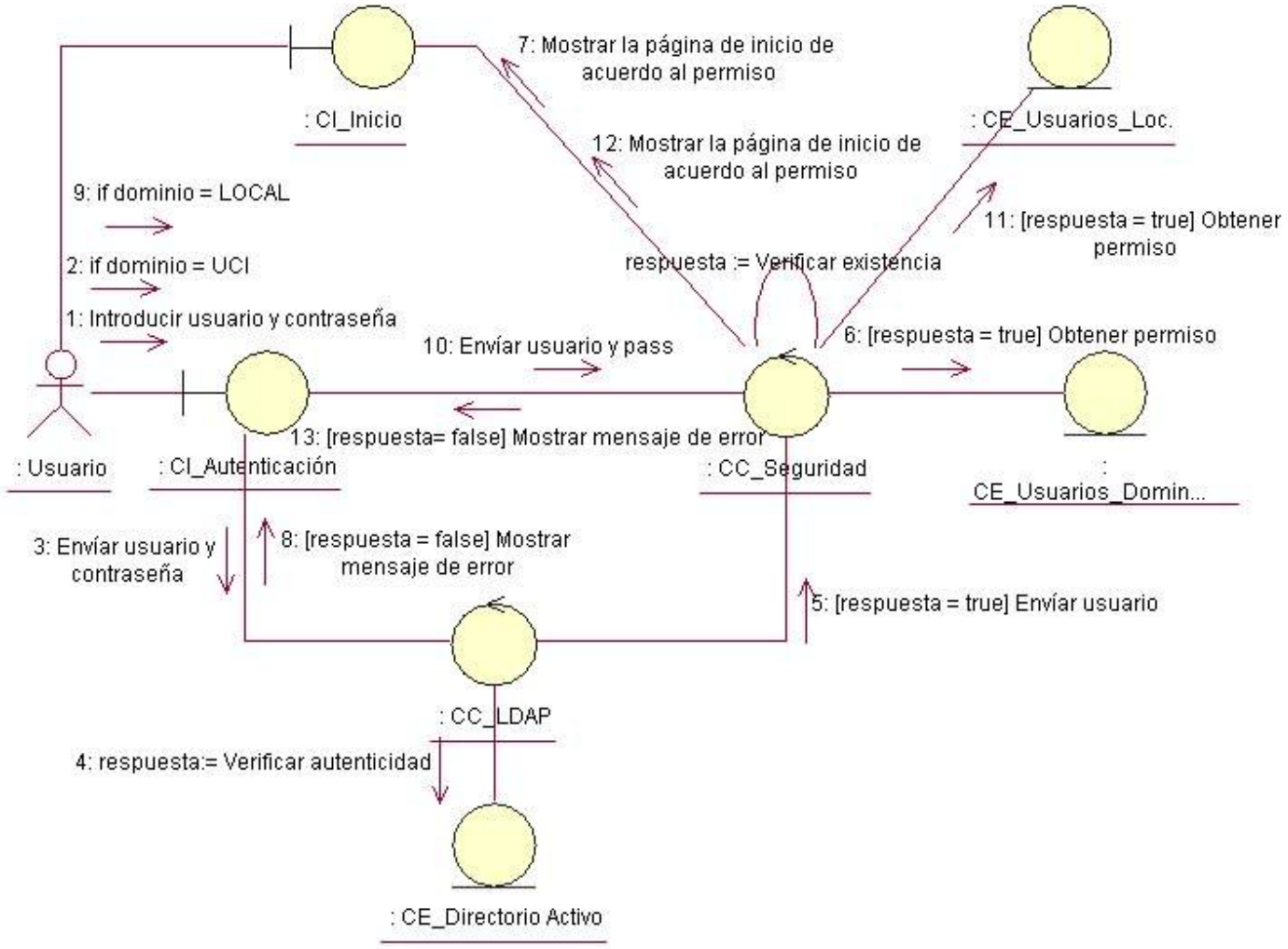
11. Recibe la Información del Análisis de Riesgo.	10. El Especialista de Análisis de Riesgo entrega la información del Análisis de Riesgo.
Mejoras	Registrar en una herramienta todo el proceso del análisis de riesgo permitirá que no haya errores de cálculo a la hora de realizar la importancia de los activos y realizar la estimación de riesgo.

Anexo 3 Descripción textual del caso de uso del negocio "Elaborar el PSI".

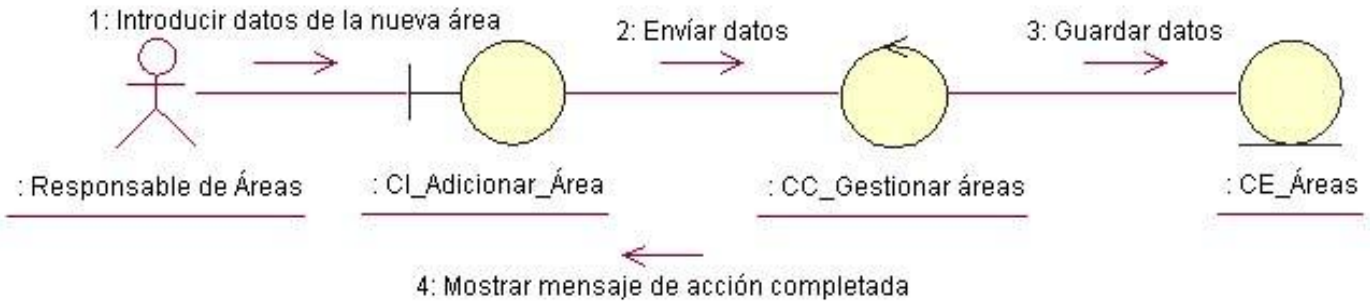
Caso de Uso	Elaborar el PSI
Actores	Director de Seguridad Informática.
Trabajadores:	Especialista de Seguridad.
Resumen:	El caso de uso se inicia cuando el Director de Seguridad Informática le entrega al Especialista de Seguridad el resultado del Análisis de Riesgo y solicita el Plan de Seguridad Informática. El Especialista de Seguridad elabora en la plantilla del PSI todo lo referente a este proceso y lo entrega al Director de Seguridad. El caso de uso termina cuando el Director de Seguridad Informática recibe el Plan de Seguridad.
Precondiciones:	Tener el resultado del análisis de riesgo.
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Negocio**
1. Entrega el resultado del Análisis de Riesgo y solicita el PSI.	2. El Especialista de Seguridad recibe los resultados del Análisis de Riesgo. 3. El Especialista de Seguridad elabora las políticas de seguridad. 4. El Especialista de Seguridad establece las medidas de seguridad.

	5. El Especialista de Seguridad establece los procedimientos. 6. El Especialista de Seguridad elabora el Plan de Contingencia. 7. El Especialista de Seguridad recoge todas las actividades anteriores en un documento. 8. Entrega el Plan de Seguridad Informática.
9. Recibe el Plan de Seguridad Informática. 10. Revisa el PSI. 11. Aprueba el PSI.	
Mejoras	Contar con una herramienta que guíe los pasos para el proceso de elaboración del PSI permitirá que no se obvie ningún paso para la elaboración del mismo y que se agilice el proceso de realizar el Plan.

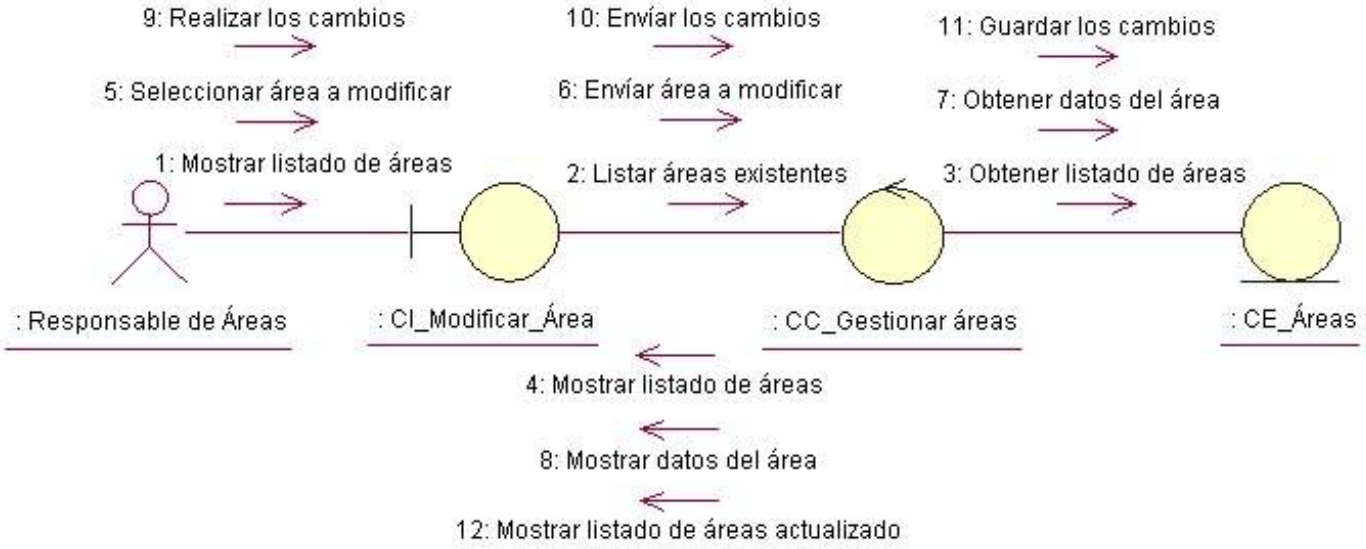
Anexo 4 Diagrama de Colaboración "Autenticar Usuario".



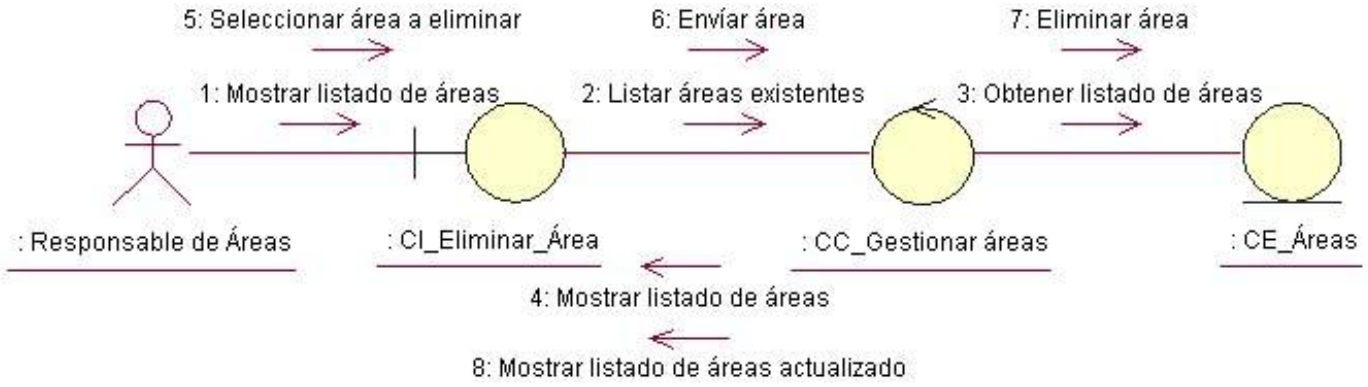
Anexo 5 Diagrama de Colaboración "Gestionar Áreas". Escenario: Adicionar Área.



Anexo 6 Diagrama de Colaboración "Gestionar Áreas". Escenario: Modificar Área.

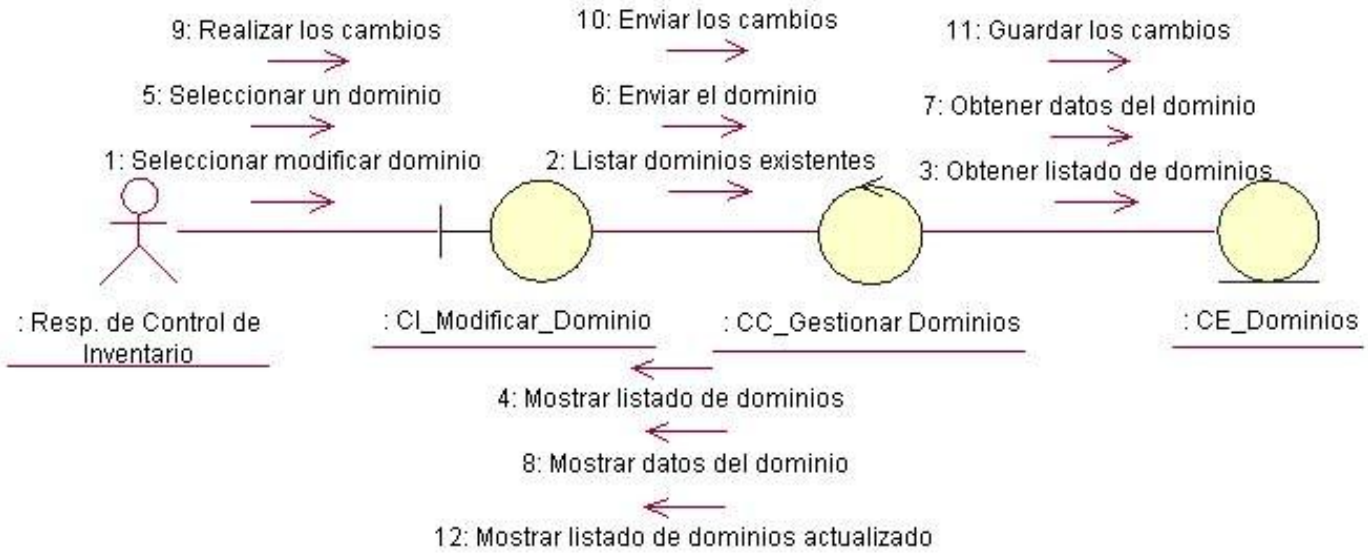


Anexo 7 Diagrama de Colaboración "Gestionar Áreas". Escenario: Eliminar Área.

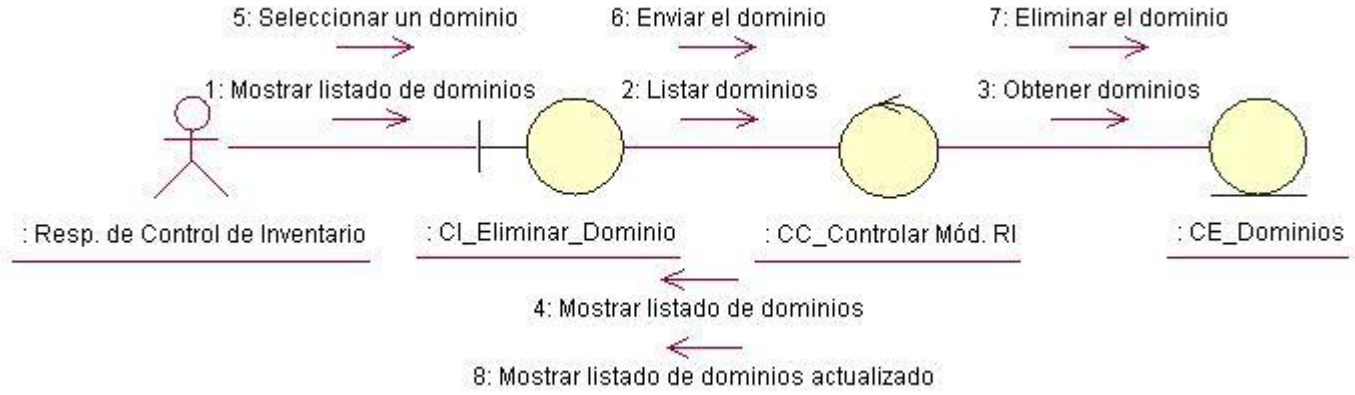


Anexo 8 Diagrama de Colaboración "Gestionar Dominio". Escenario: Adicionar Dominio.

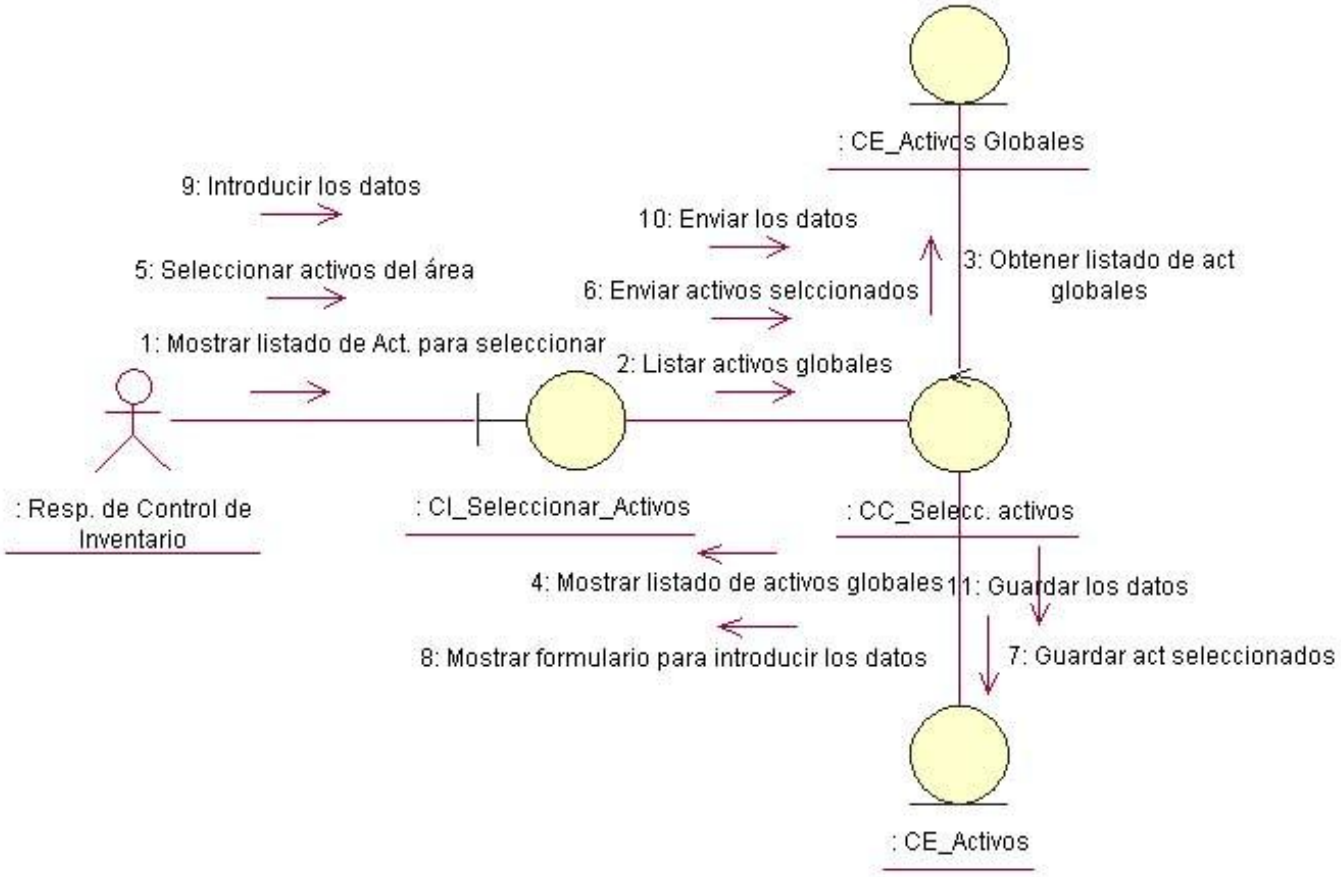
Anexo 9 Diagrama de Colaboración "Gestionar Dominio". Escenario: Modificar Dominio.



Anexo 10 Diagrama de Colaboración "Gestionar Dominio". Escenario: Eliminar Dominio.

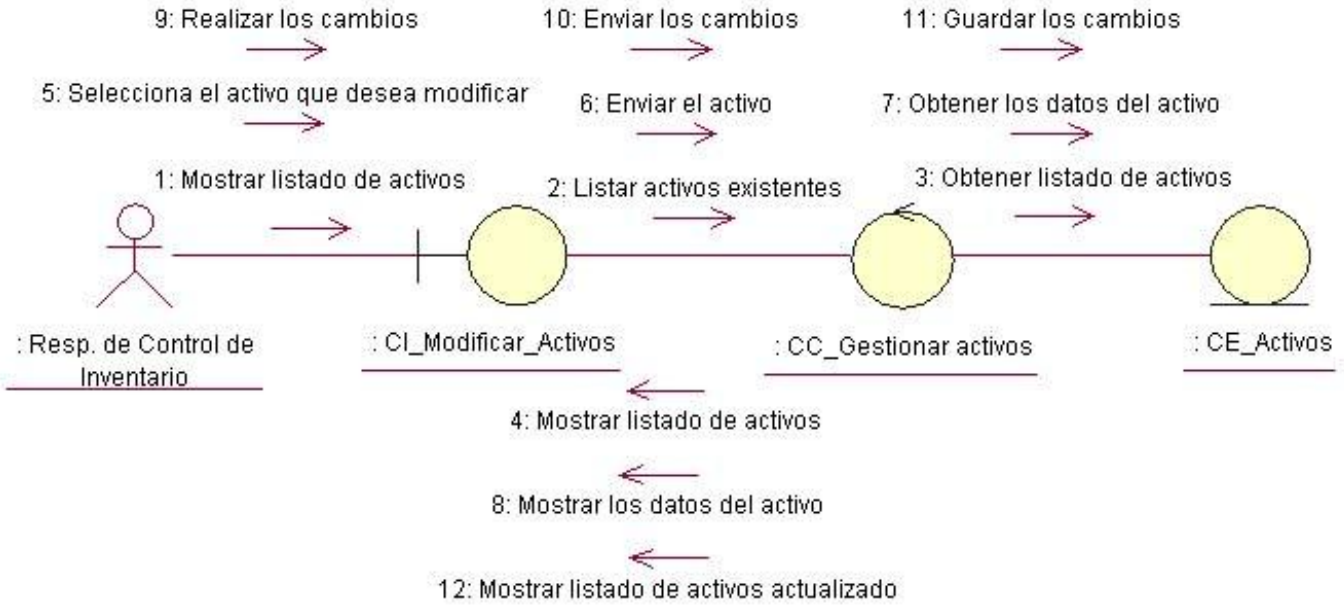


Anexo 11 Diagrama de Colaboración "Seleccionar Activos del Área".



Anexo 12 Diagrama de Colaboración "Gestionar Activos". Escenario: Adicionar Activos.

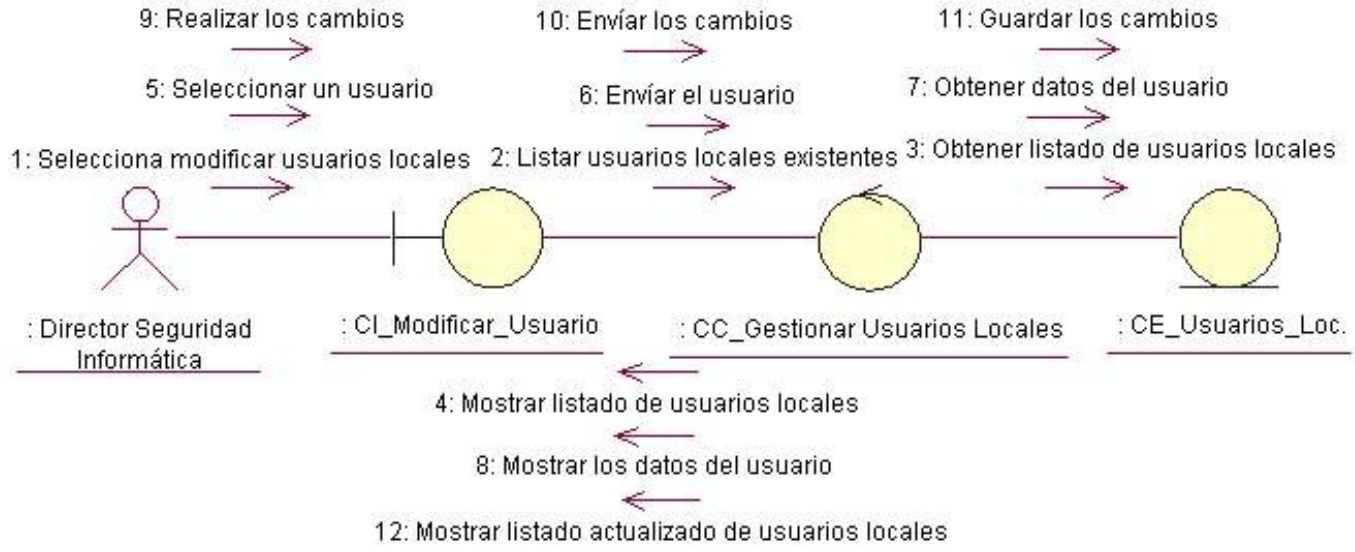
Anexo 13 Diagrama de Colaboración "Gestionar Activos". Escenario: Modificar Activos.



Anexo 14 Diagrama de Colaboración "Gestionar Activos". Escenario: Eliminar Activos.

Anexo 15 Diagrama de Colaboración "Gestionar Usuarios Loc.". Escenario: Adicionar Usuarios Locales.

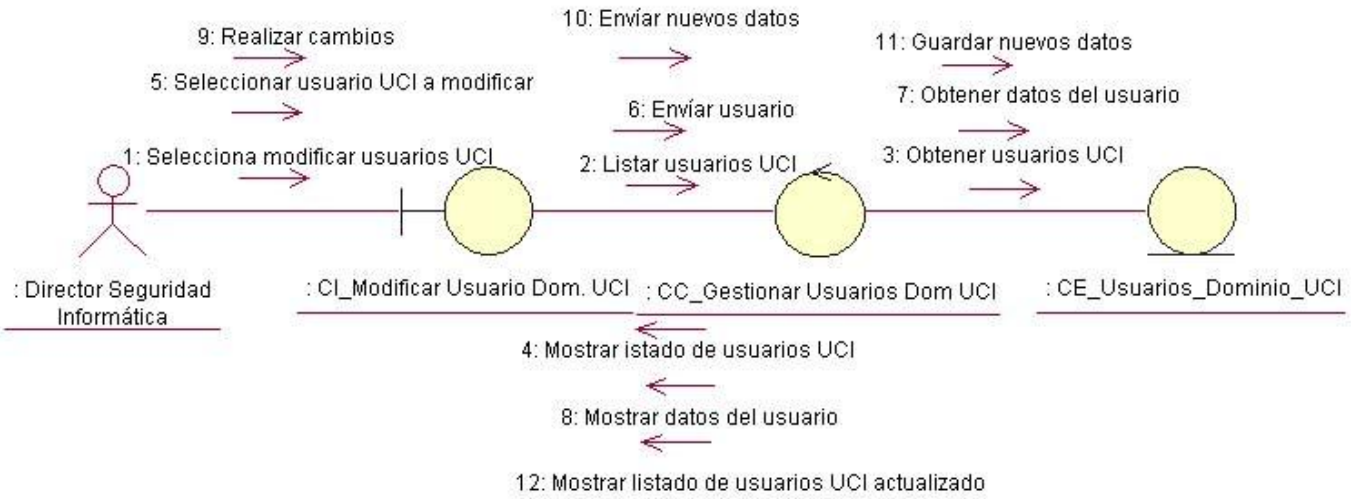
Anexo 16 Diagrama de Colaboración "Gestionar Usuarios Loc.". Escenario: Modificar Usuarios Locales.



Anexo 17 Diagrama de Colaboración "Gestionar Usuarios Loc.". Escenario: Eliminar Usuarios Locales.

Anexo 18 Diagrama de Colaboración "Gestionar Usuarios Dom. UCI". Escenario: Adicionar Usuarios Dom. UCI.

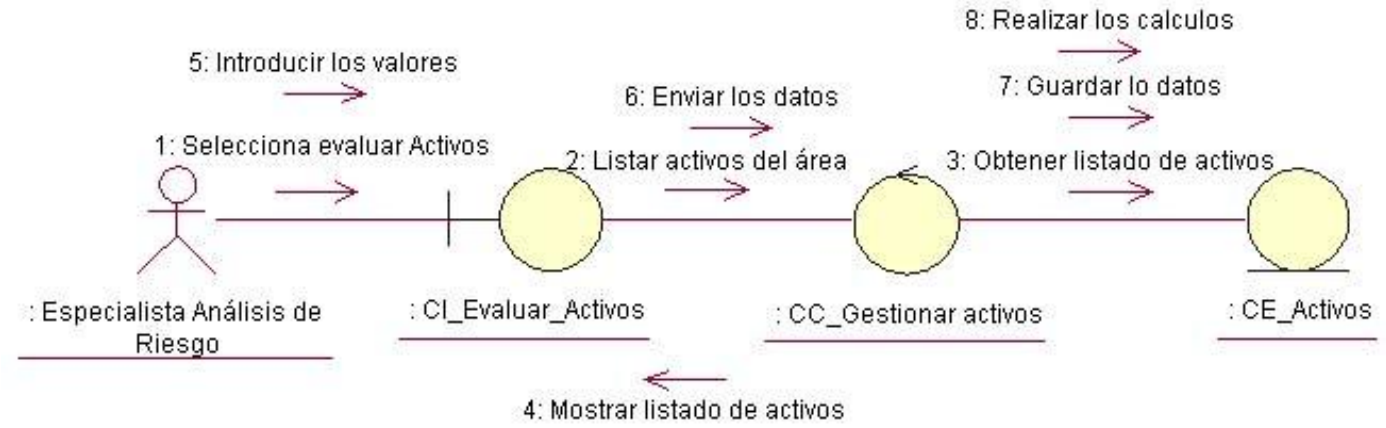
Anexo 19 Diagrama de Colaboración "Gestionar Usuarios Dom. UCI". Escenario: Modificar Usuarios Dom. UCI.



Anexo 20 Diagrama de Colaboración "Gestionar usuarios Dom. UCI". Escenario: Eliminar Usuarios Dom. UCI.

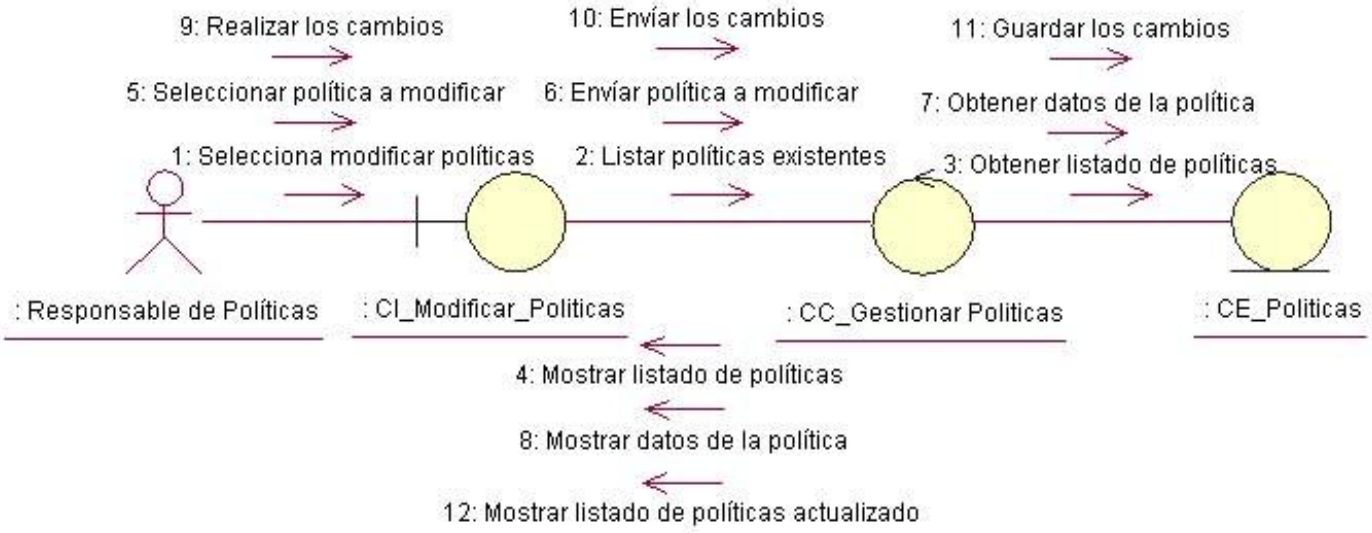
Anexo 21 Diagrama de Colaboración "Adicionar Amenaza". Escenario: Adicionar Amenaza.

Anexo 22 Diagrama de Colaboración "Calcular Importancia de Activos".



Anexo 23 Diagrama de Colaboración "Gestionar Políticas". Escenario: Adicionar Políticas.

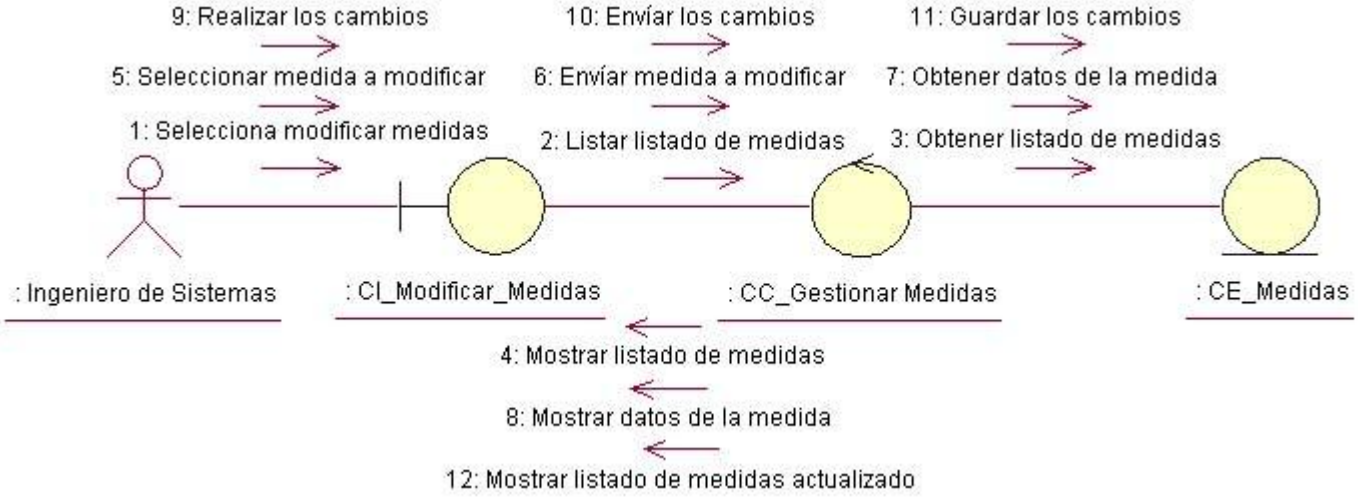
Anexo 24 Diagrama de Colaboración "Gestionar Políticas". Escenario: Modificar Políticas.



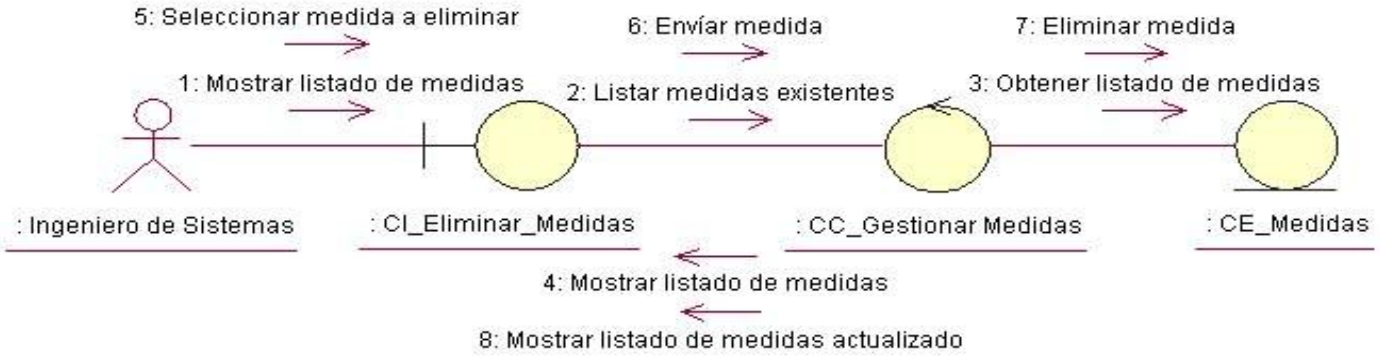
Anexo 25 Diagrama de Colaboración "Gestionar Políticas". Escenario: Eliminar Políticas.

Anexo 26 Diagrama de Colaboración "Gestionar Medidas". Escenario: Adicionar Medidas.

Anexo 27 Diagrama de Colaboración "Gestionar Medidas". Escenario: Modificar Medidas.

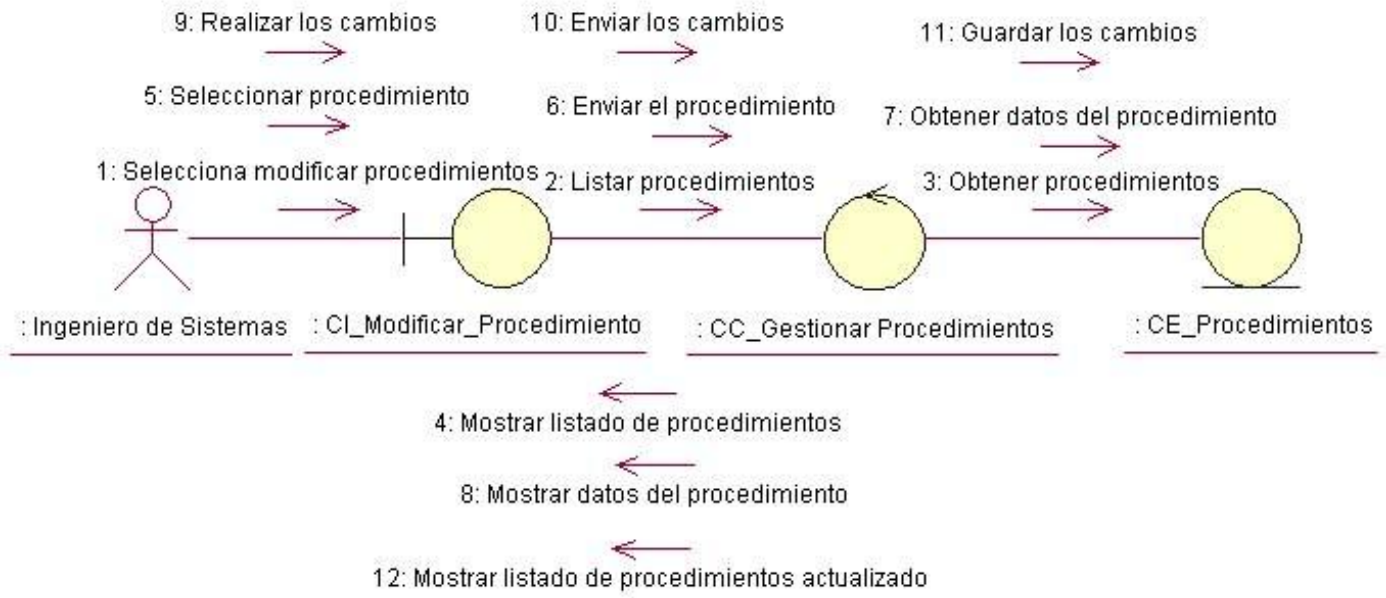


Anexo 28 Diagrama de Colaboración "Gestionar Medidas". Escenario: Eliminar Medidas.



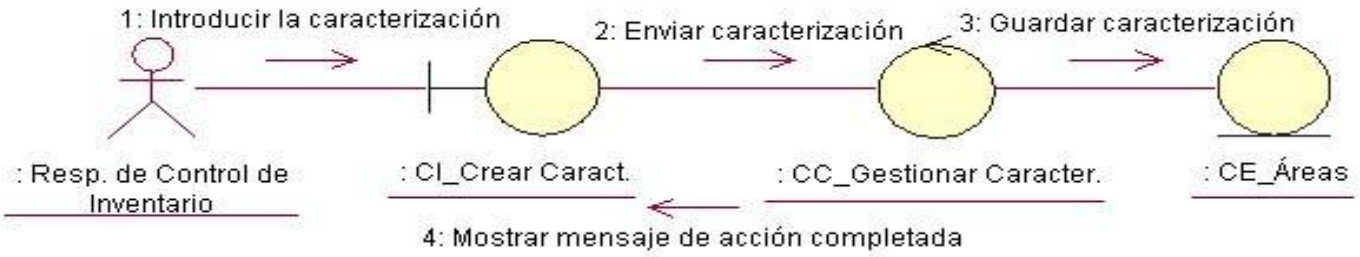
Anexo 29 Diagrama de Colaboración "Gestionar Procedimientos". Escenario: Adicionar Procedimientos.

Anexo 30 Diagrama de Colaboración "Gestionar Procedimientos". Escenario: Modificar Procedimientos.

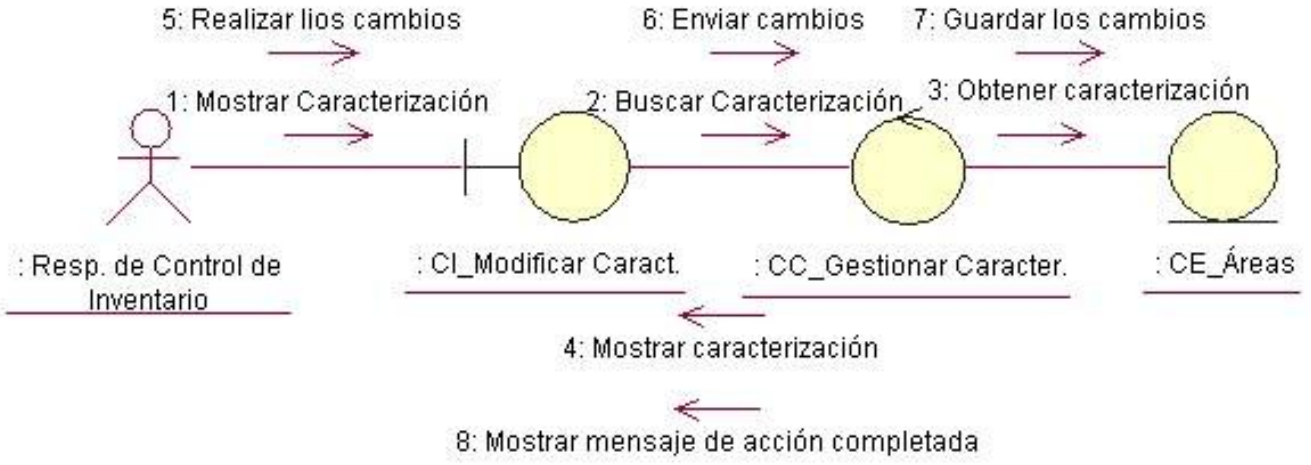


Anexo 31 Diagrama de Colaboración "Gestionar Procedimientos". Escenario: Eliminar Procedimientos.

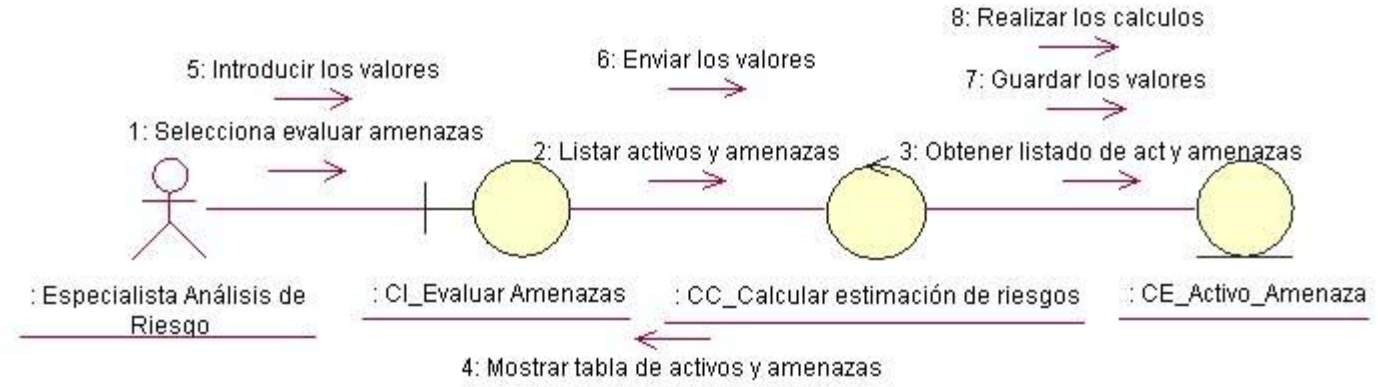
Anexo 32 Diagrama de Colaboración "Gestionar Caracterización". Escenario: Adicionar Caracterización.



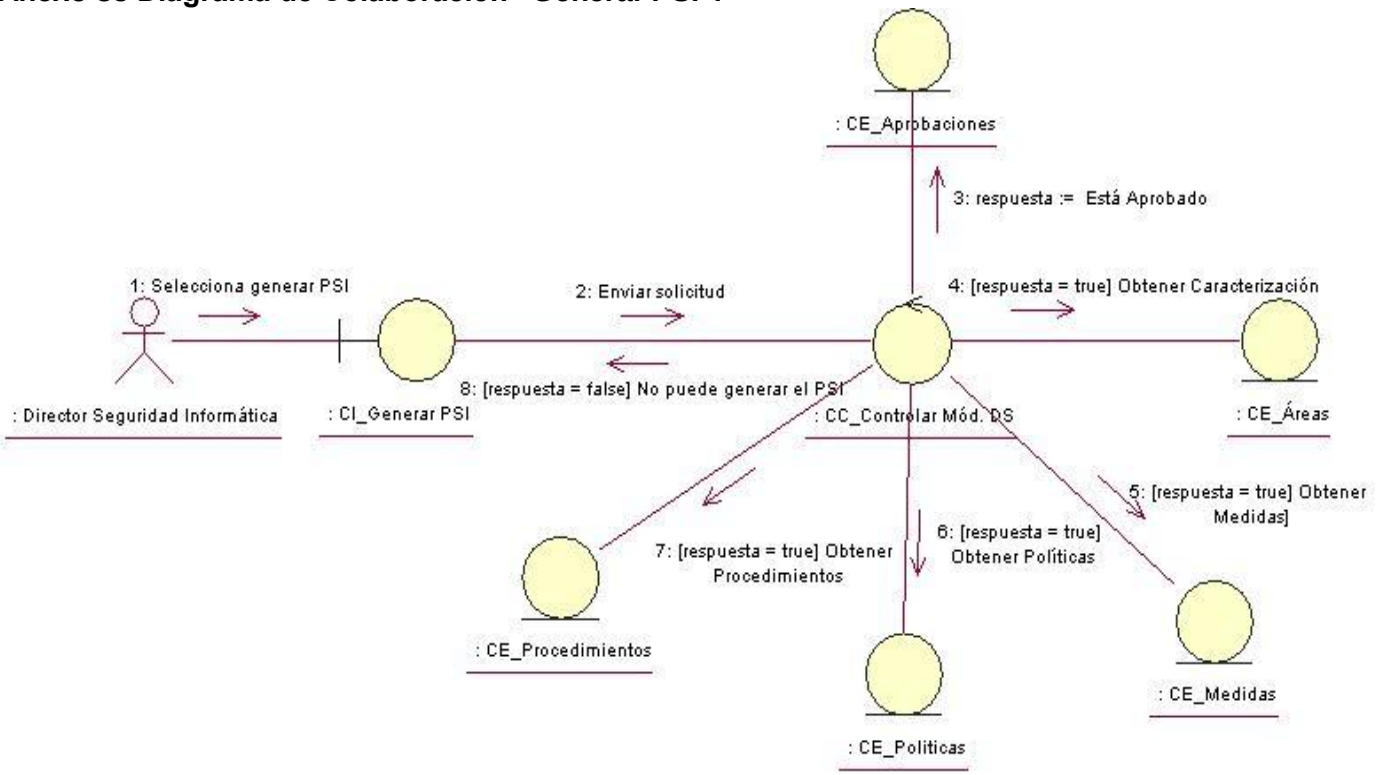
Anexo 33 Diagrama de Colaboración "Gestionar Caracterización". Escenario: Modificar Caracterización.



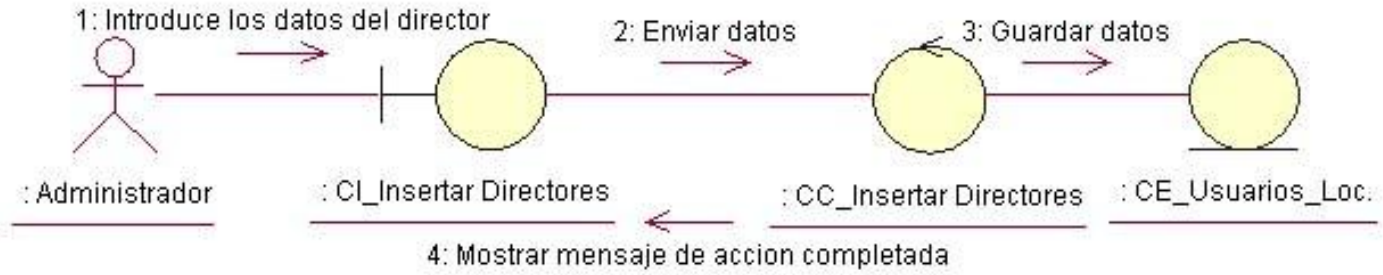
Anexo 34 Diagrama de Colaboración "Realizar Estimación de Riesgo".



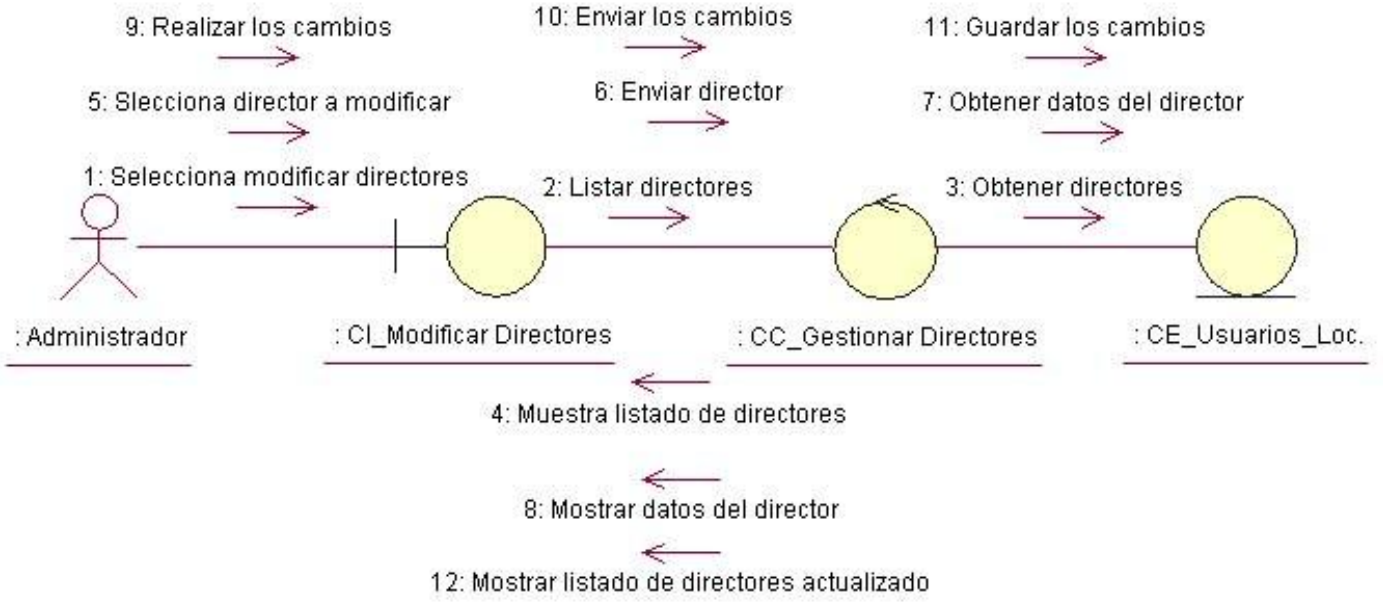
Anexo 35 Diagrama de Colaboración "Generar PSI".



Anexo 36 Diagrama de Colaboración "Gestionar Directores". Escenario: Insertar Directores.



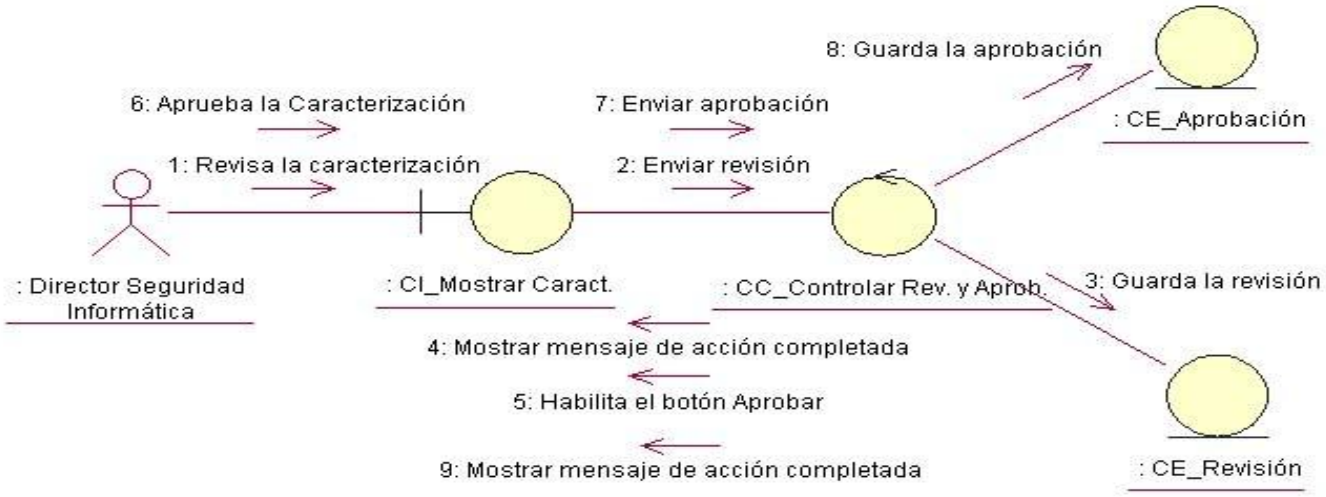
Anexo 37 Diagrama de Colaboración "Gestionar Directores". Escenario: Modificar Directores.



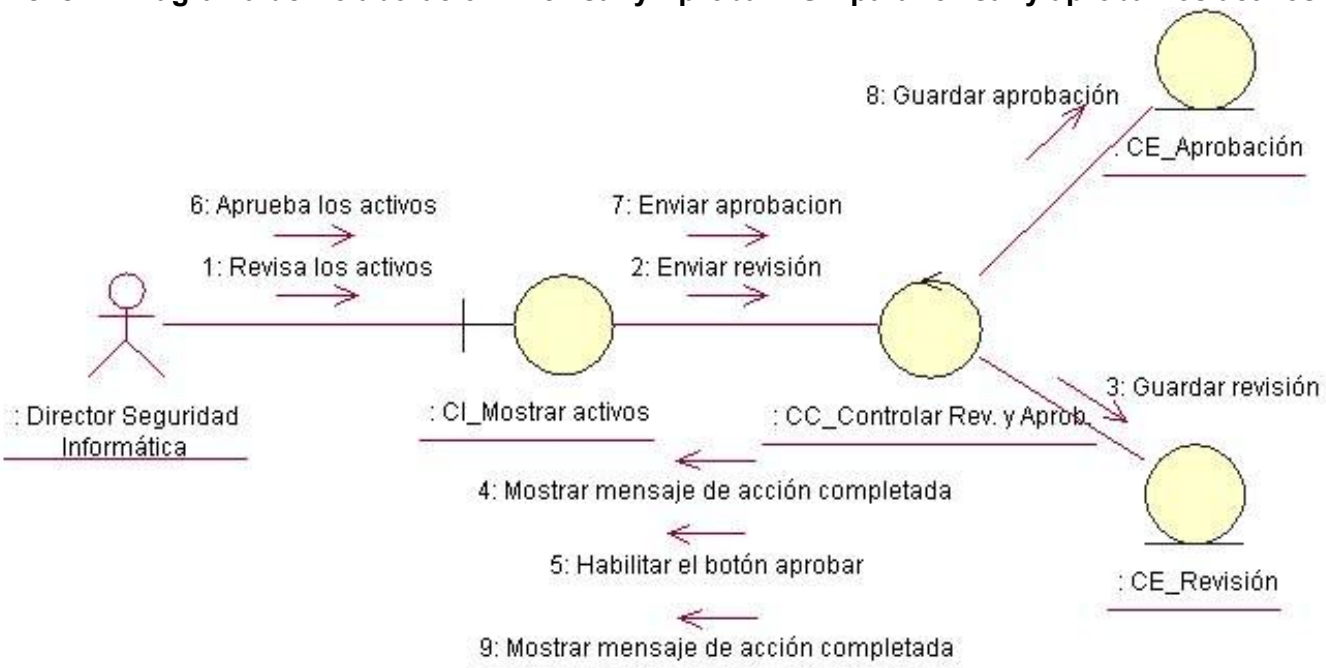
Anexo 38 Diagrama de Colaboración "Gestionar Directores". Escenario: Eliminar Directores.

Anexo 39 Diagrama de Colaboración "Cambiar Contraseña".

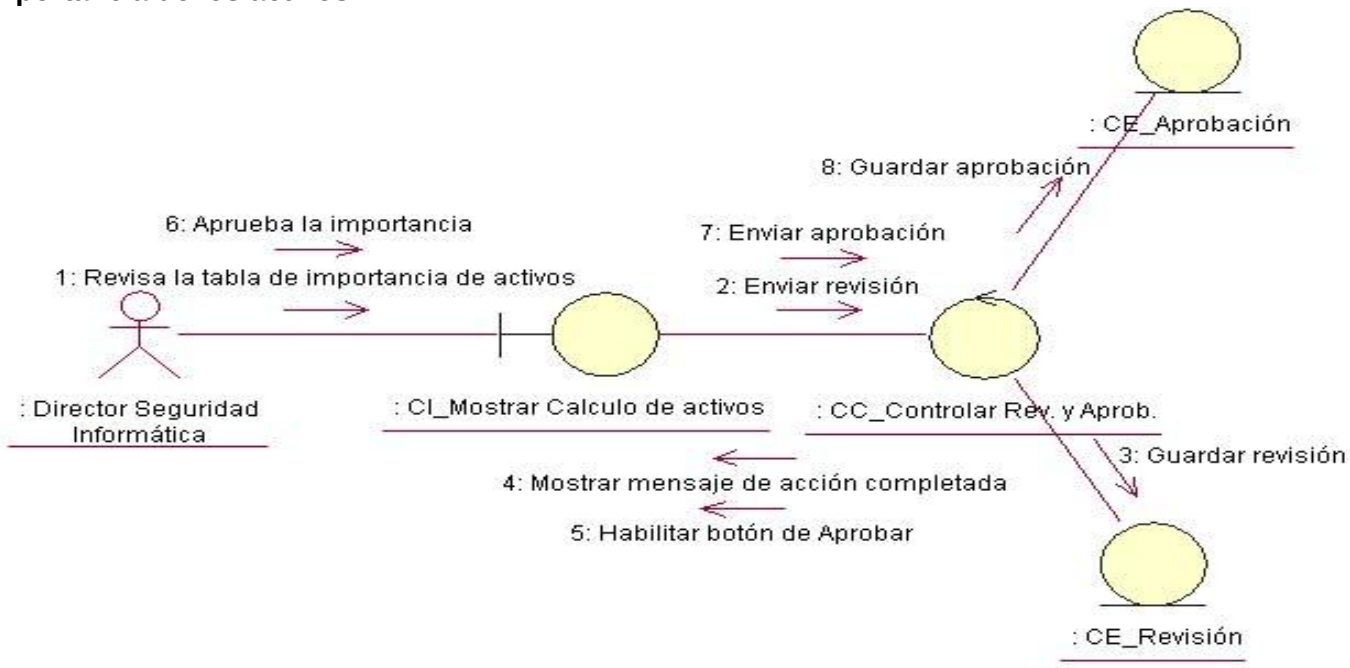
Anexo 40 Diagrama de Colaboración "Revisar y Aprobar PSI" para revisar y aprobar la caracterización.



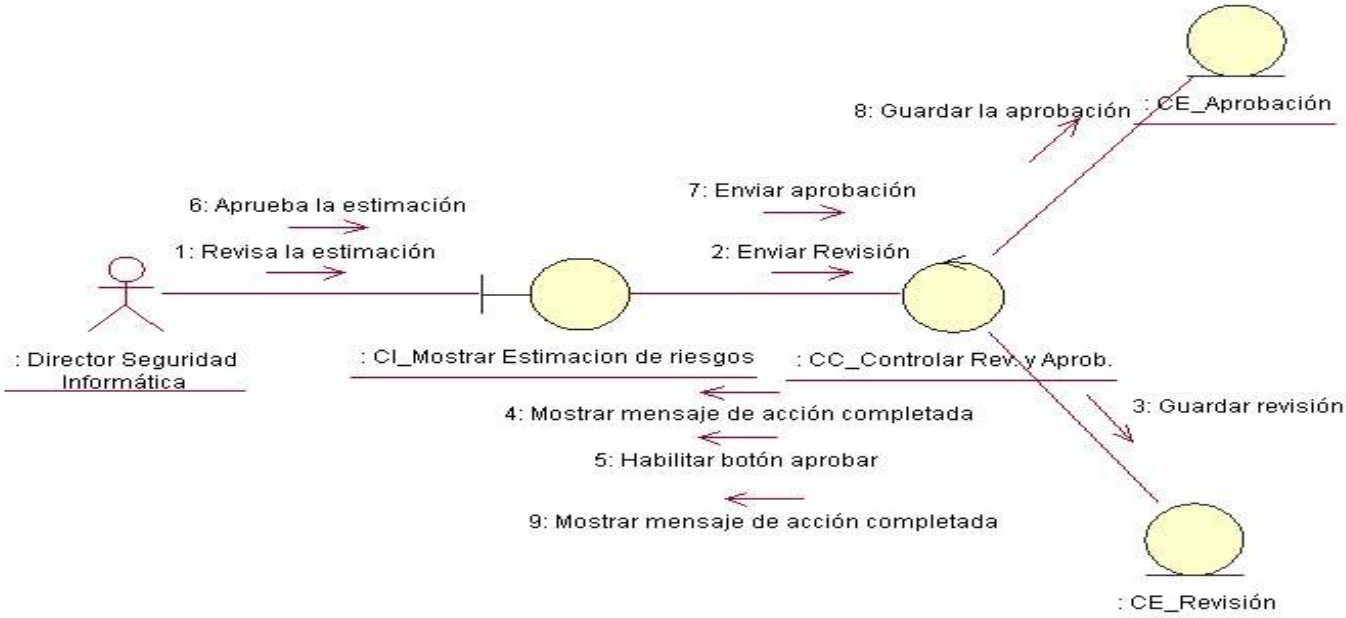
Anexo 41 Diagrama de Colaboración "Revisar y Aprobar PSI" para revisar y aprobar los activos.



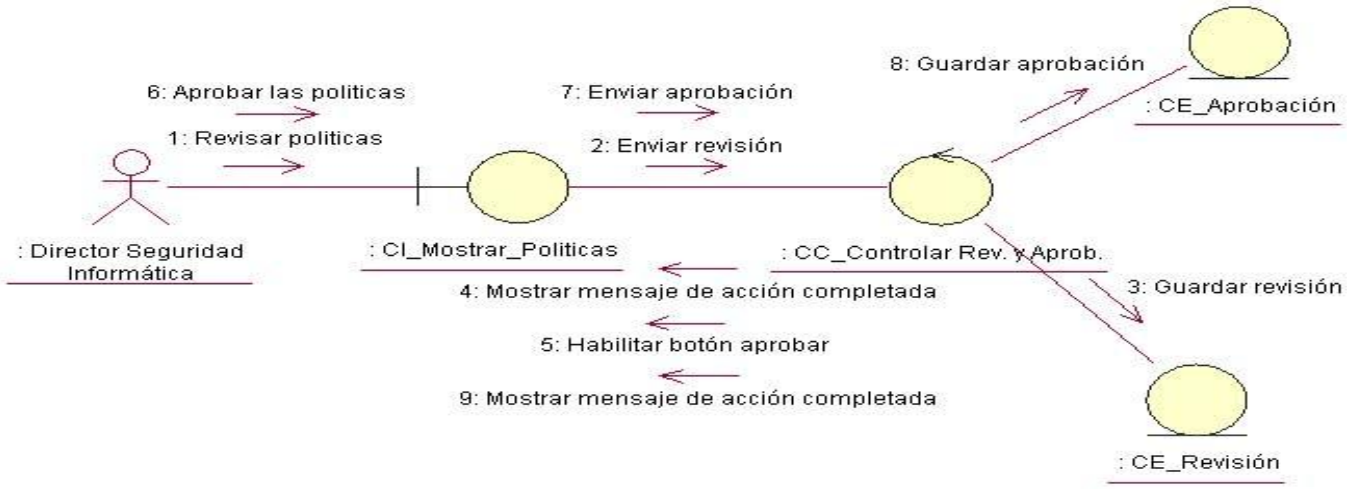
Anexo 42 Diagrama de Colaboración "Revisar y Aprobar PSI" para revisar y aprobar el cálculo de la importancia de los activos.



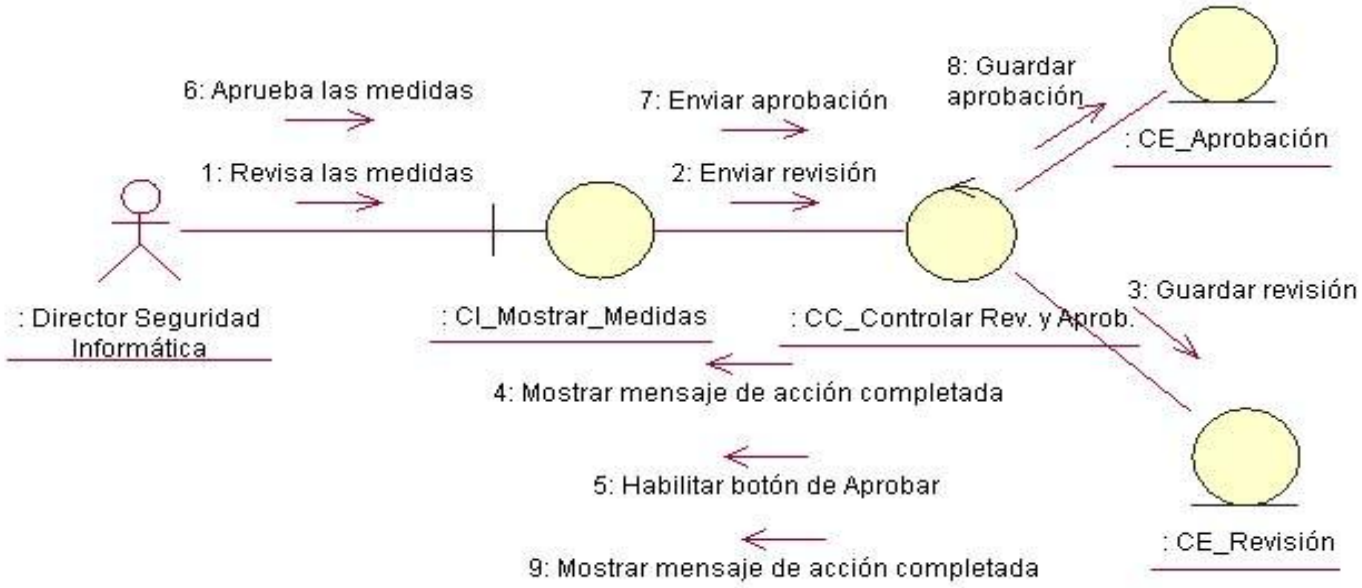
Anexo 43 Diagrama de Colaboración "Revisar y Aprobar PSI" para revisar y probar la estimación de riesgos.



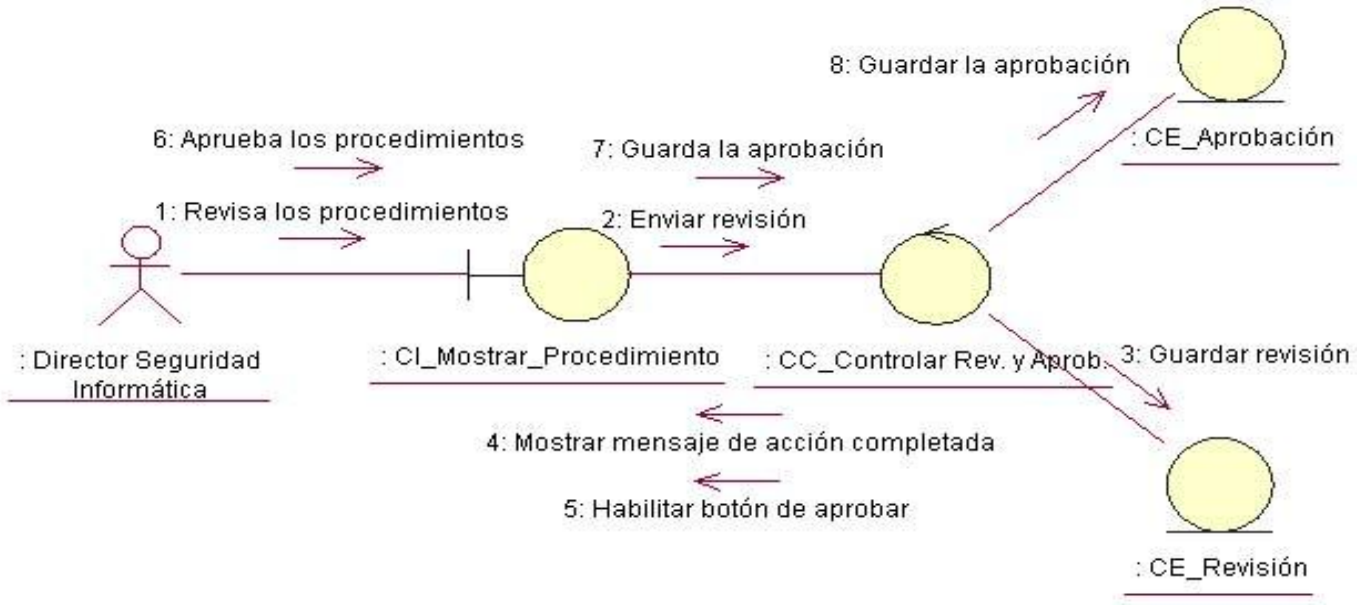
Anexo 44 Diagrama de Colaboración "Revisar y Aprobar PSI" para revisar y aprobar las políticas.



Anexo 45 Diagrama de Colaboración "Revisar y Aprobar PSI" para revisar y aprobar las medidas.



Anexo 46 Diagrama de Colaboración "Revisar y Aprobar PSI" para revisar y aprobar los procedimientos.



GLOSARIO DE TÉRMINOS

Amenaza: Situación o acontecimiento que pueda causar daños a los bienes informáticos. Puede ser una persona, un programa maligno o un suceso natural o de otra índole. Representan los posibles atacantes o factores que inciden negativamente sobre las debilidades del sistema.

Análisis de riesgos: Proceso dirigido a determinar la probabilidad de que las amenazas se materialicen sobre los bienes informáticos. Implica la identificación de los bienes a proteger, la determinación de las amenazas que actúan sobre ellos, así como la estimación de su probabilidad de ocurrencia y el impacto que puedan causar.

Bienes Informáticos: Elementos componentes del sistema informático que deben ser protegidos en evitación de que como resultado de la materialización de una amenaza sufran algún tipo de daño.

Riesgo: Probabilidad de que una amenaza se materialice sobre una vulnerabilidad del sistema informático, causando un impacto negativo en la organización.

Sistema Informático*:* Conjunto de bienes informáticos de que dispone una entidad para su correcto funcionamiento y la consecución de los objetivos propuestos

Sistema de Seguridad Informática: Conjunto de medios humanos, técnicos y administrativos, que de manera interrelacionada garantizan diferentes grados de seguridad informática en correspondencia con la importancia de los bienes a proteger y los riesgos estimados.

Vulnerabilidad: Punto o aspecto del sistema que es susceptible de ser atacado o de dañar la seguridad del mismo. Representan las debilidades o aspectos falibles o atacables en el sistema informático. Califica el nivel de riesgo de un sistema.