



Facultad 9

---

**SISTEMA AUTOMATIZADO PARA LA GESTIÓN DE RIESGOS  
EN LOS PROYECTOS INFORMÁTICOS.**

---

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

**Autor(es):** Lázaro Yasmany Benito Ledesma.  
Osmani González Cabrera.

**Tutor:** Ing. Daniel Burgo Hechavarría

**Co-tutor:** Dr. Mario González Arencibia.

*Ciudad de la Habana, Julio 5, 2008*

*Año 50 de la Revolución.*

## DEDICATORIA

A mi **Mamá** por ser tan especial conmigo y darme esa confianza que se necesita para lograr algo en la vida, por quererme y entenderme, por ayudarme, por malcriarme y por yo quererla tanto como la quiero.

A mi **Papá** que me quiere y que ayuda en lo que haga falta.

A mi **hermano** que me fue el que inició mi interés por los estudios y que tantos buenos ejemplos me dio.

A mi novia, **Yadira**, que ha sido desde que la conocí mi razón de existir, mi guía, y mi amor.

A mi **abuela** que siempre me enseña tanto.

A mi **abuelo**, porque se que parte de él se quedo aquí conmigo.

**Osmani González Cabrera**

A **mi mamá**, mi orgullo, a quien le debo todo lo que soy.

A **mi abuelo Jesús Benito Blanco**, por ocupar un lugar especial en mi corazón, preocupándose por mis estudios y guiándome siempre por el buen camino.

A **Jimagua**, por confiar y quererme como a un hijo.

A **Patry**, mi novia, por darme siempre la confianza, tranquilidad y seguridad de que todo saldrá bien, por estar cuando más la he necesitado.

A **Caro** y **Jose**, mis suegros, por apoyarme y creer en mi en todo momento.

A **mi papá** y **mis abuelas** que aunque no estén físicamente les dedico este trabajo porque se que estarían orgullosos de mi.

A toda **mi familia**, que de una forma u otra se ha preocupado y me ha apoyado a lo largo de la carrera.

A mis **amigos**.

**Lázaro Yasmany Benito Ledesma**

**AGRADECIMIENTOS.**

A **Yelenys Zulueta Veliz** por ayudarnos a dar los primeros pasos en la realización de este trabajo.

A **Burgos**, nuestro tutor, por su preocupación.

A **Yadira** por brindarnos su apoyo, constancia y experiencia en los momentos que más lo necesitábamos.

A **Liester** y **Eddy** por su ayuda con la programación.

A **Ariadna** y **Carlos** por facilitarnos parte de sus investigaciones para complementar la nuestra.

A **Lidiris** y **Yudelis** por darnos una mano con los detalles de la tesis.

A todos **los profesores** que durante estos 5 años nos han enseñado, formado y guiado a ser lo que somos hoy.

A nuestras **familias** que a pesar de la distancia siempre nos han brindado su confianza.

A **Fidel** y la **Revolución Cubana** por ser los que han hecho posible que hoy haya muchos que han realizado los sueños de ser útiles a la Patria y a la sociedad con sus conocimientos y sus ideas.

## DECLARACIÓN DE AUTORÍA

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

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

-----

Firma del Autor

Osmani González Cabrera

-----

Firma del Autor

Lázaro Benito Ledesma

-----

Firma del Tutor

Ing. Daniel Burgo Hechavarría.

## OPINIÓN DEL TUTOR

Título: Sistema automatizado para la gestión de riesgos en los proyectos informáticos.

Autores: Osmani González Cabrera

Lázaro Yasmany Benito Ledesma

El tutor del presente Trabajo de Diploma considera que durante su ejecución los autores mostraron las cualidades que a continuación se detallan.

Las diplomantes Osmani González Cabrera y Lázaro Yasmany Benito Ledesma, han demostrado responsabilidad y esmero en la realización de su trabajo de diploma. Como factores fundamentales en el cumplimiento exitoso de los objetivos trazados en la investigación, podemos mencionar su dinamismo y laboriosidad. Los estudiantes mostraron independencia, originalidad y autonomía en cada fase de la tarea, logrando resultados satisfactorios. Han trabajado de manera organizada y planificada, dando muestras de poseer una buena preparación metodológica y un gran sentido del compromiso laboral.

El documento realizado goza de una buena organización, está basado en métodos investigativos científicamente aprobados. Es capaz de llevar al lector la información referida al contenido y los objetivos perseguidos con su trabajo. El diseño realizado cumple con los objetivos trazados, y es de gran importancia.

Los estudiantes son muy exigentes en su labor.

Por todo lo anteriormente expresado, considero que los estudiantes están aptos para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de \_\_\_ puntos.

---

Firma  
Ing. Daniel Burgos Hechavarría

---

\_\_\_ de Julio de 2008

## Resumen

Actualmente, la gestión de riesgos es una de las disciplinas más importantes de los sistemas de gestión. Provee a las organizaciones una mejoría en sus capacidades para alinear el nivel de riesgos con la estrategia de la organización, además vincula el crecimiento, riesgo y retorno de la inversión. La gestión de riesgos también mejora la toma de decisiones, minimiza las pérdidas y los impactos operativos identificando y gestionando los riesgos en toda la organización.

En la facultad 9 se están llevando a cabo varios proyectos, sin embargo la mayoría de éstos están expuestos a distintos riesgos, pues durante su etapa de desarrollo pueden darse consecuencias no previstas inicialmente que pueden afectar su culminación y ocasionar daños a objetos del mismo ya sea por condiciones de trabajo, obsolescencia, pérdida de materiales o equipos, trayendo todos estos factores aparejado un déficit en la obtención de mayores ganancias, mejor eficiencia en la producción y en los servicios que brinda nuestra facultad.

Es por eso que este trabajo de diploma se propone desarrollar una aplicación capaz de contener la relación de los posibles riesgos a los que está expuesto un proyecto, y que nos muestre cuánto representa la ocurrencia de cada uno de estos para el buen desarrollo de las aplicaciones.

Para el desarrollo de esta herramienta se utiliza RUP (Proceso unificado de desarrollo de software) como metodología, UML (Lenguaje unificado de modelado) como lenguaje representativo y Rational Rose como herramienta Case, además de PHP (Hypertext Pre-processor) como lenguaje de programación para la implementación y MySQL como gestor de base de datos. En este documento se describen los principales conceptos y funcionalidades de la aplicación. Además se detallan los métodos científicos y técnicas utilizadas para el desarrollo de la investigación.

## Palabras Claves

Riesgo, Gestión de Riesgos, Tratamiento de Riesgos, Identificación de Riesgos, Análisis de Riesgos, Proyecto de desarrollo de software, Sistema automatizado para la gestión de riesgos.

## Tablas y Figuras:

Tabla 1: Subcategorías genéricas de riesgos conocidos y predecibles. ....	10
Tabla 2: Actores del sistema. ....	42
Tabla 3. Casos de uso del Sistema. ....	43
Tabla 4. Descripción del CU Autenticarse.....	48
Tabla 5. Descripción del CU Gestionar Proyectos. ....	50
Tabla 6. Descripción del CU Gestionar Equipos. ....	52
Tabla 7. Descripción del CU Gestionar Riesgos.....	53
Tabla 8. Descripción del CU Realizar Análisis. ....	55
Tabla 9. Descripción del CU Mostrar riesgos priorizados.....	56
Tabla 10. Descripción del CU Gestionar Estrategia. ....	58
Tabla 11. Descripción del CU Gestionar Respuesta. ....	60
Tabla 12. Descripción del CU Calcular métricas.....	61
Tabla 13. Documentar Experiencias. ....	62
Tabla 14. Descripción del CU Mostrar Experiencias. ....	63
Tabla 15. Descripción del CU Mostrar Registro de Riesgos. ....	64
Figura 1: Concepto de Gestión de Riesgo.....	12
Figura 2: Procedimiento de Gestión de Riesgo.....	13
Figura 3: Guías Metodológicas de MAGERIT (v1.0). ....	17
Figura 4: Submodelo de Procesos.....	19
Figura 5: Herramienta PILAR.....	20
Figura 6: GxSGSI Análisis de Impactos. ....	21
Figura 7: Arquitectura de tres capas. ....	27
Figura 8: RUP en dos dimensiones. ....	32
Figura 9: Diagrama de objetos del dominio.....	39
Figura 10. Diagrama de CU de Documentar Experiencias.....	44
Figura 11. Diagrama de CU de Gestionar Riesgos. ....	45
Figura 12. Diagrama de CU de Seguridad .....	46
Figura 13. Diagrama de Clases del Análisis: CU Documentar Experiencias. ....	65
Figura 14. Diagrama de Clases del Análisis: CU Mostrar Experiencias. ....	66
Figura 15. Diagrama de Clases del Análisis: CU Mostrar Registros de Riesgos. ....	66
Figura 16. Diagrama de Clases del Análisis: CU Calcular Métricas. ....	67
Figura 17. Diagrama de Clases del Análisis: CU Gestionar Equipo. ....	68
Figura 18. Diagrama de Clases del Análisis: CU Gestionar Estrategias. ....	68
Figura 19. Diagrama de Clases del Análisis: CU Gestionar Respuestas. ....	69
Figura 20. Diagrama de Clases del Análisis: CU Gestionar Riesgos .....	70
Figura 21. Diagrama de Clases del Análisis: CU Mostrar Riesgos Priorizados.....	70
Figura 22. Diagrama de Clases del Análisis: CU Realizar análisis. ....	71
Figura 23. Diagrama de Clases del Análisis: CU Autenticarse. ....	71
Figura 24. Diagrama de Clases del Análisis: CU Gestionar proyectos. ....	72

---

Figura 25. Diagrama de Clases del Diseño: CU Documentar Experiencias.....	73
Figura 26. Diagrama de Clases del Diseño: CU Mostrar Experiencias.....	74
Figura 27. Diagrama de Clases del Diseño: CU Mostrar Registro de Riesgos.....	75
Figura 28. Diagrama de Clases del Diseño: CU Calcular Métricas.....	76
Figura 29. Diagrama de Clases del Diseño: CU Gestionar Equipo.....	77
Figura 30. Diagrama de Clases del Diseño: CU Gestionar Estrategias.....	78
Figura 31. Diagrama de Clases del Diseño: CU Gestionar Respuestas.....	79
Figura 32. Diagrama de Clases del Diseño: CU Gestionar Riesgos.....	80
Figura 33. Diagrama de Clases del Diseño: CU Mostrar Riesgos Priorizados.....	81
Figura 34. Diagrama de Clases del Diseño: CU Realizar Análisis.....	82
Figura 35. Diagrama de Clases del Diseño: CU Autenticarse.....	83
Figura 36. Diagrama de Clases del Diseño: CU Gestionar Proyectos.....	84
Figura 37. Diagrama de Clases Persistentes.....	87
Figura 38. Modelo Físico de Datos.....	88
Figura 39. Modelo de Datos.....	89
Figura 40. Diagrama de Despliegue.....	90
Figura 41. Diagrama de Componentes de Base de Datos.....	91
Figura 42. Diagrama de componentes Código Fuente.....	92
Figura 43: Diagrama de Componentes Web o código ejecutable (I).....	93
Figura 44. Diagrama de Componentes Web o código ejecutable (II).....	94
Figura 45. Diagrama de Componentes Web o código ejecutable (III).....	95



**Índice:**

<b>Introducción.....</b>	<b>1</b>
<b>Capítulo 1. Fundamentación Teórica.....</b>	<b>7</b>
1.1 Introducción.....	7
1.2 Conceptos Asociados con la Gestión de Riesgos.....	7
1.2.1 Riesgo.....	7
1.2.1.1 Categorías de Riesgo.....	8
1.2.1.2 Identificación del Riesgo.....	9
1.2.2 Gestión.....	11
1.2.3 Gestión de Riesgo.....	11
1.3 Gestión de Riesgos en Proyectos Informáticos.....	12
1.3.1 Gestión de Riesgos en proyectos informáticos en la Facultad 9.....	13
1.4 Modelos de Gestión de Riesgos.....	14
1.5 Modelos y herramientas de Gestión de Riesgos existentes.....	15
1.5.1 MAGERIT.....	16
1.5.2 EUROMÉTODO.....	17
1.5.3 MoGeRi.....	18
1.5.4 Herramienta PILAR.....	20
1.5.5 Herramienta GxSGSI.....	21
1.6 Conclusiones.....	22
<b>Capítulo 2. Tendencias y Tecnologías Actuales a Desarrollar.....</b>	<b>23</b>
2.1 Introducción.....	23
2.2 Tecnologías de la Información y las Comunicaciones (TIC).....	23
2.4. Modelo Cliente - Servidor.....	24
2.5 Servidor Web Apache.....	25
2.6. Patrón Arquitectónico: Arquitectura en Capas.....	26
2.6.1 Arquitectura de tres niveles.....	26
2.7. Sistemas Web Modulares.....	27
2.8 Lenguaje de programación.....	28
2.8.1 Active Server Pages (ASP).....	28
2.8.2 ¿Por qué PHP como lenguaje de programación?.....	28
2.9 Gestor de bases de datos.....	29
2.9.1 Microsoft SQL Server.....	29
2.9.2 PostgreSQL.....	30
2.9.3 ¿Por qué MySQL como gestor de Bases de Datos?.....	30
2.10 Metodologías y herramientas a utilizar para el desarrollo del sistema.....	31
2.10.1 Metodología RUP.....	31
2.10.2 Lenguaje de modelado UML.....	33
2.10.3 Herramientas Case.....	33
2.10.3.1 Rational Rose.....	33
2.10.3.1 Embarcadero ERStudio 6.6.....	34
2.10.4 Macromedia Dreamweaver 8.0.....	35
2.10.5 Zend Studio.....	35
2.10.6 Conclusiones.....	36
<b>Capítulo 3. Características del Sistema.....</b>	<b>37</b>
3.1 Introducción.....	37
3.2 Modelo de Dominio.....	37
3.2.1 ¿Cuándo se aplica un modelo de Dominio?.....	37
3.2.2 Formas típicas de las clases del dominio.....	37
3.2.3 Conceptos principales del entorno de la aplicación.....	38
3.3 Especificación de los requisitos de software.....	39
3.3.1 Requerimientos Funcionales.....	39

3.3.2	Requerimientos No Funcionales.....	40
3.4	Modelo del Sistema.....	42
3.4.1	Actores del Sistema.....	42
3.4.2	Casos de uso.....	42
3.4.3	Diagrama de Casos de Uso del Sistema.....	44
3.4.3.1	Paquete Documentar Experiencias.....	44
3.4.3.2	Paquete Gestionar Riesgos.....	45
3.4.3.3	Paquete seguridad.....	46
3.5	Descripción de Casos de uso.....	47
	Conclusiones:.....	64
	<b>Capítulo 4. Análisis y Diseño del Sistema.....</b>	<b>65</b>
4.1	Introducción.....	65
4.2	Modelo del Análisis.....	65
4.2.1	Diagramas de clases de análisis.....	65
4.2.1.1	Documentar Experiencias.....	65
4.2.1.2	Gestión de Riesgos.....	67
4.2.1.3	Seguridad.....	71
4.3	Modelo del Diseño.....	73
4.3.1	Diagramas de clases y de interacción del diseño.....	73
4.3.1.1	Documentar Experiencias.....	73
4.3.1.2	Paquete Gestión de Riesgos.....	76
4.3.1.3	Seguridad.....	83
4.4	Definiciones de diseño.....	85
4.4.1	Tratamiento de errores.....	85
4.4.2	Seguridad.....	85
4.4.3	Interfaz.....	85
4.5	Diseño de la Base de Datos.....	87
4.5.1	Modelo Lógico de Datos.....	87
4.6	Diagrama de Despliegue.....	90
4.7	Modelo de Implementación.....	91
4.7.1	Diagramas de Componentes.....	91
4.7.1.1	Diagrama de componentes Base de Datos.....	91
4.7.1.2	Diagrama de componentes Código Fuente.....	92
4.7.1.3	Diagrama de componentes Código Ejecutable.....	93
4.8	Conclusiones.....	95
	<b>Capítulo 5. Análisis de Factibilidad.....</b>	<b>96</b>
5.1	Introducción.....	96
5.2	Planificación basada en casos de uso.....	96
5.3	Beneficios tangibles e intangibles.....	101
5.4	Análisis de costos y beneficios.....	101
5.5	Conclusiones.....	101
	<b>Conclusiones.....</b>	<b>102</b>
	<b>Recomendaciones.....</b>	<b>103</b>
	<b>Glosario de términos y siglas.....</b>	<b>104</b>
	<b>Referencias Bibliográficas.....</b>	<b>105</b>
	<b>Bibliografía.....</b>	<b>107</b>

## Introducción.

Actualmente, la gestión de riesgos es una de las disciplinas más importantes de los sistemas de gestión. La gestión del riesgo en las organizaciones proporciona un marco para administrar con eficacia y eficiencia, la incertidumbre, los riesgos asociados y oportunidades y consiguientemente mejorar su capacidad para construir valor.(1)

La gestión de riesgos (GR) provee a las organizaciones una mejoría en sus capacidades para alinear el nivel de riesgos con la estrategia de la organización, además vincula el crecimiento, riesgo y retorno de la inversión. La GR también mejora la toma de decisiones, minimiza las pérdidas y los impactos operativos, identifica y gestiona los riesgos en toda la organización. Igualmente la GR proporciona respuestas adecuadas a los múltiples peligros e incrementa las oportunidades de éxito y racionaliza las inversiones.

La GR también está implícita en el ámbito del software pues propone formalizar conocimiento orientado a la minimización o evitación de riesgos en proyectos de desarrollo de software, mediante la generación de principios y buenas prácticas de aplicación realista. Actualmente se han utilizado diferentes enfoques de gestión del riesgo, no obstante, son pocas las organizaciones que utilizan todavía de una forma explícita y sistemática métodos específicos para gestionar los riesgos en sus proyectos de software.

Robert Charette en su definición de riesgo plantea: En primer lugar el riesgo afecta a los futuros acontecimientos ya que el hoy y el ayer están más allá de lo que nos pueda preocupar, pues ya estamos cosechando lo que sembramos previamente con nuestras acciones del pasado. En segundo lugar, el riesgo implica cambios ya que puede venir dado por cambios de opinión, de acciones, de lugares. En tercer lugar, el riesgo implica elección, y la incertidumbre que entraña esta. Por tanto, el riesgo, como la muerte, es una de las pocas cosas inevitables de la vida. (2)

Cuando se considera el riesgo en el contexto de la ingeniería de software, los tres pilares de Charette se hacen continuamente evidentes. Es indiscutible que están presentes permanentemente las características de incertidumbre y de pérdida.

Por esa razón es que todas las empresas y entidades que se han encontrado frente a una situación de resultados inesperados por la ocurrencia de sucesos cuyas posibilidades de acontecer no tuvieron en cuenta, han coincidido en que es necesario realizar una gestión de riesgo.

La UCI es una universidad de nuevo tipo que lleva a la par la formación y preparación de sus estudiantes conjuntamente con la producción de software. En la facultad 9 son varios los proyectos productivos que se están llevando a cabo pero todos están expuestos a distintos riesgos; durante su etapa de desarrollo pueden darse consecuencias no previstas inicialmente que pueden afectar su buena culminación y ocasionar daños a objetos del mismo, (3) ya sea por condiciones de trabajo, obsolescencia, pérdida de materiales o equipos, trayendo todo esto aparejado un déficit en la obtención de mayores ganancias, mejor eficiencia en la producción y en los servicios que brinda nuestra facultad.

Se conocen informalmente los riesgos que podrían afectar el trabajo, pero estos no son registrados y mucho menos se procede a su análisis o gestión ya que actualmente en los proyectos productivos de la facultad hay una carencia de conocimientos sobre los marcos para la Gestión de Riesgos y por tanto de su utilización.

Además no existe una aplicación capaz de contener la relación de los posibles riesgos a los que está expuesto un proyecto, y que nos muestre cuánto representa la ocurrencia de cada uno de estos para el buen desarrollo de las aplicaciones.

La **situación problemática** anteriormente descrita desencadena el siguiente **problema científico**: ¿Cómo contribuir, mediante un sistema automatizado, a perfeccionar la Gestión de Riesgos de los proyectos informáticos en la Facultad 9?

Con el objetivo de resolver el problema planteado anteriormente, se define *la GR en los proyectos productivos* como el **Objeto de estudio** de la investigación.

De acuerdo a la situación antes mencionada se propone como **objetivo general**: Desarrollar un Sistema que sea capaz de gestionar los diferentes riesgos a los que se enfrenta un proyecto informático, así como la influencia de sus respectivas ocurrencias en el desarrollo del plan trazado. Y a partir de este se definen los siguientes.

**Objetivos Específicos:**

- Obtener información sobre la gestión de riesgo.
- Realizar un estudio del estado del arte.
- Realizar el análisis y diseño del Sistema Gestor de Riesgos.
- Modelar la Base de Datos.
- Realizar la implementación del Sistema Gestor.

De la relación entre el problema, el objeto de estudio y los objetivos de la investigación, se precisa *la automatización de la gestión de riesgos en los proyectos productivos de la facultad 9* como **Campo de acción**.

Por lo tanto se elaboró la siguiente **Hipótesis**, el desarrollo del software conllevará a la automatización de los servicios de gestión de riesgos para los proyectos de desarrollo del software en la facultad 9.

Para la realización de la investigación se plantearon las siguientes **Tareas**:

- Estudiar los diferentes modelos de gestión de riesgos existentes.
- Estudiar los diferentes tipos de riesgo, y sus clasificaciones.
- Realizar Investigación de la situación actual de la gestión de riesgo en los proyectos informáticos en nuestro campo de acción.
- Investigar en Internet sobre otros sitios que realicen gestiones similares a la que se va a desarrollar.
- Realizar entrevistas a los diferentes líderes de proyectos en la Facultad 9.
- Estudiar diferentes metodologías de desarrollo del software, enfatizando en las metodologías de aplicaciones de Internet (web-app).
- Analizar las ventajas y desventajas de los principales gestores de base de datos y lenguajes de programación web.
- Realizar el análisis y diseño del sistema.
- Analizar el modelo de datos a utilizar.
- Implementar la Base de Datos.

- Realizar implementación del sistema.

Y con la realización de todas las tareas antes expuestas se espera obtener los siguientes **Resultados**:

Desarrollar un sistema que brinde las siguientes posibilidades:

- Gestionar de forma segura los posibles riesgos que deben enfrentar los proyectos informáticos de la Facultad 9 de la Universidad de las Ciencias Informáticas.
- Aumento de la eficiencia en la gestión de riesgos.
- Mejor planificación del tiempo de realización y entrega de las etapas del proyecto.
- Respuesta eficiente para la toma de decisiones en la planificación del plan de actividades.

El diseño teórico ayuda a concretar el trabajo, pues en el mismo quedan definidos todos los elementos necesarios para buscar la respuesta al problema de investigación pero también es de vital importancia el diseño metodológico para la ejecución de una investigación, pues define el tiempo necesario para su realización, el costo de la misma y la calidad de los resultados obtenidos, permitiendo así definir la estrategia de investigación mas adecuada para cumplir los objetivos propuestos.

Los métodos de investigación científica que se utilizan en esta investigación son:

#### Métodos Teóricos

- **Analítico-Sintético:** Se analizan documentos, sitios web, artículos electrónicos, libros, etc., buscando las características de los diferentes modelos y herramientas existentes para la GR, permitiendo la extracción de los elementos más importantes que se relacionan con la GR en los proyectos productivos.
- **Análisis Histórico Lógico:** Se realiza el análisis histórico lógico de los procesos para la gestión de riesgos que actualmente se llevan cabo en los diferentes proyectos productivos en la Facultad 9.
- **Modelación:** Se utiliza este método para la modelación de los diagramas necesarios para llevar a cabo la construcción del sistema funcional.

## Métodos Empíricos

Entrevista: Se realizan entrevistas a varios Jefes de Proyecto de la Facultad 9 para conocer las condiciones actuales en las cuales se desarrolla el proceso de GR.

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

### **Capítulo I. Fundamentación Teórica de Gestión de Riesgos.**

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

### **Capítulo II. Tendencias y tecnologías actuales a desarrollar.**

En este capítulo se tratan las tecnologías utilizadas en el desarrollo de aplicaciones Web haciendo énfasis en las utilizadas para el desarrollo del sistema a implementar. También se explica la metodología y el lenguaje de modelado utilizado.

### **Capítulo III. Presentación de la Solución Propuesta.**

Se realiza el estudio del problema, se plantean las reglas del negocio a considerar en la aplicación, así como su modelo de dominio donde se describen las clases que están involucradas. También en este capítulo se describe la solución propuesta, exponiendo elementos imprescindibles para una solución exitosa: como lo son los requerimientos funcionales y no funcionales.

De forma general se describen los actores del sistema y se realiza el modelos de casos de uso con sus correspondientes descripciones.

### **Capítulo IV Construcción de la Solución propuesta.**

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

sistema en términos de componentes organizando a estos de acuerdo a los nodos específicos en el modelo de despliegue

### **Capítulo V. Estudio de factibilidad.**

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



## Capítulo 1. Fundamentación Teórica.

### 1.1 Introducción.

El presente capítulo tiene como objetivo exponer los fundamentos teóricos generales que sirven de punto de partida a la solución del problema antes mencionado. En él se definen algunos conceptos que serán de utilidad en la comprensión del mismo y de la propuesta de solución, también se hace alusión a algunos de los métodos y herramientas de gestión de riesgos que existen en el mundo relacionados con esta temática.

### 1.2 Conceptos Asociados con la Gestión de Riesgos.

#### 1.2.1 Riesgo.

El Riesgo se puede definir como aquella eventualidad que imposibilita el cumplimiento de un objetivo. De manera cuantitativa el riesgo es una medida de las posibilidades de incumplimiento o exceso del objetivo planteado. Así definido un riesgo conlleva dos tipos de consecuencias: ganancias o pérdidas.(4)

En lo relacionado con tecnología, generalmente el riesgo plantea solamente una amenaza, determinando el grado de exposición a la ocurrencia de una pérdida.

La Organización Internacional por la normalización (ISO) define riesgo tecnológico como:

“La probabilidad de que una amenaza se materialice, utilizando vulnerabilidad existente de un activo, o un grupo de activos, generándole pérdidas o daños.”(4)

En la definición anterior se identifican varios elementos que se deben comprender para entender el concepto de Riesgo, ellos son:

**Probabilidad:** Establecer la probabilidad de ocurrencia puede realizarse de manera cuantitativa o cualitativa, pero siempre considerando que la medida no debe contemplar la existencia de ninguna acción paliativa, es decir debe considerarse en cada caso que probabilidad existe de que la amenaza se presente independientemente del hecho que sea o no contrarrestada.

**Amenaza:** Son aquellas acciones que pueden ocasionar consecuencias negativas para la organización.

**Vulnerabilidades:** Son ciertas condiciones inherentes a los activos presentes o en su entorno que facilitan que las amenazas se materialicen, llevando a esos activos a ser vulnerables.

**Activos:** Los activos a reconocer son aquellos relacionados con sistemas de información. Ejemplo el hardware, software, servicios, documentos, entre otros.

**Impactos:** Las consecuencias de las distintas amenazas son siempre negativas. Las amenazas pueden ser financieras, no financieras, de corto plazo, de largo plazo.(4)

Aunque ha habido amplios debates sobre la definición adecuada para riesgo de software, hay acuerdo común en que el riesgo siempre implica dos características:

- Incertidumbre: El acontecimiento que caracteriza al riesgo puede o no puede ocurrir; por ejemplo, no hay riesgos de un 100 por ciento de probabilidad.
- Pérdida: Si el riesgo se convierte en una realidad, ocurrirán consecuencias no deseadas o pérdidas.(5)

#### **1.2.1.1 Categorías de Riesgo.**

Quando se analizan los riesgos es importante cuantificar el nivel de incertidumbre y el grado de pérdidas asociado con cada riesgo. Para hacerlo, se consideran diferentes categorías de riesgos.

Los **riesgos del proyecto** amenazan al plan del proyecto. Es decir, si los riesgos del proyecto se hacen realidad, es probable que la planificación temporal del proyecto se retrase y que los costos aumenten. Los riesgos del proyecto identifican los problemas potenciales de presupuesto, planificación temporal, personal (asignación y organización), recursos, cliente, requisitos y su impacto en un proyecto de software.

Los **riesgos técnicos** amenazan la calidad y la planificación temporal del software que hay que producir. Si un riesgo técnico se convierte en realidad, la implementación puede llegar a ser difícil o imposible. Los riesgos técnicos identifican problemas potenciales de diseño, implementación, de interfaz, verificación y de mantenimiento.

Los **riesgos del negocio** amenazan la viabilidad del software a construir. Los riesgos del negocio a menudo ponen en peligro el proyecto o el producto. Los candidatos para los cinco principales riesgos del negocio son:

1. **Riesgo de Mercado.** Construir un producto o sistema excelente que no quiere nadie en realidad.
2. **Riesgo Estratégico.** Construir un producto que no encaja en la estrategia comercial general de la compañía.
3. Construir un producto que el departamento de ventas no sabe cómo vender.
4. **Riesgo de Dirección.** Perder el apoyo de una gestión experta debido a cambios de enfoque o a cambios de personal.
5. **Riesgo de Presupuesto.** Perder presupuesto o personal asignado.(6)

Existen también **estrategias** frente al riesgo. Por un lado están las **reactivas**, cuyo método es evaluar las consecuencias del riesgo cuando este ya se ha producido (ya no es un riesgo) y actuar en consecuencia. Este tipo de estrategias acarrea consecuencias negativas, al poner el proyecto en peligro. Y por el otro las **preactivas**, que aplican el método de evaluación previa y sistemática de los riesgos y sus posibles consecuencias, a la par que conforman planes de contingencias para evitar y minimizar las consecuencias. Consecuentemente, este tipo de estrategias permite lograr un menor tiempo de reacción ante la aparición de riesgos impredecibles.

#### 1.2.1.2 Identificación del Riesgo.

La identificación del riesgo es un intento sistemático para especificar las amenazas al plan del proyecto (estimaciones, planificación temporal, carga de recursos, etc.). Identificando los **riesgos conocidos** y **predecibles**, el gestor del proyecto da un paso adelante para evitarlos cuando sea posible y controlarlos cuando sea necesario.

Existen dos tipos diferenciados de riesgos en cuanto al **nivel de afectación**, los **riesgos genéricos** y los **riesgos específicos** del producto. Los riesgos genéricos son una amenaza potencial para todos

los proyectos de software. Los riesgos específicos de producto sólo los pueden identificar los que tienen una clara visión de la tecnología el personal y el entorno específico del proyecto en cuestión.(7)

Tanto los riesgos genéricos como los específicos del producto se deberían identificar sistemáticamente. Un método para identificar riesgos es la creación de la lista de comprobación de elementos de riesgo.

La lista de comprobación se puede utilizar para identificar riesgos y se enfoca en un subconjunto de riesgos conocidos y predecibles en las siguientes subcategorías genéricas que se muestran en la siguiente tabla:

Riesgo	Descripción
Tamaño del producto	Riesgos asociados con el tamaño general del software a construir o a modificar.
Impacto en el negocio	Riesgos asociados las limitaciones impuestas por la gestión o por el mercado.
Características del Cliente	Riesgos asociados con la sofisticación del cliente y la habilidad del desarrollador para comunicarse con el cliente en los momentos oportunos.
Definición del proceso	Riesgos asociados con el grado de definición del proceso del software y su seguimiento por la organización de desarrollo.
Entorno de desarrollo	Riesgos asociados con la disponibilidad y calidad de las herramientas se van a emplear en la construcción del producto.
Tecnología a construir	Riesgos asociados con la complejidad del sistema a construir y la tecnología punta que contiene el sistema.
Tamaño y experiencia de la plantilla	Riesgos asociados con la experiencia técnica y de proyectos de los ingenieros del software que van a realizar el trabajo

**Tabla 1: Subcategorías genéricas de riesgos conocidos y predecibles.**

La lista de comprobación de elementos de riesgo puede organizarse de diferentes maneras. Se pueden responder a cuestiones relevantes de cada uno de los temas apuntados anteriormente para

cada proyecto de software. Las respuestas a estas preguntas permiten al planificador del proyecto estimar el impacto del riesgo. Un formato diferente de lista de comprobación de elementos de riesgo contiene simplemente las características relevantes para cada subcategoría genérica. Finalmente, se lista un conjunto de "componentes y controladores del riesgo" junto con sus probabilidades de aparición. Los controladores del rendimiento, el soporte, el coste y la planificación temporal del proyecto se estudian como respuesta a preguntas posteriores.

Existen riesgos en cuanto al criterio del *área que amenazan*, entre los cuales se encuentran los **riesgos del proyecto** que atentan contra los recursos o al plan del proyecto en general, los **riesgos técnicos** que amenazan la calidad o el desempeño del proyecto, y los **riesgos del negocio** que amenazan la viabilidad del software a construir y a la organización que desarrolla el software.

Otro criterio que categoriza a los riesgos es según la *relación con el equipo de proyecto*, estos pueden ser **externos** (están más allá del control o influencia del equipo del proyecto) o los **riesgos internos** que son los relacionados con situaciones que el equipo de proyecto puede controlar o influenciar.

### 1.2.2 Gestión.

En el artículo realizado por Guillermo Restrepo González, se plantea que la Gestión:

“Es el proceso de planear, organizar, dirigir, evaluar y controlar”.(8)

### 1.2.3 Gestión de Riesgo.

La evolución de la tecnología contribuye a un aumento de las oportunidades, pero también a un incremento de los riesgos que hay que gestionar. En la figura 1 se muestra un ejemplo del concepto de gestión de riesgo.

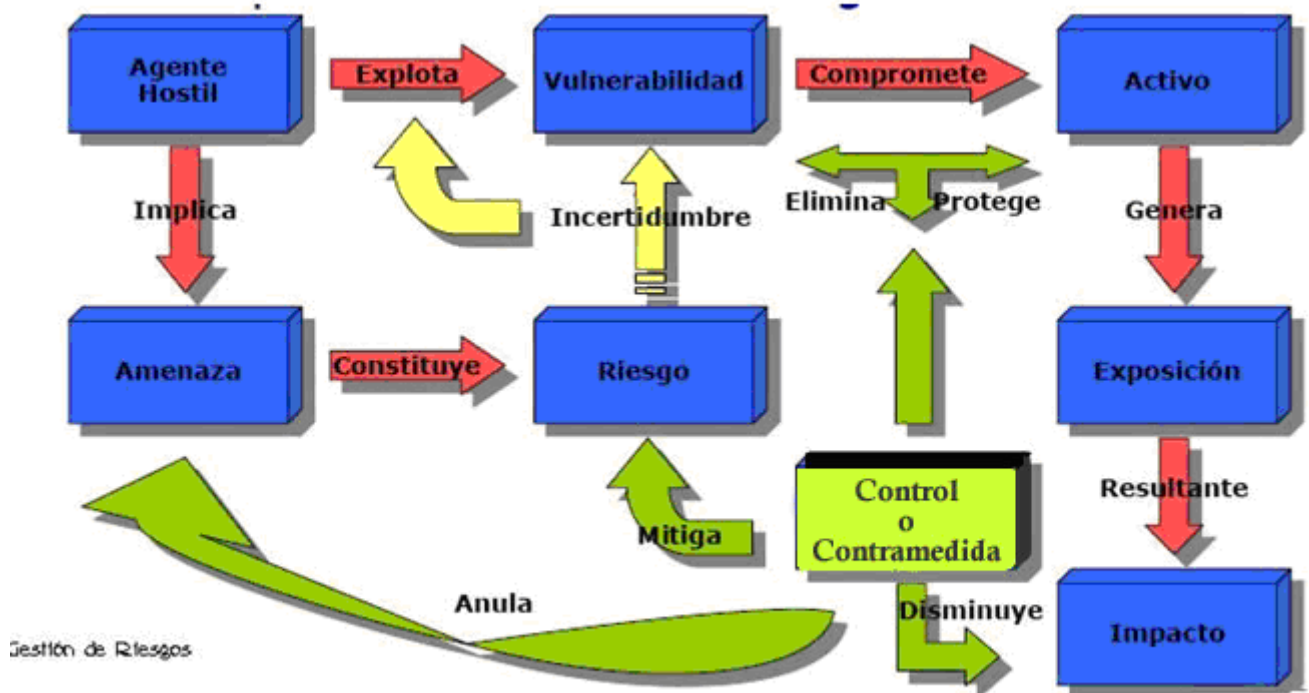


Figura 1: Concepto de Gestión de Riesgo.

**La Gestión de Riesgo se considera:**

- Enfoque sistemático para reducir la probabilidad de riesgos y/o limitar los daños causados por el riesgo mediante el uso de contramedidas adecuadas o acciones preventivas.(9)
- El proceso, distinto de la evaluación del riesgo, consistente en sopesar las alternativas políticas existentes mediante consultas con las partes interesadas, analizar la evaluación del riesgo y otros factores legítimos y, si fuera necesario, seleccionar las opciones de prevención y control.(10)

**1.3 Gestión de Riesgos en Proyectos Informáticos.**

La gestión de riesgos en el ámbito del software procura formalizar conocimiento orientado a la minimización o evitación de riesgos en proyectos de desarrollo de software, mediante la generación de

principios y buenas prácticas de aplicación realista.(11) Hasta el momento se ha propuesto y utilizado diferentes enfoques de gestión del riesgo. Sin embargo, es evidente que pocas organizaciones utilizan todavía de una forma explícita y sistemática métodos específicos para gestionar los riesgos en sus proyectos software.

La administración o gestión de riesgos es un proceso iterativo que se aplica durante todo el proyecto y se desarrolla en cuatro etapas.(12) Los resultados de la administración de riesgos deben ser documentados en un plan de administración de riesgos. La figura 3 refleja el procedimiento.

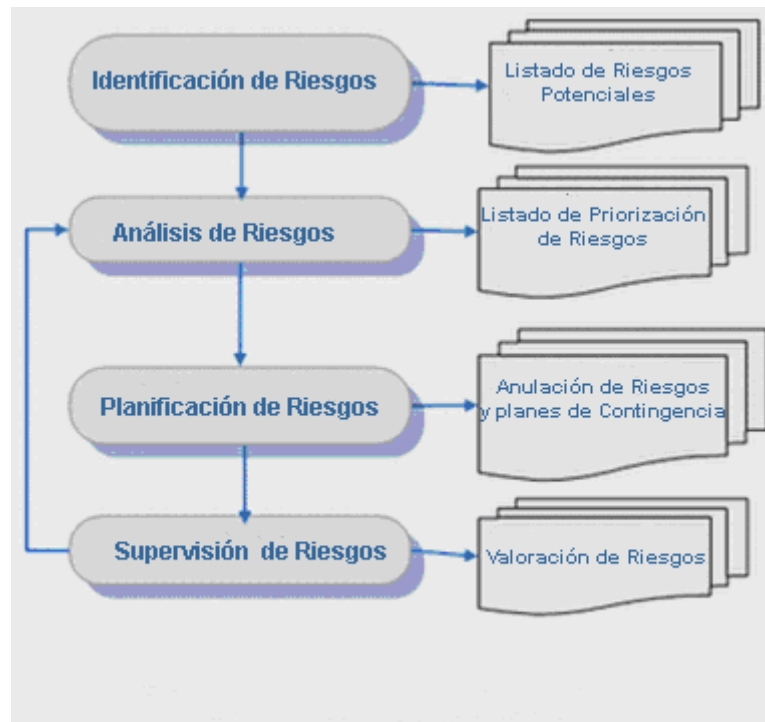


Figura 2: Procedimiento de Gestión de Riesgo

### 1.3.1 Gestión de Riesgos en proyectos informáticos en la Facultad 9.

La problemática actual del software se centra en la incapacidad para estimar tiempo, costo y esfuerzo para el desarrollo de un producto, así como la falta de calidad del producto, provocando que la mayoría de los Proyectos Informáticos fracasen en su intento de cumplir con el cliente o simplemente con el plan de trabajo.

En la facultad 9 son varios los proyectos productivos que se están llevando a cabo pero todos están expuestos a distintos riesgos, ya sean por condiciones de trabajo, obsolescencia, pérdida de

materiales o equipos, trayendo todo esto aparejado un déficit en la obtención de mayores ganancias, mejor eficiencia en la producción y en los servicios que brinda nuestra facultad, ya que actualmente no existe una aplicación capaz de contener la relación de los posibles riesgos a los que está expuesto un proyecto, y que nos muestre cuánto representa la ocurrencia de cada uno de estos para el buen desarrollo de las aplicaciones.

Actualmente en los proyectos productivos de la facultad hay una carencia de conocimientos sobre los marcos para la Gestión de Riesgos y por tanto de su aplicación. De cierta forma, se conocen informalmente los riesgos que podrían afectar el trabajo, pero estos no son registrados y mucho menos se procede a su análisis o gestión.

Esto es una de las cosas que actualmente esta afectando en gran medida a los proyectos que se llevan a cabo en la facultad 9 pues no existe una responsabilidad con la gestión de riesgos cuando se planifica un proyecto en específico.

Lo que se quiere con este trabajo es realizar una aplicaron informática para que toda persona encargada en la gestión de riesgos de su proyecto tenga una herramienta que le haga el trabajo mas fácil, y así poder realizarlo en mas breve tiempo e incluso poder sacar mayor provecho que si se hiciera manualmente.

#### **1.4 Modelos de Gestión de Riesgos.**

La gestión de riesgos ha pasado por tres generaciones de modelos de riesgos en proyectos Informáticos:

- La primera Generación G1 (Casuística)

Esta generación data de principios de los años 80 y está basada en listas 'casuísticas' de riesgos especiales para proyectos, esto consiste en identificar casos de riesgo y extrapolarlos a otros proyectos. En esta generación se definen los Riesgos tecnológicos y las Listas de comprobación de riesgos, y se limitan las tareas a la identificación de riesgos en los proyectos con técnicas basadas en cuestionarios, listas de incidencias y de las medidas para contrarrestarlas.

- La Segunda Generación G2 (Taxonómica)



Es la generación taxonómica de análisis de riesgos en los proyectos. Cocho, Adam y Torralba, apuntan que los modelos de la G2 se han limitado a analizar los riesgos al inicio del proyecto y a planificar medidas. Definen esta visión como “preventiva”, “teorizante” y de medidas “curativas”, improvisadas en mayor o menor medida, durante el avance del proyecto, para paliar los riesgos según se presentan. Posteriormente califican a los modelos de la G2 como “meramente reactivos, con unas relaciones de causa-efecto basadas sólo en una confianza que parte de experiencias poco validadas”.

Esta segunda generación data de principios de los años 90. Está basada en modelos de procesos y eventos. Dentro de esta generación se pueden incluir:

- Modelo de Boehm: De este modelo surgió una de las ideas fundamentales que las metodologías posteriores adoptarían: el análisis de riesgos. Este modelo tiene un carácter iterativo; en sus primeras fases plantea la necesidad de realizar diversas iteraciones dirigidas a mitigar los riesgos más críticos encontrados en el proyecto.
- Modelo de Riesgos del SEI: Este modelo consiste en la propuesta de un cuestionario desarrollado por el Software Engineering Institute (SEI) que tiene como objetivo colaborar en la identificación de riesgos de los proyectos.
- La Tercera Generación G3 (Causal)

Es la generación causal, nacida a mediados de los 90 y referida en particular a proyectos informáticos. Surge de forma simultánea en Europa y en EEUU, partiendo de la preocupación por proyectos de tanto riesgo como la adquisición o el desarrollo de software. Aprovecha los métodos de gestión de riesgos usados en los sistemas. Articula también una causalidad más explicativa y por lo tanto más predictiva entre los elementos del modelo, sobre todo entre los factores de riesgo y sus medidas reductoras o salvaguardas.(13)

Esta es la generación actualmente emergente. Arranca con Eurométodo 96, MAGERIT 97, ISPL 98, etc.

### **1.5 Modelos y herramientas de Gestión de Riesgos existentes.**

Se llevó a cabo un estudio sobre los principales métodos o metodologías de gestión de riesgos utilizados actualmente a nivel internacional así como herramientas software que se basan en los

mismos. También se enfatizó en el modelo de gestión de riesgos Mogerit desarrollado en la UCI y que será utilizado para desarrollar el software en cuestión. A continuación se describen los modelos de gestión de riesgos:

### **1.5.1 MAGERIT.**

MAGERIT es el acrónimo de "Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información de las Administraciones Públicas". Es una metodología de carácter público, perteneciente al Ministerio de Administraciones Públicas. Es un método formal para investigar los riesgos que soportan los sistemas de información, para recomendar las medidas apropiadas que deberían adoptarse para controlar estos riesgos.(7)

MAGERIT persigue los siguientes objetivos:

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

La versión 1.0 de MAGERIT se presenta en siete guías metodológicas:

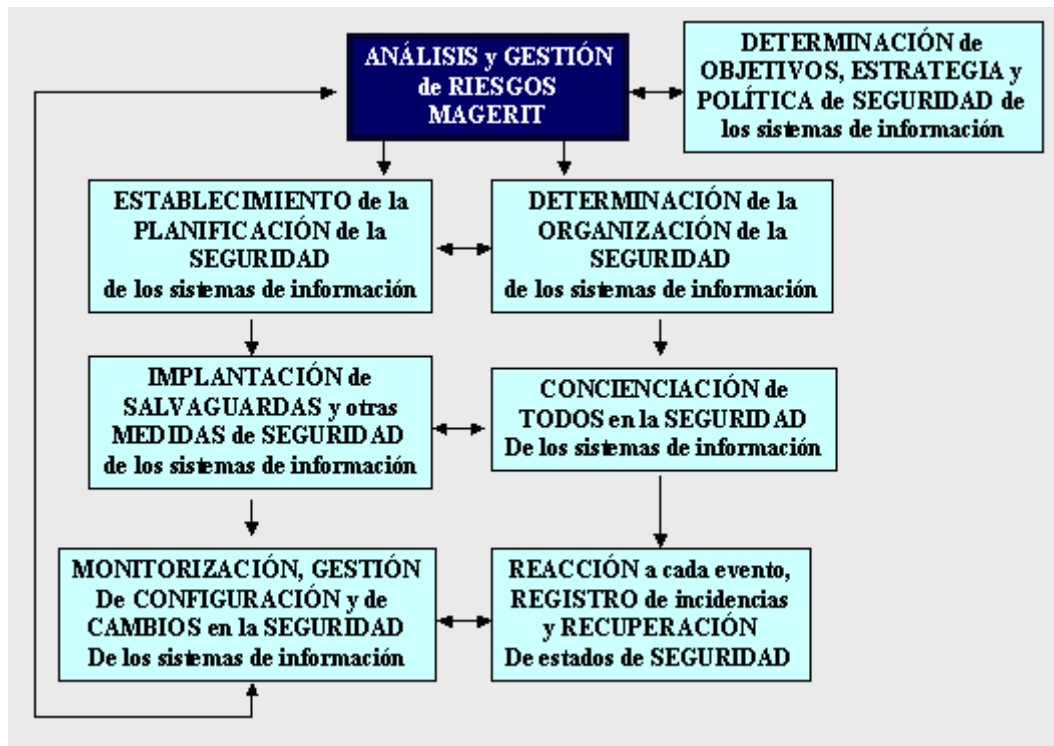


Figura 3: Guías Metodológicas de MAGERIT (v1.0).

### 1.5.2 EUROMÉTODO.

Eurométodo es un proyecto de la Comisión Europea que fue diseñado para ayudar a las organizaciones en la adquisición de sistemas de información efectivos y de los servicios asociados en distintas situaciones. Anima a los clientes y proveedores a controlar los costes y los plazos previstos, gestionar los riesgos y mejorar el entendimiento mutuo.

Los objetivos de Eurométodo son:

- Ayudar al entendimiento mutuo entre clientes y proveedores de proyectos y servicios de sistemas de información dentro de un mercado internacional abierto, proporcionando asesoramiento apoyado por un conjunto de conceptos y terminología que utilizarán en las transacciones

- Mejorar la adquisición de sistemas de información y servicios, teniendo presente la situación del problema y los riesgos asociados.
  - Proporcionar un marco para la unificación de la terminología de los métodos.
- Eurométodo proporciona un marco, es decir, un conjunto de conceptos y una

terminología:

- Para mejorar la relación cliente-proveedor.
- Para unificar los métodos.(14)

La adquisición tiene lugar en tres niveles:

- En el nivel de prestación de servicios o producción del proyecto, los agentes utilizan una gran variedad de métodos y herramientas para tareas como el desarrollo del sistema de información, aseguramiento de la calidad, gestión de la configuración y manejo del ordenador
- En el nivel de gestión del servicio o proyecto, los agentes cuentan con el apoyo de métodos y herramientas de gestión de proyectos
- Eurométodo se ha diseñado para ser utilizado a nivel contractual por la gestión de adquisiciones, en especial dentro de la gestión de contratos.

Eurométodo proporciona un método para definir, planificar y ejecutar la adquisición de un sistema de información y los servicios asociados.

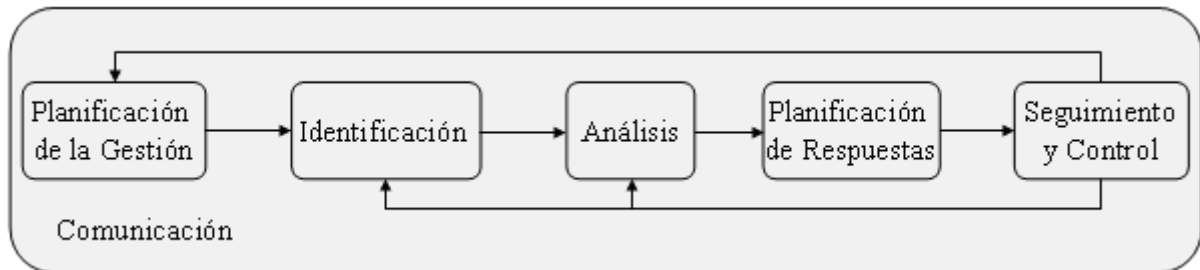
Se utiliza para valorar y determinar:

- La situación del problema y los riesgos asociados.
- El objetivo de la adquisición.
- La estrategia de la adquisición, la adaptación del sistema de información y la prestación de servicios.
- El plan de entregas que describa las relaciones cliente-proveedor a nivel contractual, incluyendo el intercambio de entregables.

### **1.5.3 MoGeRi.**

MoGeRi (Modelo de Gestión de Riesgos) es un modelo cubano diseñado en la Universidad de las Ciencias Informáticas (UCI). Este método se hizo con el objetivo de lograr una GR eficiente en los proyectos de desarrollo de software de la UCI.

El modelo consta de seis procesos que se muestran en la Figura 4: la planificación de la GR, la identificación de los riesgos, el análisis de los riesgos, la planificación de las respuestas, el seguimiento y control y la comunicación de la información sobre los riesgos.(3)



**Figura 4: Submodelo de Procesos.**

El Modelo de GR para la UCI es el que se utilizará como guía para el desarrollo de este trabajo ya que:

- Fomenta la comunicación del equipo de proyecto, dentro de este y de este con su entorno.
- Promueve la reutilización y registro de datos, no solo de los riesgos sino de la información histórica del proyecto.
- Se inserta, apoya y complementa la planificación, seguimiento y control, y de forma general, la Gestión de Proyectos.
- Apoya la gestión de los recursos en el proyecto, pues el impacto de los riesgos afectan directamente los recursos.
- Los procesos, actividades y tareas propuestos, son aplicables en cualquier fase del proyecto, lo que facilita su implantación en todo proyecto donde sea oportuna la GR.
- Recoge las actividades propuestas por estándares de calidad internacionales.
- Los resultados de los instrumentos aplicados ayudaron a identificar otros problemas de investigación.
- Los expertos concuerdan en su efectividad, tanto en la concepción teórica como en los resultados que se obtendrán con su aplicación.
- En resumen, se definen procesos que permiten planificar las actividades sobre los riesgos, identificarlos, analizarlos, planificar las respuestas ante ellos, seguir y controlar los riesgos en el contexto del proyecto, así como comunicar la información generada al respecto en la producción de Software.

### 1.5.4 Herramienta PILAR.

La herramienta PILAR es un procedimiento informático-lógico para el análisis y la gestión de los riesgos de un sistema de información siguiendo la Metodología MAGERIT. Esta herramienta es de uso exclusivo en la administración pública española.

En PILAR se analizan los riesgos en varias dimensiones: confidencialidad, integridad, disponibilidad, autenticidad y trazabilidad. La herramienta PILAR mostrada en la Figura 4 ofrece en Análisis y Gestión de Riesgos un:

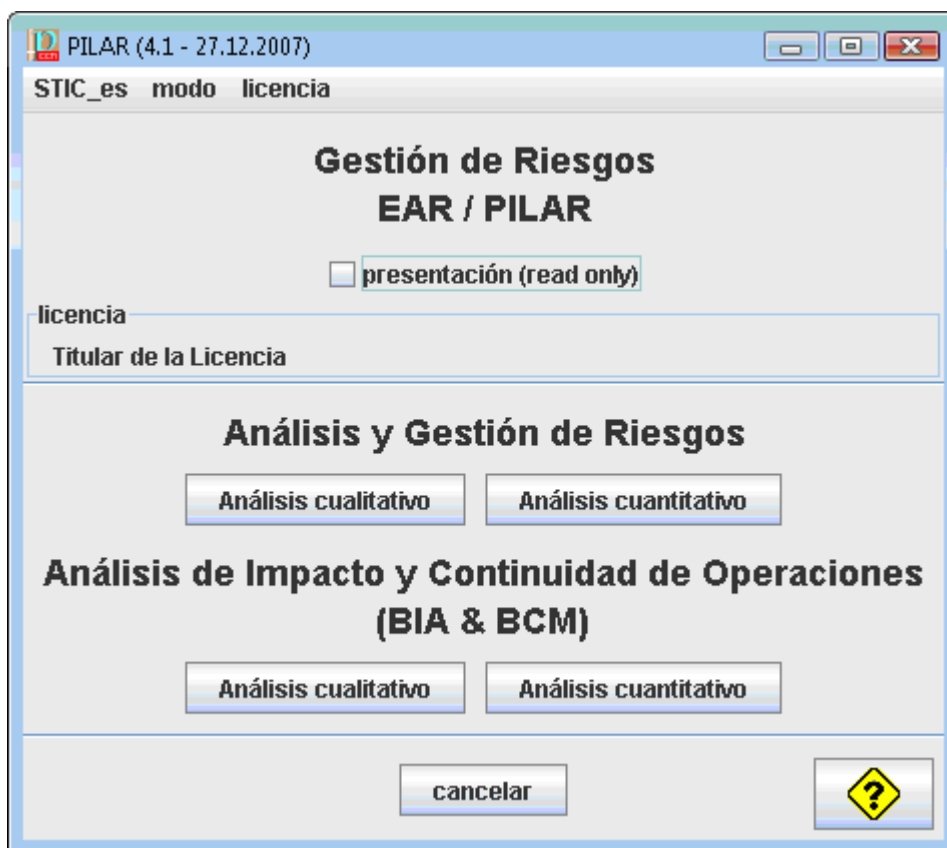


Figura 5: Herramienta PILAR.

*Análisis cualitativo:* Las valoraciones de los elementos (activos, amenazas y salvaguardas) se realizan por niveles, desde la irrelevancia hasta la máxima importancia o criticidad. Esto hace un análisis rápido de que lo más importante es la relativización de impactos y riesgos.

*Análisis cuantitativo:* Las valoraciones son numéricas, típicamente en términos dinerarios. Esto hace el análisis mucho más preciso, si se consigue validar los datos empleados.

El análisis puede llegar a ofrecer resultados de recuperación de la inversión en salvaguardas, en términos de riesgo menguante.

Para tratar el riesgo se proponen:

- salvaguardas (o contra medidas).
- normas de seguridad.
- procedimientos de seguridad.

Analizándose el riesgo residual a lo largo de diversas etapas de tratamiento.

Para tratar el riesgo se proponen:

- salvaguardas (o contra medidas).
- elementos de respaldo (*back up*).
- planes de recuperación de desastres.

Analizándose el impacto residual a lo largo de diversas etapas de tratamiento.(15)

### 1.5.5 Herramienta GxSGSI.

Es una herramienta de gestión de Análisis de Riesgos de Seguridad que incorpora una serie de funcionalidades que complementan la totalidad de controles y especificaciones exigidos por la norma ISO 27001.(16)

The screenshot shows a software interface with a table titled 'Análisis de Impactos'. The table has columns for 'Descripción', 'Tipo', and 'A'. The data is as follows:

Descripción	Tipo	A
15 MINUTOS	D	0
3 HORAS	D	0
1 DIA	D	1
1 SEMANA	D	2
DESTRUCCIÓN PARCIAL	D	3
DESTRUCCIÓN TOTAL	D	3
REVELACION INTERNA	C	0
REVELACION EXTERNA	C	3
ERRORES PEQUEÑA ESCALA	I	1
ERRORES GRAN ESCALA	I	3
MODIFICACIÓN DELIBERADA	I	3
Disponibilidad		3
Confidencialidad		3
Integridad		3
TOTALES		3

Figura 6: GxSGSI Análisis de Impactos.

Sus principales características son:

- Ha sido diseñado para automatizar, agilizar y realizar por completo el análisis de riesgos de seguridad de una organización.
- Su versatilidad de configuración le permite trabajar con multitud de revisiones.
- Es multiempresa.
- Reduce el tiempo y de captura de datos hasta en un 80%.
  
- Genera todos los informes requeridos en una auditoría de certificación ISO 27001 en cuestión de minutos.
  
- No es necesario un alto nivel de calificación en seguridad para poder gestionar el programa.
  
- Proporciona una base sólida para diseñar cualquier plan de continuidad ante desastres.

### **1.6 Conclusiones.**

En este capítulo se abordaron temas y conceptos relacionados con la Gestión de Riesgos para un mayor entendimiento del problema en cuestión. Además se realizó un estudio sobre los principales métodos y software que gestionan riesgos a nivel internacional y nacional.



## **Capítulo 2. Tendencias y Tecnologías Actuales a Desarrollar.**

### **2.1 Introducción.**

En este capítulo se realizó un estudio sobre las principales tecnologías y herramientas existentes y más utilizadas para el diseño y desarrollo de aplicaciones Web en el mundo. Además se hace mención del Lenguaje de programación y Gestor de Base de Datos escogidos para el desarrollo de la aplicación y su justificación, así como la arquitectura y patrón arquitectónico que se siguió, entre otros.

### **2.2 Tecnologías de la Información y las Comunicaciones (TIC).**

Actualmente la instrumentación tecnológica es una prioridad en la comunicación, ya que las tecnologías de la comunicación son la diferencia entre una civilización desarrollada y otra en vías de desarrollo. Éstas poseen la característica de ayudar a comunicarnos porque se desaparecen las distancias geográficas y el tiempo.

Las tecnologías de la información y de las comunicaciones (TIC) son un término que se utiliza actualmente para hacer referencia a una gama amplia de servicios, aplicaciones, y tecnologías, que utilizan diversos tipos de equipos y de programas informáticos, y que a menudo se transmiten a través de las redes de telecomunicaciones.(17)

Actualmente las TIC son muy importantes ya que incluyen una serie de tecnologías que apoyan a la comunicación e información entre personas por ejemplo el acopio de información con la World Wide Web, también el almacenamiento, elaboración, análisis y presentación de la información, incluyendo diferentes medios para textos, datos, gráficos, fotos, audio, tales como Word, Excel, Access, PowerPoint, Sitio Web, Video, sistemas de información geográficas, sistemas del manejo de decisiones, etc. Y la difusión de información por medios ya sean computadoras, teléfonos, fax, equipos de radio, televisor entre otros, y por infraestructura ya sean las líneas fijas, radiosondas o por satélite.

### **2.3. Aplicaciones Web.**

Una aplicación Web es un sistema informático que los usuarios utilizan accediendo a un servidor Web a través de Internet. La creciente popularidad de las aplicaciones Web se debe a sus múltiples ventajas, entre las cuales se pueden citar:

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

### **2.4. Modelo Cliente - Servidor.**

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes.(18)

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

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

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

Ventajas de la arquitectura cliente-servidor:

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

### **2.5 Servidor Web Apache.**

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

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

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

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

Apache ha sido desde abril de 1996 el servidor Web más popular. En noviembre de 2005 la Netcraft Web Server Survey encontró que más del 70% de los sitios Web en Internet usan a Apache como servidor Web, haciéndolo más ampliamente usado que los otros servidores Web de forma combinada.(19)

## 2.6. Patrón Arquitectónico: Arquitectura en Capas.

Un patrón de arquitectura de software describe un problema particular y recurrente del diseño, que surge en un contexto específico, y presenta un esquema genérico y probado de su solución.(20) Este patrón define cómo organizar el modelo de diseño en capas, que pueden estar físicamente distribuidas, lo cual quiere decir que los componentes de una capa sólo pueden hacer referencia a componentes en capas inmediatamente inferiores. Este patrón es importante porque simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores. Además, nos ayuda a identificar qué puede reutilizarse, y proporciona una estructura que nos ayuda a tomar decisiones sobre qué partes comprar y qué partes construir.

Principales estilos de arquitecturas estratificadas de las aplicaciones distribuidas contemporáneas:

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

### 2.6.1 Arquitectura de tres niveles.

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

- La capa de la Presentación.

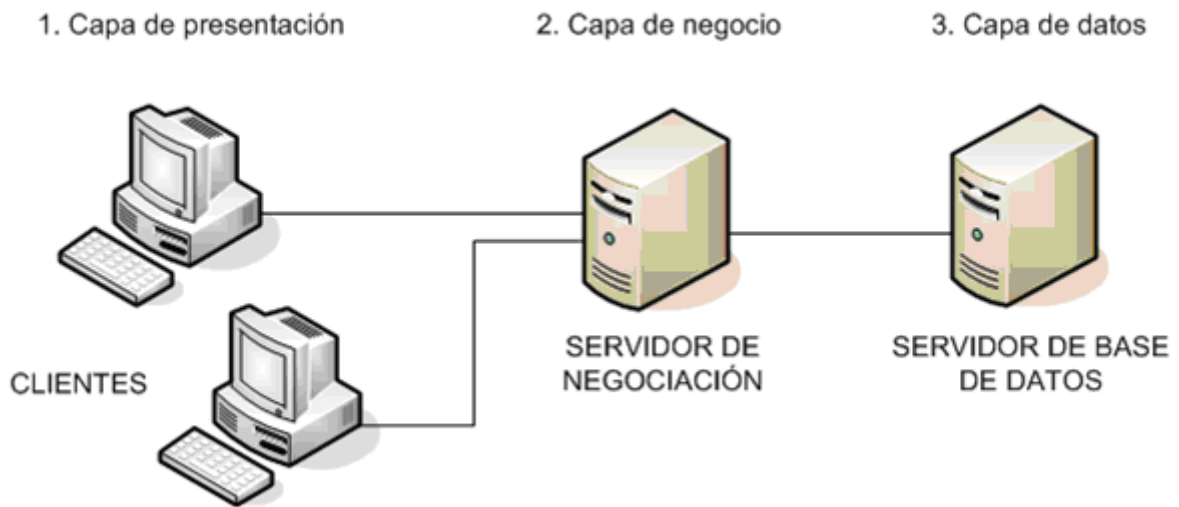
Esta capa reúne todos los aspectos del software que tiene que ver con las interfaces y la interacción con los diferentes tipos de usuarios humanos. Estos aspectos típicamente incluyen el manejo y aspecto de las ventanas, el formato de los reportes, menús, gráficos y elementos multimedia en general.

- La capa de Negocio (Capa Lógica).

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

- La capa de Datos.

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



**Figura 7: Arquitectura de tres capas.**

### **2.7. Sistemas Web Modulares.**

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

No importa el volumen de información que contenga el Sistemas Web Modular, la estructura, diseño y plataforma sobre el cual se construye, permite mantener una respuesta óptima a gran velocidad, aún cuando muchos usuarios solicitan la información al mismo tiempo. En este tipo de Sitios Web la seguridad juega un papel importante, la lectura de la información se realiza de acuerdo a lo que el usuario desea ver, sin arriesgar otra información saturando la pagina actual.

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

## 2.8 Lenguaje de programación.

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

### 2.8.1 Active Server Pages (ASP).

ASP significa Active Server Pages es un componente de Microsoft Internet Information Server. Este lenguaje reemplaza la forma tradicional de intercambiar información entre usuarios, además permite que el código sea incrustado en un documento HTML y que corra en el servidor. ASP es seguro y fácil de programar, se ejecuta sobre el servidor y envía datos de regreso al cliente (a través del buscador). ASP ha pasado por cuatro iteraciones mayores, ASP 1.0, ASP 2.0, ASP 3.0 y ASP.NET que forma parte de la plataforma .NET de Microsoft. Las versiones anteriores a ASP.NET se denominan como *ASP clásico*.

ASP.NET es usado por programadores para construir sitios Web domésticos, aplicaciones Web y servicios XML.

### 2.8.2 ¿Por qué PHP como lenguaje de programación?

PHP es un lenguaje de programación usado generalmente para la creación de contenido para sitios Web habitualmente en combinación con el motor de base datos MySQL, aunque cuenta con soporte nativo para otros motores, incluyendo el estándar ODBC, lo que amplía en gran medida sus posibilidades de conexión.

PHP es un acrónimo recurrente que significa "Hypertext Pre-processor", y se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web. Se utiliza también para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica usando la biblioteca GTK+.(21)

PHP es la gran opción de desarrollo de aplicaciones Web porque el código de PHP se ejecuta sin cambios en una gran variedad de sistemas, y en distintos servidores. Este lenguaje tiene muchas virtudes que lo han convertido en la opción de muchos programadores debido a que:

- Brinda facilidad de aprendizaje del lenguaje.
- Gran cantidad de funciones desarrolladas (PHP incorpora mas de 1000 funciones).
- Tiene una amplia disponibilidad de secuencia de comandos regeneradas en PHP.

- Tiene la capacidad de incrustar código PHP en las páginas HTML, además se vincula fácilmente con las principales bases de datos.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Es multiplataforma y Multisistema Operativo.
- Posee una amplia documentación, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Miles de ejemplos y código fuente disponible.
- Perfecta integración del Apache-PHP-MySQL.
- No depende de un único proveedor de servicios.
- El código fuente es abierto y gratuito.
- Existen gran cantidad de scripts en PHP ya programados y totalmente gratuitos, que permiten fácilmente añadir todo tipo de funcionalidad a nuestra Web (foros, Chat, encuestas, tiendas de comercio electrónico, etc.).
- Biblioteca nativa de funciones sumamente amplia e incluida.

## 2.9 Gestor de bases de datos.

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

### 2.9.1 Microsoft SQL Server.

Es un sistema de gestión de bases de datos relacionales (SGBD) basada en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Así de tener unas ventajas que a continuación se pueden describir.(22)

Entre sus características figuran:

- Soporte de transacciones.
- Gran estabilidad.

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

Desventajas.

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

### **2.9.2 PostgreSQL.**

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

PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. PostGreSQL es un sistema de bases de datos bastante maduro y profesional, pero entre sus inconvenientes es que puede no ser tan rápido como MySQL en cierto rango de datos, por ejemplo MySQL es más rápido entre 10 000 y 32 000 de registros.(23)

### **2.9.3 ¿Por qué MySQL como gestor de Bases de Datos?**

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

Es un Sistema de Gestión de Base de Datos Relacional. El modelo relacional se caracteriza a muy grandes rasgos por disponer que toda la información debe estar contenida en tablas, y las relaciones



entre datos deben ser representadas explícitamente en esos mismos datos. Esto añade velocidad y flexibilidad.

MySQL es un software de código abierto esto quiere decir que es accesible para cualquiera, para usarlo o modificarlo. Se puede descargar MySQL desde Internet y usarlo sin pagar nada, de esta manera cualquiera puede inclinarse a estudiar el código fuente y cambiarlo para adecuarlo a sus necesidades. MySQL usa el GPL (GNU Licencia Publica General) para definir que se puede y no se puede hacer con el software en diferentes situaciones.(24)

Trabajar con MySQL es factible ya que es muy rápido, tiene buenas utilidades de administración, es confiable, robusto y fácil de usar tanto para volúmenes de datos grandes como pequeños, sin límites en los tamaños de registros. Además tiene un conjunto muy práctico de características desarrolladas en cooperación muy cercana con los usuarios. MySQL hoy en día ofrece un rico y muy útil conjunto de funciones. La conectividad, velocidad y seguridad ya que tiene mejor control de acceso a usuarios, hace de MySQL altamente conveniente para acceder a bases de datos en Internet. Además el lenguaje que se utiliza para el desarrollo de este trabajo que es PHP tiene una integración perfecta con MySQL ambas resultan muy útiles para diseñar de forma rápida y eficaz aplicaciones Web dirigidas a bases de datos.

## **2.10 Metodologías y herramientas a utilizar para el desarrollo del sistema.**

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

### **2.10.1 Metodología RUP.**

Es necesario al enfrentarse a la construcción de un software seguir una serie de pasos que lleven a la realización de un producto robusto y que cumpla con todos los requerimientos que a este se le imponen. Para lograr esto se hace necesario el uso de una metodología. La metodología que se emplea para el desarrollo de este trabajo de diploma es RUP (Rational Unified Process, Proceso Unificado de Desarrollo de Software).

El Proceso Unificado es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Sin embargo el Proceso unificado es mas que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas

de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.(25)

Esta metodología hace énfasis en la adopción de las mejores prácticas del desarrollo de software, como una manera de reducir los riesgos inherentes en el desarrollo de una nueva aplicación de software, de esta manera se logran resultados más predecibles, con procesos comunes que mejoran la comunicación y crean un entendimiento de todas las tareas y responsabilidades. Podemos decir además que es un proceso muy organizativo, orientado a objetos, el cual se basa en roles. En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales divididos en 4 fases. Los 6 primeros flujos son conocidos como flujos de ingeniería y los tres últimos como de apoyo. En la Figura 8: RUP en Dos Dimensiones se representa el proceso en el que se grafican los flujos de trabajo y las fases y muestra la dinámica expresada en iteraciones y puntos de control.

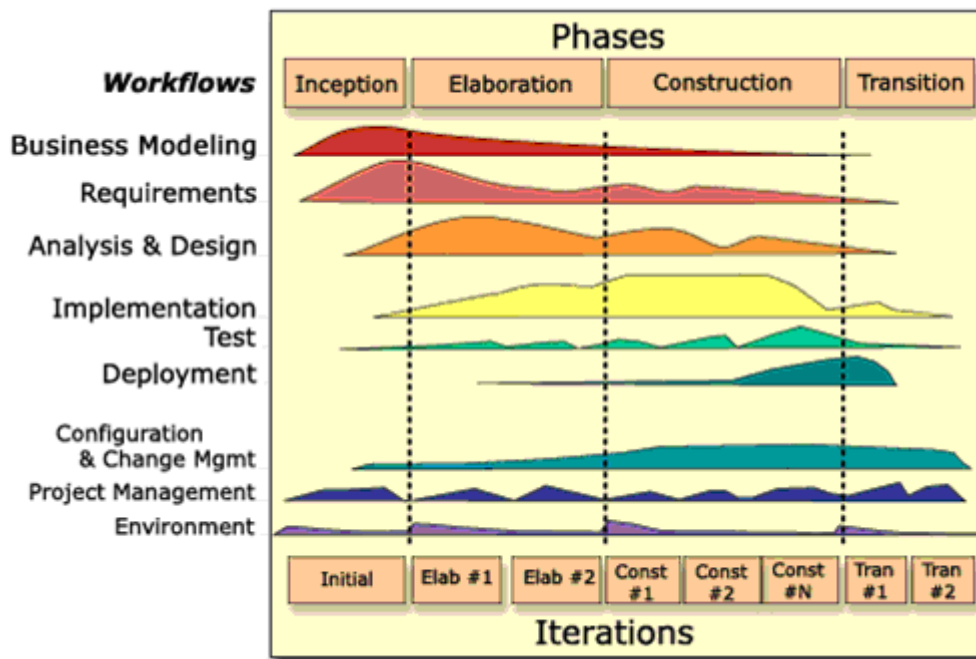


Figura 8: RUP en dos dimensiones.

Entre sus principales características de RUP tenemos las siguientes:

- Iterativo e Incremental.
- Dirigido por los Casos de Uso.
- Centrado en la Arquitectura.(26)

### 2.10.2 Lenguaje de modelado UML.

El Lenguaje unificado de Modelado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar los artefactos de un sistema de software.(26) Está formado por diagramas que contienen elementos y sus relaciones.

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

Además tiene las siguientes características:

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

### 2.10.3 Herramientas Case.

#### 2.10.3.1 Rational Rose.

Es la herramienta Case desarrollada por los creadores de UML que cubren todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes y certificación de las distintas fases. Permite una trazabilidad real entre modelo (análisis y diseño) y el código ejecutable. Rational Rose domina el mercado de herramientas para el análisis, modelamiento, diseño y construcción orientada a objetos, tiene todas las características que los desarrolladores, analistas, y arquitectos exigen soporte UML incomparable, desarrollo basado en componentes con soporte para

arquitecturas líderes en la industria y modelos de componentes, facilidad de uso e integración optimizada.

La corporación Rational Rose ofrece el Proceso Unificado de Rational (RUP), que unifica las mejores prácticas de muchas disciplinas en un consistente y completo proceso del ciclo de vida, que permite al equipo de desarrollo disminuir los tiempos de liberación, además de hacer más predecible el software que ellos producen.

### 2.10.3.1 Embarcadero ERStudio 6.6

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

#### Características:

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

Tiene como **ventajas:**

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

#### **2.10.4 Macromedia Dreamweaver 8.0**

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

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

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

#### **2.10.5 Zend Studio**

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

un comprensivo conjunto de herramientas de edición, depurado, análisis, optimización y bases de datos, Zend Studio acelera los ciclos de desarrollo y simplifica los proyectos complejos.

#### **2.10.6 Conclusiones.**

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

## Capítulo 3. Características del Sistema.

### 3.1 Introducción.

En el presente capítulo se hace la descripción de la propuesta de sistema, se describe el objeto de estudio, así como su modelo de dominio donde se describen las clases que están involucradas. Se especifican los detalles de la construcción de la herramienta, se realiza la propuesta del sistema, se analizan los requerimientos funcionales y no funcionales, y los casos de uso del sistema.

De forma general se realiza el modelo de casos de uso del sistema y sus descripciones correspondientes.

### 3.2 Modelo de Dominio.

El Modelo de Dominio (o Modelo Conceptual) es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés, y enlaza estos objetos unos con otros. La identificación y la asignación de un nombre para estos objetos ayudan a desarrollar un glosario de términos que permitirán comunicarse mejor a todos los que están trabajando en el sistema. Más adelante, los objetos del dominio ayudan a identificar algunas de las clases que se analizan y diseñan en el sistema.

#### 3.2.1 ¿Cuándo se aplica un modelo de Dominio?

- ✓ Los flujos de información son difusos (múltiples orígenes, sólo eventos, sucesos).
- ✓ Imposibilidad de determinar subsistemas (exceso de interconexiones).
- ✓ Solapamiento de responsabilidades.
- ✓ Múltiples responsabilidades.
- ✓ Difícil establecimiento de reglas de funcionamiento.

Resumiendo, se aplica el modelo de dominio cuando no es posible encontrar una estructura en los procesos de negocios, es decir, estos no están bien definidos y no se pueden determinar con claridad sus fronteras ni quienes se benefician de estos.

#### 3.2.2 Formas típicas de las clases del dominio.

- ✓ Objetos del entorno que representan cosas que se manipulan.
- ✓ Objetos del mundo real y conceptos de los que el futuro sistema debe de hacer seguimiento.
- ✓ Sucesos que ocurrirán o han ocurrido.

Un modelo conceptual es una descripción del dominio de un problema real, no es una descripción del diseño del software.

### 3.2.3 Conceptos principales del entorno de la aplicación.

Antes de comenzar con los conceptos generales del dominio, es preciso determinar el entorno organizacional del mismo, el cual se realiza teniendo presente los siguientes aspectos:

- ✓ Determinar organigrama.
- ✓ Determinar eventos.
- ✓ Determinar objetos reales.
- ✓ Determinar conceptos.
- ✓ ¿Quién o quienes participan?

**Proyecto:** Es el entorno en el que se realizará la gestión de riesgo.

**Gestión de Riesgo:** Conjunto de acciones que se llevan a cabo para la disminución de la exposición a los riesgos en un proyecto.

**Proceso:** Parte fundamental por la que esta compuesta la gestión de riesgos.

**Equipo de GR:** Conjunto de personas encargados de realizar los diferentes procesos dentro de la gestión de riesgos.

**Registro de Riesgo:** Documento en el cual se registran todas las características de cada uno de los riesgos.

**Mogeri:** Modelo para la gestión de riesgos.

**Riesgo:** La probabilidad de que una amenaza se materialice, utilizando vulnerabilidad existente de un activo, o un grupo de activos, generándole pérdidas o daños.



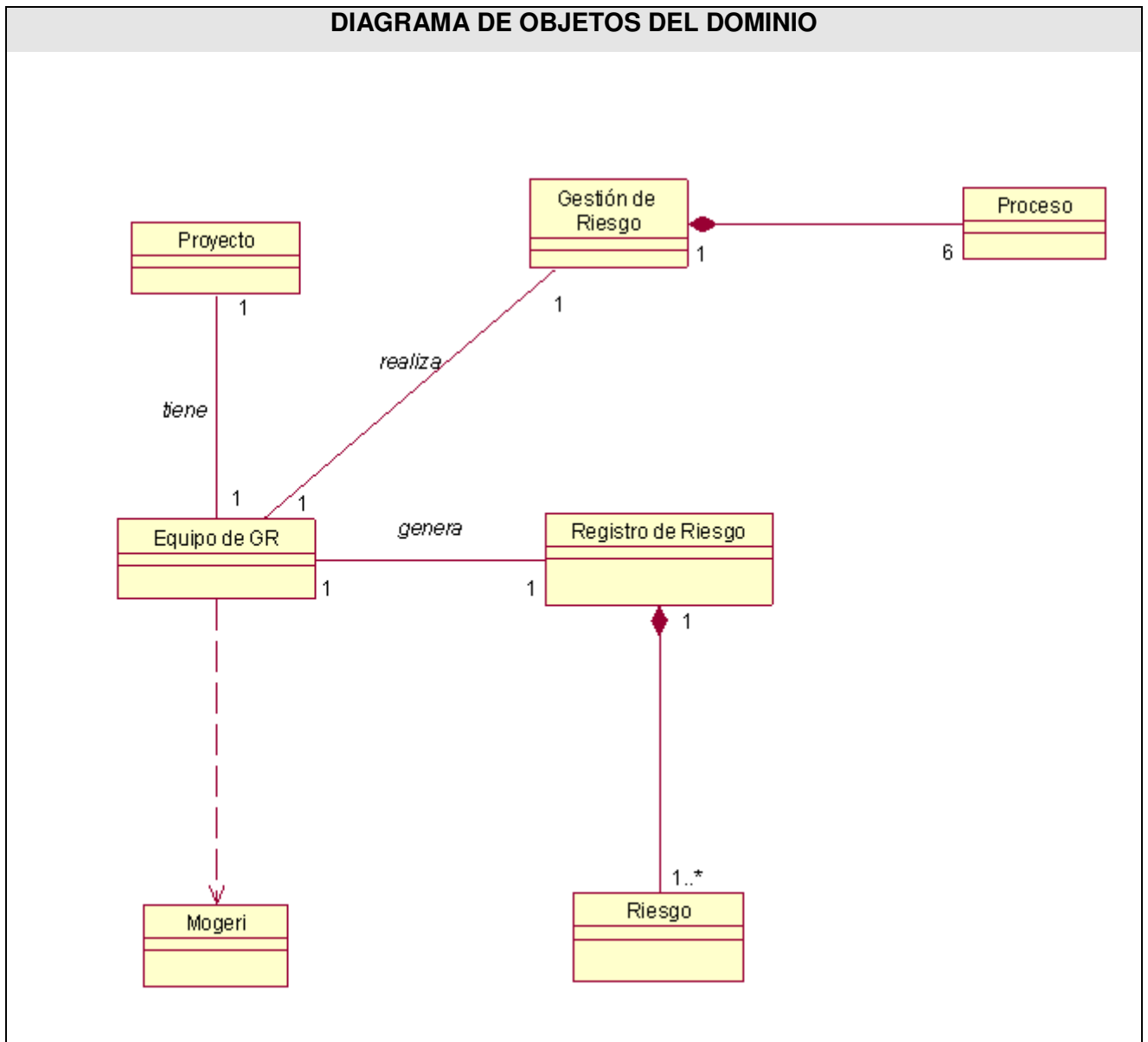


Figura 9: Diagrama de objetos del dominio.

### 3.3 Especificación de los requisitos de software.

#### 3.3.1 Requerimientos Funcionales.

- R1. Gestionar proyectos.
  - R1.1 Insertar proyectos.
  - R1.2 Modificar proyectos.

- R1.3 Eliminar proyectos.
- R2. Gestionar equipo.
  - R2.1 Insertar equipo.
  - R2.2 Modificar equipo.
  - R2.3 Eliminar equipo.
- R3. Autenticar usuario.
- R4. Gestionar riesgos.
  - R4.1 Insertar riesgos.
  - R4.2 Modificar riesgos.
  - R4.3 Eliminar riesgos.
  - R4.4 Mostrar Registro de Riesgos.
- R5. Realizar análisis.
  
- R6. Mostrar riesgos priorizados.
- R7. Gestionar estrategias.
  - R7.1 Insertar estrategias.
  - R7.2 Eliminar estrategias.
  - R7.3 Modificar estrategias.
  
- R8. Gestionar respuestas.
  - R8.1 Insertar respuestas.
  - R8.2 Eliminar respuestas.
  - R8.3 Modificar respuestas.
- R9. Calcular métricas.
- R10. Gestionar experiencias.
  - R10.1 Insertar experiencias.
  - R10.2 Mostrar experiencias.

### 3.3.2 Requerimientos No Funcionales.

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

**Apariencia o interfaz externa.**

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

**Usabilidad.**

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

**Rendimiento.**

Debido a la importancia de la información que procesamos en nuestra aplicación, el sistema deberá de ser lo más estable y confiable posible.

**Soporte.**

Servidor:

Se requiere de cualquier servidor con las siguientes características:

Apache (Servidor Web) Versión 2 o superior

PHP (Lenguaje de programación) Versión 5 o superior

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

Cliente:

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

**Portabilidad.**

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

**Seguridad.**

Existencia de distintos roles que establezcan las acciones que pueden realizar los usuarios.

Identificar al usuario antes de que pueda realizar cualquier acción sobre el sistema.

Verificación sobre acciones irreversibles (por ejemplo las eliminaciones).

Se usan mecanismos de encriptación de los datos que por cuestiones de seguridad no deben viajar al servidor en texto claro, como es el caso de las contraseñas.

**Políticos-Culturales y Legales.**

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

**Confiabilidad.**

Garantía de un tratamiento adecuado de las excepciones y validación de las entradas del usuario para evitar entradas inadecuadas.

El sistema debe tener soporte para recuperación ante fallos y errores.

### 3.4 Modelo del Sistema

#### 3.4.1 Actores del Sistema.

Actores	Justificación
Administrador	Persona encargada de gestionar a los recursos del sistema. El administrador tiene la posibilidad de insertar, modificar o eliminar cada uno de los proyectos del sistema.
Analista	Realiza el análisis cualitativo a los riesgos.
Documentador	Documenta las experiencias personales.
Identificador	Gestiona los datos de los riesgos del proyecto.
Líder	Gestiona los datos de los integrantes del equipo
Planificador de Respuestas	Gestiona los datos de las estrategias de los riesgos del proyecto.
Seguimiento y Control	Realiza el cálculo de métricas a los proyectos.
Usuario	Accede al sistema

Tabla 2: Actores del sistema.

#### 3.4.2 Casos de uso.

Cod.	Nombre de caso de uso	Paquete	Justificación de la selección.
1.	Autenticarse	Seguridad	Este caso de uso es de vital importancia para la seguridad del sistema.
2.	Calcular Métricas	Gestión de Riesgos	Se realiza el cálculo de métricas a los proyectos.
3.	Gestionar equipo	Gestión de Riesgos	Se gestionan los datos de los integrantes del equipo.
4.	Gestionar estrategias	Gestión de Riesgos	Se gestionan los datos de las estrategias de los riesgos del proyecto.
5.	Gestionar proyecto	Seguridad	Se gestionan los datos de los proyectos.
6.	Gestionar respuestas	Gestión de Riesgos	Se Gestionar los datos de las respuestas a las estrategias de los riesgos del proyecto.

<b>7.</b>	Gestionar riesgos	Gestión de Riesgos	Se gestionan los datos de los riesgos del proyecto.
<b>8.</b>	Mostrar experiencias	Documentar Experiencias	Se muestran las experiencias personales.
<b>9.</b>	Mostrar Registro Riesgos	Documentar Experiencias	Muestra el Registro de Riesgos.
<b>10.</b>	Mostrar riesgos priorizados	Gestión de Riesgos	Listar todos los riesgos priorizados por la exposición.
<b>11.</b>	Realizar análisis	Gestión de Riesgos	Realizar el análisis cualitativo a los riesgos.
<b>12.</b>	Documentar Experiencias	Documentar Experiencias	Documentar las experiencias personales.

**Tabla 3. Casos de uso del Sistema.**

3.4.3 Diagrama de Casos de Uso del Sistema.

3.4.3.1 Paquete Documentar Experiencias.

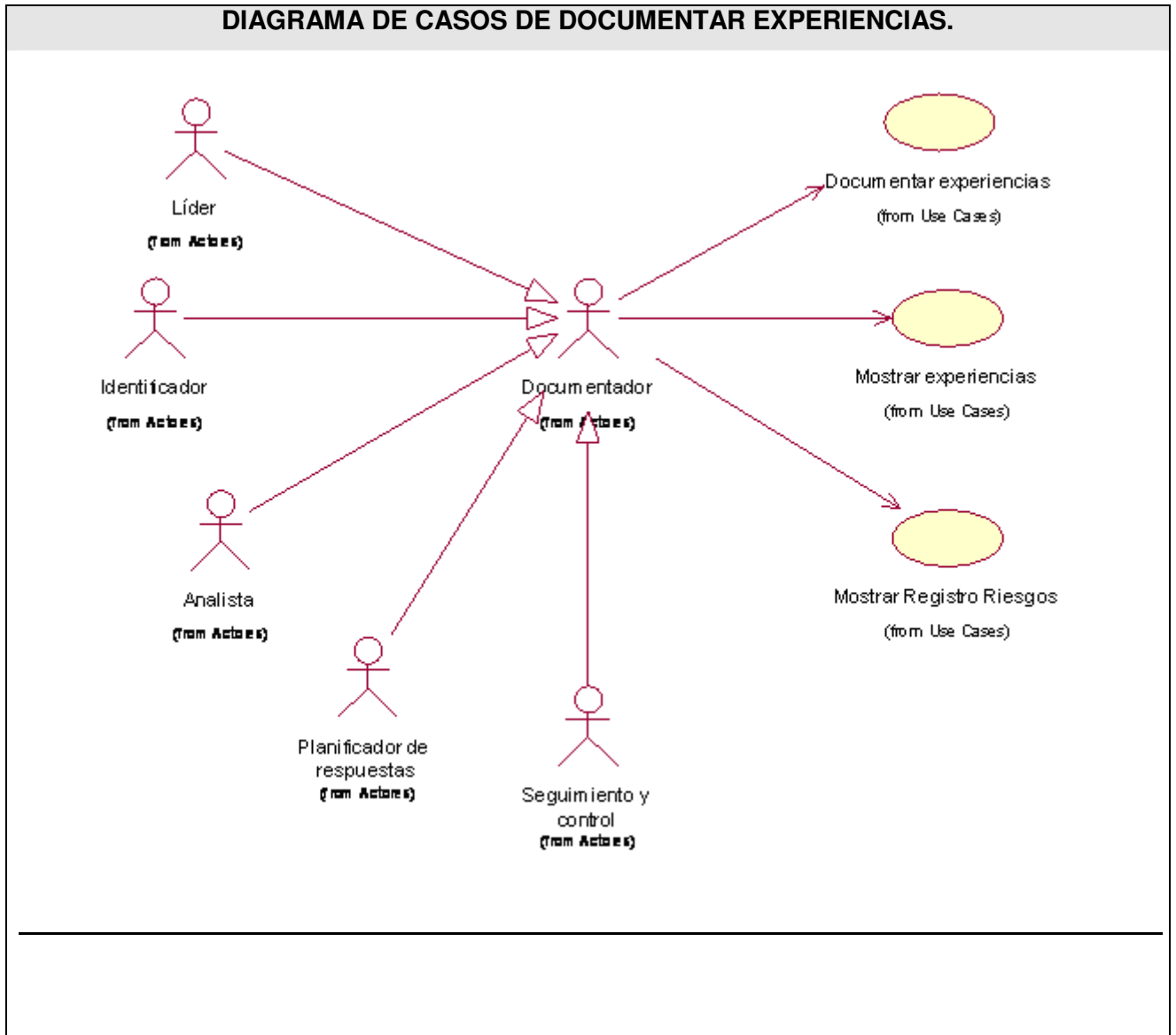


Figura 10. Diagrama de CU de Documentar Experiencias.

3.4.3.2 Paquete Gestionar Riesgos.

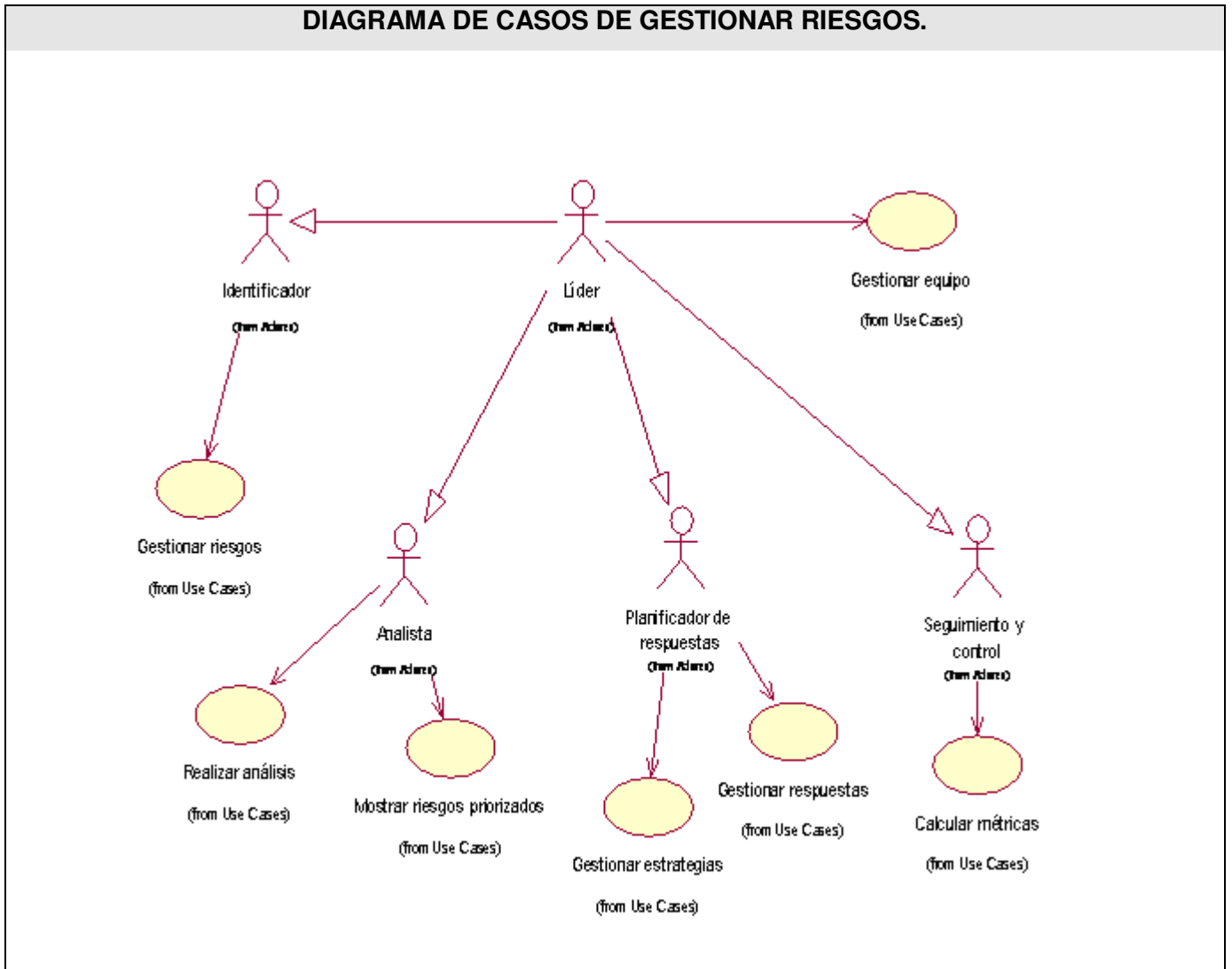


Figura 11. Diagrama de CU de Gestionar Riesgos.

3.4.3.3 Paquete seguridad

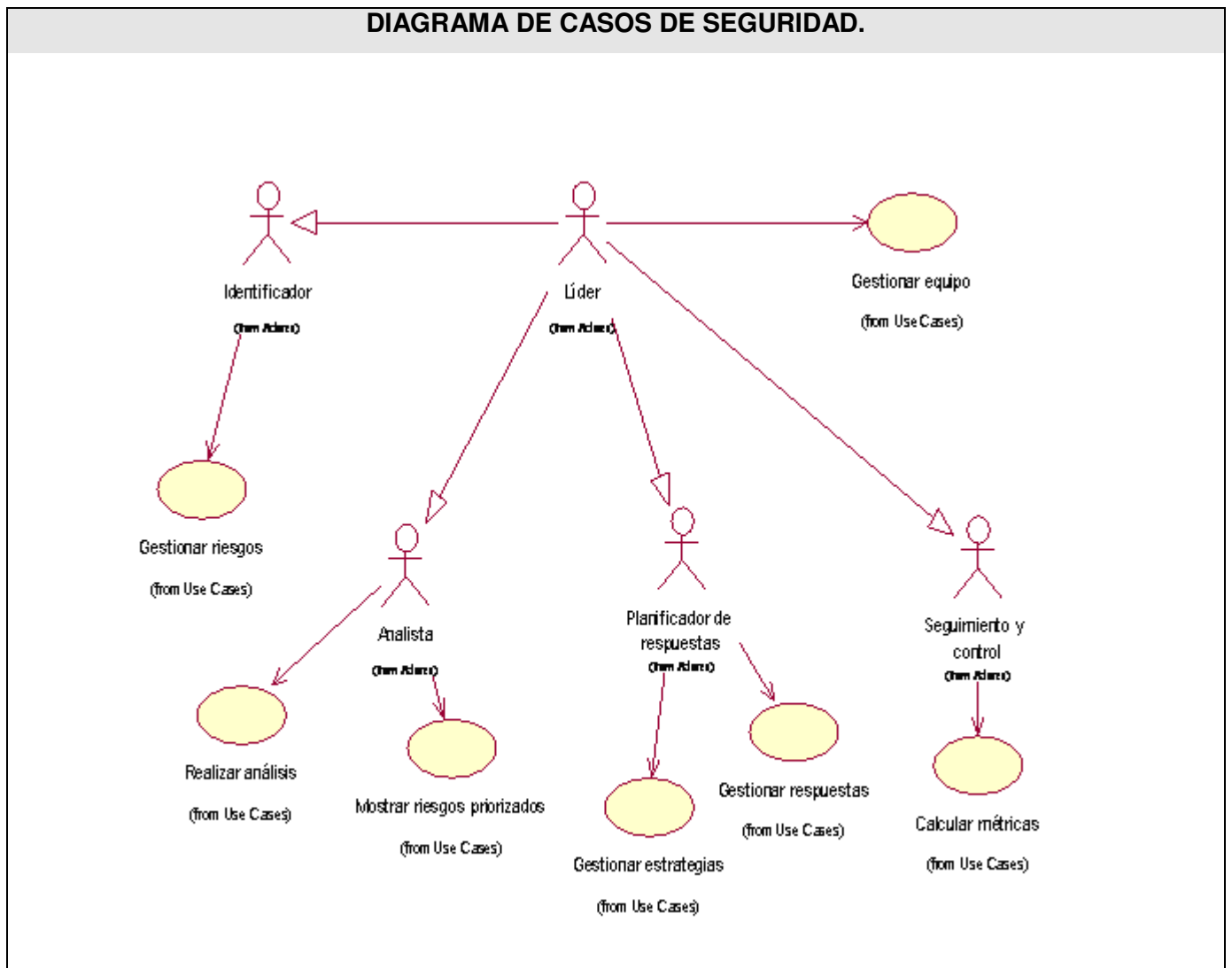


Figura 12. Diagrama de CU de Seguridad



3.5 Descripción de Casos de uso.

<b>Nombre del CU</b>	<b>Autenticarse</b>	
<b>Actores</b>	Usuario (inicia)	
<b>Objetivo</b>	Brindar la sesión requerida en dependencia del usuario que sea.	
<b>Resumen</b>	El CU se inicia cuando el usuario introduce sus datos del dominio UCI en el sistema para autenticarse; y en este, según los datos introducidos le son asignados al usuario sus privilegios y en caso de que no sean correctos se le niega el acceso mostrando un mensaje de error.	
<b>Referencias</b>	<b>R3.</b>	
<b>Precondiciones</b>	Debe ser un usuario del dominio UCI.	
<b>Poscondiciones</b>	El sistema otorga los permisos correspondientes a cada usuario.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
	1. El sistema muestra la interfaz para autenticarse.	
2. El usuario escribe usuario y contraseña del dominio UCI.	2.1 El sistema verifica que los datos entrados por el usuario sean correctos. 2.2 El sistema muestra la información según los privilegios del usuario.	
<b>Curso Alternativo de los Eventos</b>		
Acción 2.1	Si los datos no son correctos el sistema muestra un mensaje donde lo informa y se retorna paso 2.	
<b>Prioridad</b>	Crítico	
<div style="border: 1px solid gray; padding: 10px; background-color: #f0f0f0;"> <p><b>Autenticarse</b></p> <hr/> <p>Por favor, ingrese su nombre de usuario y contraseña del dominio UCI.</p> <p style="text-align: right;">                     Usuario: <input type="text"/>                      Contraseña: <input type="password"/> </p> <p style="text-align: right;"><input type="button" value="Enviar"/></p> </div>		

**Tabla 4. Descripción del CU Autenticarse.**

<b>Nombre del CU</b>	<b>Gestionar Proyectos</b>	
<b>Actores</b>	Administrador (inicia).	
<b>Objetivo</b>	Gestionar los datos de los proyectos.	
<b>Resumen</b>	El administrador tiene la posibilidad de insertar, modificar o eliminar cada uno de los proyectos del sistema.	
<b>Referencias</b>	<b>R1.</b>	
<b>Precondiciones</b>	El Administrador debe estar previamente autenticado.	
<b>Pos condiciones</b>	El Administrador debe haber insertado, eliminado o modificado uno o más proyectos.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
1. El administrador selecciona la opción "Gestión de proyectos" del menú.	2. El sistema muestra una interfaz con la lista de los proyectos y diferentes opciones que le permiten al administrador:	
2.1 Insertar, ver sección "Insertar nuevo proyecto".		
2.2. Eliminar, ver sección "Eliminar proyecto".		
2.3. Modificar, ver sección "Modificar proyecto".		
<b>Sección1. Insertar nuevo proyecto</b>		
3.El administrador toma la opción de insertar nuevo proyecto	4. El sistema muestra la interfaz para agregar al proyecto	
5. El administrador introduce los datos del nuevo proyecto y selecciona la opción "Aceptar".	6. El sistema valida los datos de entrada.	
	7. El sistema almacena la información en su base de datos y se muestra el listado con el nuevo proyecto.	
	8. Retornar al paso 2, culminando así el caso de uso.	
<b>Curso alternativo de los eventos</b>		
Acción 6	6.1 El sistema solicita los datos del nuevo proyecto ya que hubo un error a la hora de entrar los datos.	
	6.2 El sistema retoma el paso 2.	

<b>Sección 2. Eliminar proyecto</b>																	
3. El administrador elige la opción "Eliminar proyecto".	4. El sistema elimina el proyecto seleccionado de la base de datos.																
	5. El sistema muestra el nuevo listado sin el proyecto																
	6. Retomar el paso 2.																
<b>Sección 3. Modificar investigador.</b>																	
3. El administrador selecciona la opción de "Modificar proyecto".	4. El sistema muestra la interfaz para modificar el proyecto																
	5. El sistema modifica los datos seleccionados por el administrador.																
	6. El sistema muestra el nuevo listado con los datos del proyecto ya modificados.																
	7. Retomar paso 2.																
<b>Curso alterno de los eventos</b>																	
3.1 El administrador selecciona la opción de cancelar	3.1.2 El sistema cancela la operación de modificar proyecto.																
	3.1.3 Retomar paso2.																
<b>Prioridad</b>	Crítico																
<p><b>Gestión de Proyectos</b></p> <table border="1"> <thead> <tr> <th>No.</th> <th>Nombre del Proyecto</th> <th>Líder</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Administrativo</td> <td>Osmani González Cabrera</td> <td>Modificar   Eliminar</td> </tr> <tr> <td>2</td> <td>A jugar</td> <td>Lazaro Yasmany Benito Ledesma</td> <td>Modificar   Eliminar</td> </tr> <tr> <td colspan="4" style="text-align: right;">Insertar Nuevo Proyecto</td> </tr> </tbody> </table>		No.	Nombre del Proyecto	Líder	Acción	1	Administrativo	Osmani González Cabrera	Modificar   Eliminar	2	A jugar	Lazaro Yasmany Benito Ledesma	Modificar   Eliminar	Insertar Nuevo Proyecto			
No.	Nombre del Proyecto	Líder	Acción														
1	Administrativo	Osmani González Cabrera	Modificar   Eliminar														
2	A jugar	Lazaro Yasmany Benito Ledesma	Modificar   Eliminar														
Insertar Nuevo Proyecto																	

Tabla 5. Descripción del CU Gestionar Proyectos.

<b>Nombre del CU</b>	<b>Gestionar Equipos</b>
<b>Actores</b>	Líder (inicia).
<b>Objetivo</b>	Gestionar los datos de los integrantes del equipo.
<b>Resumen</b>	El líder tiene la posibilidad de insertar, modificar o eliminar cada uno de los integrantes del equipo del sistema.
<b>Referencias</b>	<b>R2.</b>
<b>Precondiciones</b>	El líder debe estar previamente autenticado.
<b>Pos condiciones</b>	El líder debe haber insertado, eliminado o modificado uno o más integrantes del equipo.
<b>Curso Normal de los Eventos</b>	

Acciones del Actor	Respuesta del Sistema
1. El administrador selecciona la opción "Gestionar equipo" del menú.	2. El sistema muestra una interfaz con la lista de los integrantes del equipo y diferentes opciones que le permiten al líder:
2.1 Insertar, ver sección "Insertar nuevo integrante".	
2.2. Eliminar, ver sección "Eliminar integrante".	
2.3. Modificar, ver sección "Modificar integrante".	
<b>Sección1. Insertar nuevo integrante</b>	
3.El líder toma la opción de insertar nuevo integrante del equipo	4. El sistema muestra la interfaz para agregar al integrante
5. El líder introduce los datos del nuevo integrante y selecciona la opción "Aceptar".	6. El sistema valida los datos de entrada.
	7. El sistema almacena la información en su base de datos y se muestra el listado con el nuevo integrante.
	8. Retornar al paso 2, culminando así el caso de uso.
<b>Curso alternativo de los eventos</b>	
Acción 6	6.1 El sistema solicita los datos del nuevo integrante ya que hubo un error a la hora de entrar los datos.
	6.2 El sistema retoma el paso 2.
<b>Sección 2. Eliminar integrante</b>	
3. El líder elige la opción "Eliminar proyecto".	4. El sistema elimina el proyecto seleccionado de la base de datos.
	5. El sistema muestra el nuevo listado sin el integrante.
	6. Retomar el paso 2.
<b>Sección 3. Modificar integrante.</b>	
3. El líder selecciona la opción de "Modificar integrante".	4. El sistema muestra la interfaz para modificar el integrante.
	5. El sistema modifica los datos seleccionados por el líder.
	6. El sistema muestra el nuevo listado con los datos del integrante ya modificados.
	7. Retomar paso 2.
<b>Curso alternativo de los eventos</b>	
3.1 El líder selecciona la opción de cancelar	3.1.2 El sistema cancela la operación de modificar integrante.
	3.1.3 Retomar paso2.
<b>Prioridad</b>	Crítico

Gestión de integrantes			
No.	Nombre	Rol	Acción
1	Lazaro Yasmany Benito Ledesma	Líder	Modificar   Eliminar
2	Lazaro Yasmany Benito Ledesma	Identificador	Modificar   Eliminar
3	Yudelis Delgado Sanabria	Identificador	Modificar   Eliminar
4	Yudelis Delgado Sanabria	Analista	Modificar   Eliminar
5	Regla María Ramos Suárez	Analista	Modificar   Eliminar

Insertar Nuevo Integrante

Tabla 6. Descripción del CU Gestionar Equipos.

<b>Nombre del CU</b>	<b>Gestionar Riesgos</b>	
<b>Actores</b>	Identificador (inicia).	
<b>Objetivo</b>	Gestionar los datos de los riesgos del proyecto.	
<b>Resumen</b>	El Identificador tiene la posibilidad de insertar, modificar o eliminar cada uno de los riesgos del proyecto del sistema.	
<b>Referencias</b>	<b>R4.</b>	
<b>Precondiciones</b>	El identificador debe estar previamente autenticado.	
<b>Pos condiciones</b>	El identificador debe haber insertado, eliminado o modificado uno o más integrantes del equipo.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
1. El identificador selecciona la opción "Gestionar riesgos" del menú.	2. El sistema muestra una interfaz con la lista de los riesgos y diferentes opciones que le permiten al identificador:	
2.1 Insertar, ver sección "Insertar nuevo riesgo".		
2.2. Eliminar, ver sección "Eliminar riesgo".		
2.3. Modificar, ver sección "Modificar riesgo".		
<b>Sección1. Insertar nuevo riesgo.</b>		
3. El identificador toma la opción de	4. El sistema muestra la interfaz para agregar el riesgo.	

insertar un nuevo riesgo.															
5. El identificador introduce los datos del nuevo riesgo y selecciona la opción "Aceptar".	6. El sistema valida los datos de entrada.														
	7. El sistema almacena la información en su base de datos y se muestra el listado con el nuevo riesgo.														
	8. Retornar al paso 2, culminando así el caso de uso.														
<b>Curso alternativo de los eventos</b>															
Acción 6	6.1 El sistema solicita los datos del nuevo riesgo ya que hubo un error a la hora de entrar los datos.														
	6.2 El sistema retoma el paso 2.														
<b>Sección 2. Eliminar riesgo.</b>															
3. El identificador elige la opción "Eliminar riesgo".	4. El sistema elimina el riesgo seleccionado de la base de datos.														
	5. El sistema muestra el nuevo listado sin el riesgo.														
	6. Retomar el paso 2.														
<b>Sección 3. Modificar riesgo.</b>															
3. El identificador selecciona la opción de "Modificar riesgo".	4. El sistema muestra la interfaz para modificar el riesgo.														
	5. El sistema modifica los datos seleccionados por el identificador.														
	6. El sistema muestra el nuevo listado con los datos del riesgo ya modificados.														
	7. Retomar paso 2.														
<b>Curso alternativo de los eventos</b>															
3.1 El identificador selecciona la opción de cancelar	3.1.2 El sistema cancela la operación de modificar riesgo.														
	3.1.3 Retomar paso2.														
<b>Prioridad</b>	Crítico														
<b>Gestion de riesgos</b>															
<table border="1"> <thead> <tr> <th>No.</th> <th>Nombre del riesgo</th> <th>Descripcion</th> <th>Insertar Riesgo</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Que existan problemas con la electricidad</td> <td>Provoca retraso en el proyecto</td> <td>Modificar   Eliminar</td> </tr> <tr> <td>2</td> <td>Que se pierda mucho tiempo en reuniones</td> <td>Puede suceder que se realicen demasiadas reuniones.</td> <td>Modificar   Eliminar</td> </tr> </tbody> </table>				No.	Nombre del riesgo	Descripcion	Insertar Riesgo	1	Que existan problemas con la electricidad	Provoca retraso en el proyecto	Modificar   Eliminar	2	Que se pierda mucho tiempo en reuniones	Puede suceder que se realicen demasiadas reuniones.	Modificar   Eliminar
No.	Nombre del riesgo	Descripcion	Insertar Riesgo												
1	Que existan problemas con la electricidad	Provoca retraso en el proyecto	Modificar   Eliminar												
2	Que se pierda mucho tiempo en reuniones	Puede suceder que se realicen demasiadas reuniones.	Modificar   Eliminar												
Total de páginas: <b>[1]</b>															

Tabla 7. Descripción del CU Gestionar Riesgos.

<b>Nombre del CU</b>	<b>Realizar análisis.</b>
----------------------	---------------------------

<b>Actores</b>	Analista (inicia)	
<b>Objetivo</b>	Realizar el análisis cualitativo a los riesgos.	
<b>Resumen</b>	El caso de uso se inicia cuando el analista solicita realizar el análisis cualitativo a un riesgo, el sistema muestra un listado con los riesgos que no se les ha realizado dicho análisis, el analista selecciona uno e introduce los valores de probabilidad e impacto al sistema.	
<b>Referencias</b>	<b>R5</b>	
<b>Precondiciones</b>	El analista debe estar previamente autenticado. El listado de riesgos no atendidos debe tener al menos un riesgo.	
<b>Pos condiciones</b>	Queda almacenada la probabilidad y el impacto de al menos un riesgo.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
1. El analista selecciona la opción “Análisis cualitativo” del menú.	2. El sistema muestra el listado de los riesgos a los que no se les ha hecho análisis cualitativo.	
3. El analista selecciona el riesgo al que va a insertarle la probabilidad e impacto.	4. El sistema muestra la interfaz para insertar los datos al riesgo.	
5. El analista selecciona los datos solicitados y selecciona la opción “Aceptar”.	6. El sistema almacena la información en su base de datos. 7. Retomar acción 2.	
<b>Curso alternativo de los eventos.</b>		
5.1 El analista desea cancelar el llenado de los datos, para esto oprime el botón “Cancelar”.	5.2 El sistema cancela la operación de realizar análisis.	
<b>Prioridad</b>	Crítico	



Gestion de riesgos			
No.	Nombre del riesgo	Descripcion	Acción
1	Falta de experiencia del personal	el personal puede que se atrase en la entrega de las tareas ya que tendria que estudiar para la realizacion de las mismas.	Modificar   Eliminar
2	Que los miembros del equipo no cumplan con los plazos de entrega fijados.	Que los miembros del equipo no cumplan con los plazos de entrega fijados.	Modificar   Eliminar
3	Que el equipo de desarrollo no realice debidamente el seguimiento de los riesgos.	Que el equipo de desarrollo no realice debidamente el seguimiento de los riesgos.	Modificar   Eliminar
4	Que existan muchos cambios en la fecha de entrega porque el producto no ha estado terminado.	Que existan muchos cambios en la fecha de entrega porque el producto no ha estado terminado.	Modificar   Eliminar
5	Que las tareas a realizar en el proyecto no se distribuyan equitativamente.	Que las tareas a realizar en el proyecto no se distribuyan equitativamente.	Modificar   Eliminar

Insertar Nuevo Riesgo

Total de páginas: [1]

Tabla 8. Descripción del CU Realizar Análisis.

<b>Nombre del CU</b>	<b>Mostrar riesgos priorizados.</b>	
<b>Actores</b>	Analista (inicia).	
<b>Objetivo</b>	Listar todos los riesgos priorizados por la exposición.	
<b>Resumen</b>	El CU se inicia cuando el analista solicita ver el listado de los riesgos priorizados; en este el sistema le muestra el listado priorizado por la exposición.	
<b>Referencias</b>	<b>R6.</b>	
<b>Precondiciones</b>	El analista debe estar previamente autenticado. El listado de riesgos debe tener al menos un riesgo insertado.	
<b>Pos condiciones</b>	Es mostrado el listado de todos los riesgos priorizados.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	

1. El analista selecciona la opción de “Priorizar riesgos”.	2. El sistema busca en la base de datos.															
	3. El sistema muestra el listado de riesgos priorizados.															
<b>Curso alterno de los eventos.</b>																
	3.1 El sistema no muestra el listado porque no existe ningún riesgo al que se le haya realizado el análisis cualitativo hasta el momento.															
<b>Prioridad</b>	Secundario															
<b>Listado de riesgos priorizados</b>																
<table border="1"> <thead> <tr> <th>No</th> <th>Nombre del riesgo</th> <th>Impacto</th> <th>Probabilidad</th> <th>Exposición</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Que existan problemas con la electricidad</td> <td>0.9</td> <td>0.4</td> <td>0.36</td> </tr> <tr> <td>2</td> <td>Que se pierda mucho tiempo en reuniones</td> <td>0.7</td> <td>0.2</td> <td>0.14</td> </tr> </tbody> </table>		No	Nombre del riesgo	Impacto	Probabilidad	Exposición	1	Que existan problemas con la electricidad	0.9	0.4	0.36	2	Que se pierda mucho tiempo en reuniones	0.7	0.2	0.14
No	Nombre del riesgo	Impacto	Probabilidad	Exposición												
1	Que existan problemas con la electricidad	0.9	0.4	0.36												
2	Que se pierda mucho tiempo en reuniones	0.7	0.2	0.14												
Total de páginas: <b>[1]</b>																

**Tabla 9. Descripción del CU Mostrar riesgos priorizados.**

<b>Nombre del CU</b>	<b>Gestionar Estrategia</b>	
<b>Actores</b>	Planificador de respuesta (inicia).	
<b>Objetivo</b>	Gestionar los datos de las estrategias de los riesgos del proyecto.	
<b>Resumen</b>	El Planificador de respuesta tiene la posibilidad de insertar, modificar o eliminar cada una de las estrategias de los riesgos del proyecto del sistema.	
<b>Referencias</b>	<b>R7.</b>	
<b>Precondiciones</b>	El Planificador de respuesta debe estar previamente autenticado. El listado de riesgos debe tener al menos un riesgo insertado.	
<b>Pos condiciones</b>	El Planificador de respuesta debe haber insertado, eliminado o modificado una o más estrategias.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
1. El Planificador de respuesta selecciona la opción "Gestionar Estrategia" del menú.	2. El sistema muestra una interfaz con la lista de los riesgos	
3. El Planificador de respuesta selecciona la opción "Definir Estrategia" para el riesgo deseado.	4. El sistema muestra una interfaz con la lista de estrategias del riesgo y las diferentes opciones que le permiten al planificador de respuesta:	
4.1 Insertar, ver sección "Insertar Estrategia".		
4.2. Eliminar, ver sección "Eliminar Estrategia".		
4.3. Modificar, ver sección "Modificar Estrategia".		
<b>Sección1. Insertar Estrategia.</b>		
5. El planificador de respuesta toma la opción de insertar estrategia.	6. El sistema muestra la interfaz para agregar la estrategia.	
7. El planificador de respuesta introduce los datos de la nueva estrategia y selecciona la opción "Aceptar".	8. El sistema valida los datos de entrada.	
	9. El sistema almacena la información en su base de datos y se muestra el listado con la nueva estrategia.	

	10. Retornar al paso 4, culminando así el caso de uso.									
<b>Curso alternativo de los eventos</b>										
Acción 8	8.1 El sistema solicita los datos de la nueva estrategia ya que hubo un error a la hora de entrar los datos.									
	8.2 El sistema retoma el paso 4.									
<b>Sección 2. Eliminar Estrategia.</b>										
5. El planificador de respuesta elige la opción "Eliminar Estrategia".	6. El sistema elimina la estrategia seleccionada de la base de datos.									
	7. El sistema muestra el nuevo listado sin la estrategia.									
	8. Retomar el paso 4.									
<b>Sección 3. Modificar Estrategia.</b>										
5. El planificador de respuesta selecciona la opción de "Modificar Estrategia".	6. El sistema muestra la interfaz para modificar la estrategia.									
	7. El sistema modifica los datos seleccionados por el planificador de respuesta.									
	8. El sistema muestra el nuevo listado con los datos de la estrategia ya modificados.									
	9. Retomar paso 4.									
<b>Curso alternativo de los eventos</b>										
5.1 El planificador de respuesta selecciona la opción de cancelar	5.1.2 El sistema cancela la operación de modificar la estrategia.									
	5.1.3 Retomar paso 4.									
<b>Prioridad</b>	Crítico									
<p><b>Gestion de estrategias</b></p> <table border="1"> <thead> <tr> <th>No.</th> <th>Descripción</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Previa preparacion del personal</td> <td>Modificar   Eliminar</td> </tr> <tr> <td>2</td> <td>capacitacion sobre la marcha</td> <td>Modificar   Eliminar</td> </tr> </tbody> </table> <p style="text-align: right;">Insertar Nueva Estrategia</p>		No.	Descripción	Acción	1	Previa preparacion del personal	Modificar   Eliminar	2	capacitacion sobre la marcha	Modificar   Eliminar
No.	Descripción	Acción								
1	Previa preparacion del personal	Modificar   Eliminar								
2	capacitacion sobre la marcha	Modificar   Eliminar								

Tabla 10. Descripción del CU Gestionar Estrategia.

<b>Nombre del CU</b>	<b>Gestionar Respuesta</b>
<b>Actores</b>	Planificador de respuesta (inicia).
<b>Objetivo</b>	Gestionar los datos de las respuestas a las estrategias de los riesgos del proyecto.
<b>Resumen</b>	El Planificador de respuesta tiene la posibilidad de insertar,

	modificar o eliminar cada una de las respuestas a las estrategias de los riesgos del proyecto del sistema.
<b>Referencias</b>	<b>R8.</b>
<b>Precondiciones</b>	El Planificador de respuesta debe estar previamente autenticado. El listado de riesgos debe tener al menos un riesgo insertado. Debe haber al menos una estrategia en el listado.
<b>Pos condiciones</b>	El Planificador de respuesta debe haber insertado, eliminado o modificado una o más respuestas.
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El Planificador de respuesta selecciona la opción "Gestionar Respuesta" del menú.	2. El sistema muestra una interfaz con la lista de los riesgos
3. El Planificador de respuesta selecciona la opción "Definir Respuesta" para el riesgo deseado.	4. El sistema muestra una interfaz con la lista de las estrategias del riesgo.
5. El Planificador de respuesta selecciona la opción "Definir Respuesta" para la estrategia deseada	6. El sistema muestra una interfaz con la lista de las respuestas de la estrategia del riesgo y las diferentes opciones que le permiten al planificador de respuesta:
4.1 Insertar, ver sección "Insertar Respuesta".	
4.2. Eliminar, ver sección "Eliminar Respuesta".	
4.3. Modificar, ver sección "Modificar Respuesta".	
<b>Sección1. Insertar Respuesta.</b>	
5. El planificador de respuesta toma la opción de insertar respuesta.	6. El sistema muestra la interfaz para agregar la respuesta.
7. El planificador de respuesta introduce los datos de la nueva respuesta y selecciona la opción "Aceptar".	8. El sistema valida los datos de entrada.
	9. El sistema almacena la información en su base de datos y se muestra el listado con la nueva respuesta.
	10. Retornar al paso 4, culminando así el caso de uso.
<b>Curso alternativo de los eventos</b>	
Acción 8	8.1 El sistema solicita los datos de la nueva respuesta ya que hubo un error a la hora de entrar

	los datos.						
	8.2 El sistema retoma el paso 4.						
<b>Sección 2. Eliminar Respuesta.</b>							
5. El planificador de respuesta elige la opción "Eliminar Respuesta".	6. El sistema elimina la respuesta seleccionada de la base de datos.						
	7. El sistema muestra el nuevo listado sin la respuesta.						
	8. Retomar el paso 4.						
<b>Sección 3. Modificar Respuesta.</b>							
5. El planificador de respuesta selecciona la opción de "Modificar Respuesta".	6. El sistema muestra la interfaz para modificar la respuesta.						
	7. El sistema modifica los datos seleccionados por el planificador de respuesta.						
	8. El sistema muestra el nuevo listado con los datos de la respuesta ya modificados.						
	9. Retomar paso 4.						
<b>Curso alternativo de los eventos</b>							
5.1 El planificador de respuesta selecciona la opción de cancelar	5.1.2 El sistema cancela la operación de modificar la respuesta.						
	5.1.3 Retomar paso 4.						
<b>Prioridad</b>	Crítico						
<p><b>Gestion de respuestas</b></p> <table border="1"> <thead> <tr> <th>No.</th> <th>Descripción</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>preparar al personal con profes bien capacitados y con experiencia en los temas requeridos</td> <td>Modificar   Eliminar</td> </tr> </tbody> </table> <p style="text-align: right;">Insertar Nueva Respuesta</p>		No.	Descripción	Acción	1	preparar al personal con profes bien capacitados y con experiencia en los temas requeridos	Modificar   Eliminar
No.	Descripción	Acción					
1	preparar al personal con profes bien capacitados y con experiencia en los temas requeridos	Modificar   Eliminar					

Tabla 11. Descripción del CU Gestionar Respuesta.

<b>Nombre del CU</b>	<b>Calcular métricas.</b>
<b>Actores</b>	Seguimiento y Control (inicia)
<b>Objetivo</b>	Realizar el cálculo de métricas a los proyectos.
<b>Resumen</b>	El caso de uso se inicia cuando el encargado del seguimiento y control solicita realizar el cálculo de métricas, el sistema muestra una interfaz para entrar datos al sistema.
<b>Referencias</b>	<b>R9</b>
<b>Precondiciones</b>	El encargado del seguimiento y control debe estar previamente autenticado.

<b>Pos condiciones</b>	Queda almacenada el resultado de al menos una métrica.					
<b>Curso Normal de los Eventos</b>						
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>					
1. El encargado del seguimiento y control selecciona la opción "Calcular Métricas" del menú.	2. El sistema muestra el listado de las métricas.					
3. El encargado del seguimiento y control selecciona la métrica que desea calcular	4. El sistema muestra la interfaz para insertar los datos de la métrica.					
5. El encargado del seguimiento y control entra los datos solicitados y selecciona la opción "Aceptar".	6. El sistema calcula el resultado de la métrica. 7. El sistema almacena el resultado en su base de datos. 8. Retomar acción 2.					
<b>Curso alternativo de los eventos.</b>						
5.1 El encargado del seguimiento y control desea cancelar el llenado de los datos, para esto oprime el botón "Cancelar".	5.2 El sistema cancela la operación de realizar el cálculo de la métrica.					
<b>Prioridad</b>	Secundario					
<p>Seleccione la métrica a calcular:</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Nombre</th> </tr> </thead> <tbody> <tr> <td>Métrica de Estimación de Costos (MEC).</td> </tr> <tr> <td>Métricas de Precisión de Fuentes (MPF)</td> </tr> <tr> <td>Métricas de Idoneidad de las Estrategias (MIE)</td> </tr> </tbody> </table>			Nombre	Métrica de Estimación de Costos (MEC).	Métricas de Precisión de Fuentes (MPF)	Métricas de Idoneidad de las Estrategias (MIE)
Nombre						
Métrica de Estimación de Costos (MEC).						
Métricas de Precisión de Fuentes (MPF)						
Métricas de Idoneidad de las Estrategias (MIE)						

Tabla 12. Descripción del CU Calcular métricas.

<b>Nombre del CU</b>	<b>Documentar Experiencias.</b>
<b>Actores</b>	Documentador (inicia)
<b>Objetivo</b>	Documentar las experiencias personales.
<b>Resumen</b>	El caso de uso se inicia cuando el documentador solicita documentar experiencias, el sistema muestra una interfaz para recoger los datos, se entran los datos y se guardan en el sistema.
<b>Referencias</b>	<b>R11.1</b>
<b>Precondiciones</b>	El documentador debe estar previamente autenticado.
<b>Pos condiciones</b>	Queda guardada al menos una experiencia en el sistema.
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>


1. El documentador selecciona la opción “Documentar Experiencias” del menú.	2. El sistema muestra la interfaz para la entrada de datos de la experiencia.
3. El documentador entra los datos de la experiencia y selecciona la opción “Aceptar”.	4. El sistema almacena la información en su base de datos. 5. Retomar acción 2.
<b>Curso alternativo de los eventos.</b>	
3.1 El encargado del seguimiento y control desea cancelar el llenado de los datos, para esto oprime el botón “Cancelar”.	3.2 El sistema cancela la operación de actualizar el riesgo.
<b>Prioridad</b>	Secundario
	

Tabla 13. Documentar Experiencias.

<b>Nombre del CU</b>	<b>Mostrar Experiencias.</b>
<b>Actores</b>	Documentador (inicia)
<b>Objetivo</b>	Mostrar las experiencias personales.
<b>Resumen</b>	El caso de uso se inicia cuando el documentador solicita mostrar experiencias, el sistema da la opción de seleccionar un rol específico, luego muestra un listado con las experiencias de ese rol.
<b>Referencias</b>	<b>R11.2</b>
<b>Precondiciones</b>	El documentador debe estar previamente autenticado.
<b>Pos condiciones</b>	Es mostrado el listado de todas las experiencias.
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El documentador selecciona la opción “Mostrar Experiencias” del menú.	2. El sistema muestra la opción de seleccionar un rol específico.



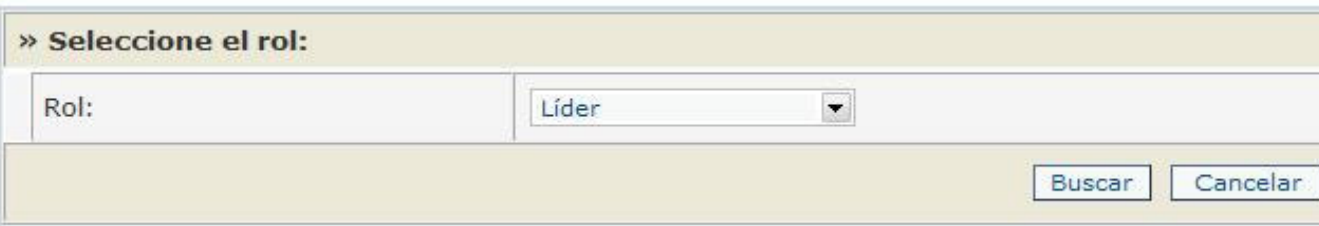
3. El documentador selecciona el rol y la opción "Aceptar".	4. El sistema busca la información en su base de datos. 5. El sistema muestra el listado de las experiencias
<b>Curso alternativo de los eventos.</b>	
3.1 El documentador desea cancelar la selección del rol, para esto oprime el botón "Cancelar".	3.2 El sistema cancela la operación de mostrar la experiencia.
<b>Prioridad</b>	Secundario
	

Tabla 14. Descripción del CU Mostrar Experiencias.

<b>Nombre del CU</b>	<b>Mostrar Registro de Riesgos.</b>	
<b>Actores</b>	Documentador (inicia)	
<b>Objetivo</b>	Mostrar el Registro de Riesgos.	
<b>Resumen</b>	El caso de uso se inicia cuando el documentador solicita Mostrar Registro de Riesgos, el sistema muestra un listado de todos los riesgos y da la opción de seleccionar uno en específico, luego muestra el registro completo de dicho riesgo.	
<b>Referencias</b>	<b>R10.3</b>	
<b>Precondiciones</b>	El documentador debe estar previamente autenticado.	
<b>Pos condiciones</b>	Es mostrado el registro del riesgo seleccionado.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
1. El documentador selecciona la opción "Mostrar Registro de Riesgos" del menú.	2. El sistema muestra un listado de todos los riesgos y da la opción de seleccionar uno en específico.	
3. El documentador selecciona el riesgo y la opción "Ver".	4. El sistema busca la información en su base de datos. 5. El sistema muestra el registro de dicho riesgo	
<b>Curso alternativo de los eventos.</b>		
3.1 El documentador desea regresar al listado de	3.2 El sistema regresa al listado de riesgos.	

riesgos, para esto oprime la opción "Volver".	
<b>Prioridad</b>	Secundario
<b>Registro de riesgos</b>	
» RIESGO:	
<b>Nombre del riesgo:</b>	Que existan problemas con la electricidad
<b>Descripción:</b>	Provoca retraso en el proyecto
<b>Fecha:</b>	2002-04-25
<b>Clasificación:</b>	Tecnológico
<b>Frecuencia:</b>	2
<b>Causas o fuentes :</b>	aaa
<b>Impacto:</b>	0.4
<b>Probabilidad:</b>	0.9
	[ Volver a listado]

**Tabla 15. Descripción del CU Mostrar Registro de Riesgos.**

**Conclusiones:**

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

## Capítulo 4. Análisis y Diseño del Sistema.

### 4.1 Introducción.

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

### 4.2 Modelo del Análisis.

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

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

#### 4.2.1 Diagramas de clases de análisis.

##### 4.2.1.1 Documentar Experiencias.

##### CU Documentar experiencias.

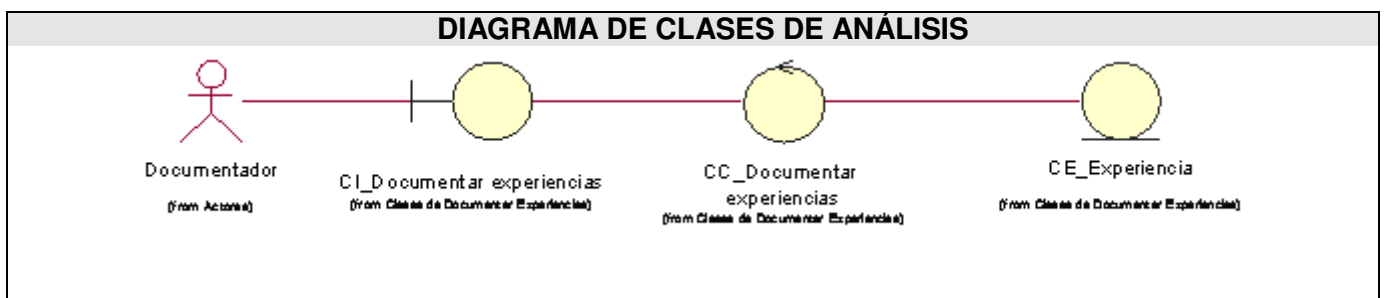


Figura 13. Diagrama de Clases del Análisis: CU Documentar Experiencias.

**CU Mostrar Experiencias.**

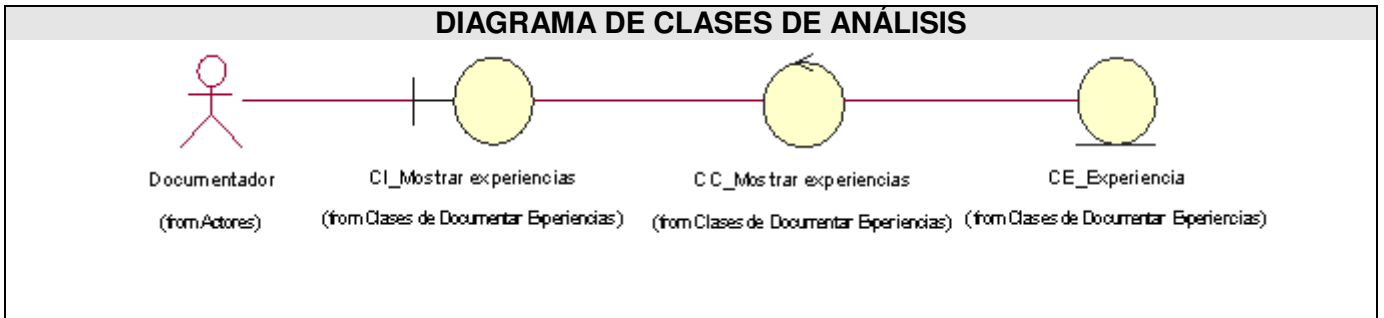


Figura 14. Diagrama de Clases del Análisis: CU Mostrar Experiencias.

**CU Mostrar Registros de Riesgos.**

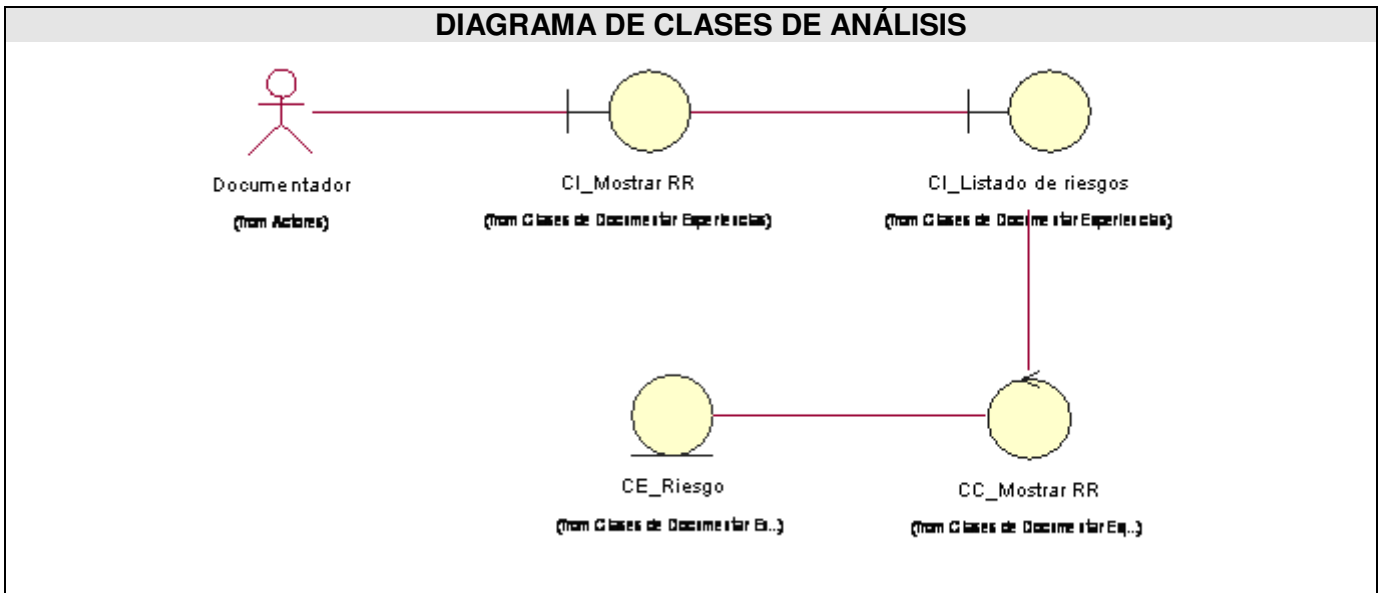


Figura 15. Diagrama de Clases del Análisis: CU Mostrar Registros de Riesgos.

4.2.1.2 Gestión de Riesgos.  
 CU Calcular Métricas.

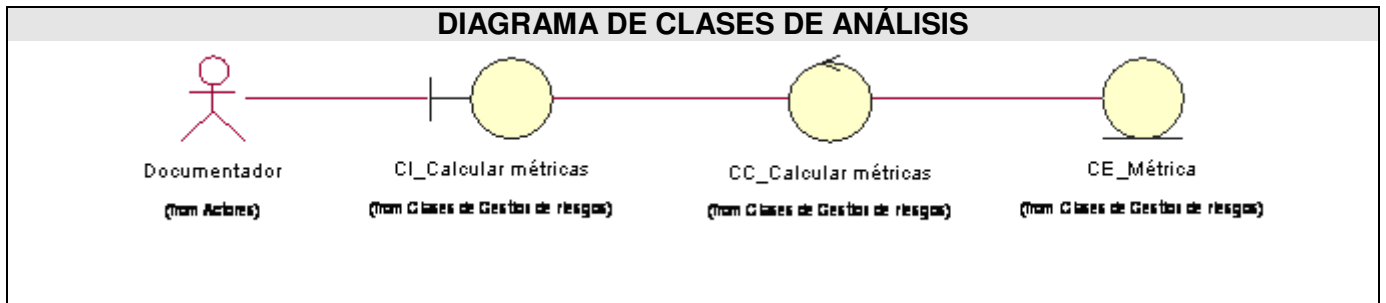


Figura 16. Diagrama de Clases del Análisis: CU Calcular Métricas.

**CU Gestionar Equipo.**

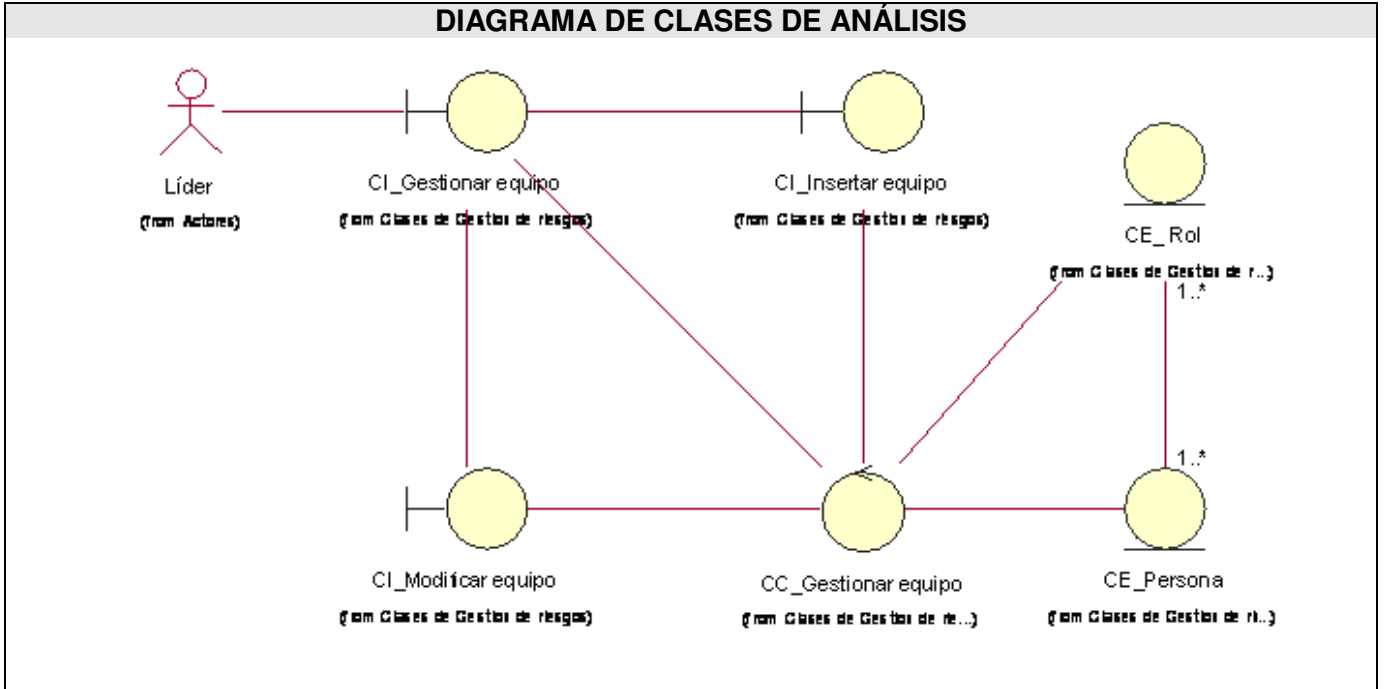


Figura 17. Diagrama de Clases del Análisis: CU Gestionar Equipo.

**CU Gestionar Estrategias.**

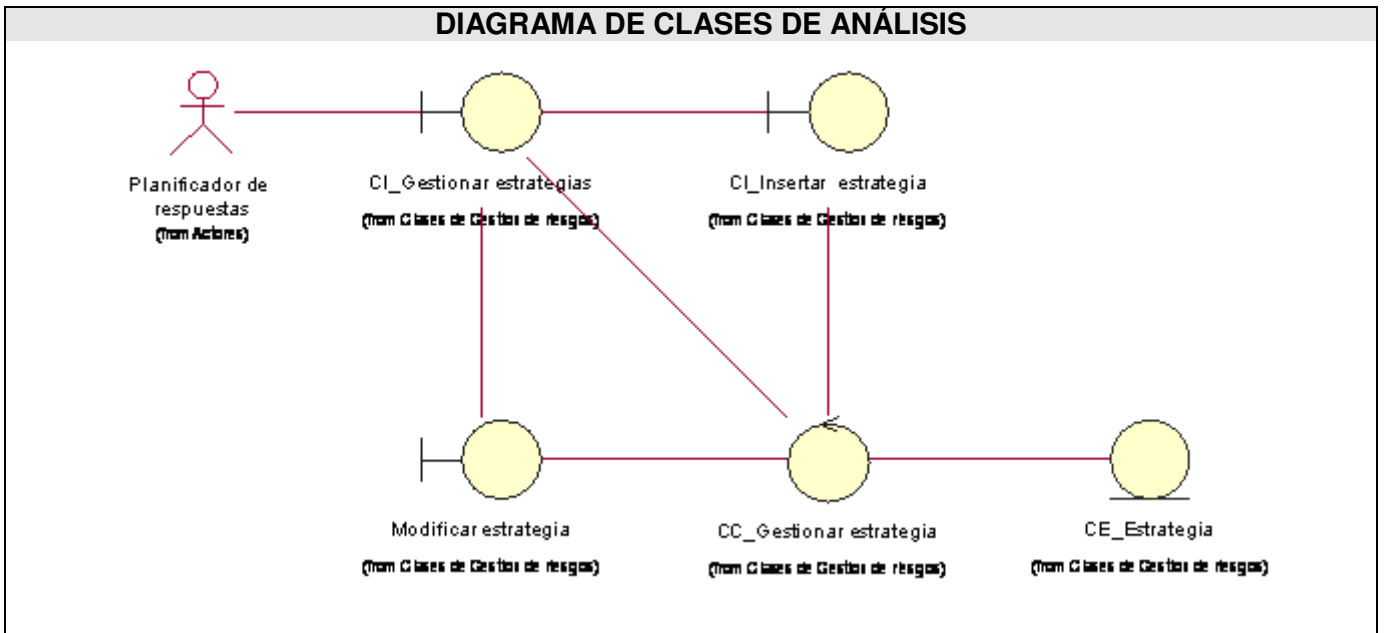


Figura 18. Diagrama de Clases del Análisis: CU Gestionar Estrategias.

CU Gestionar Respuestas.

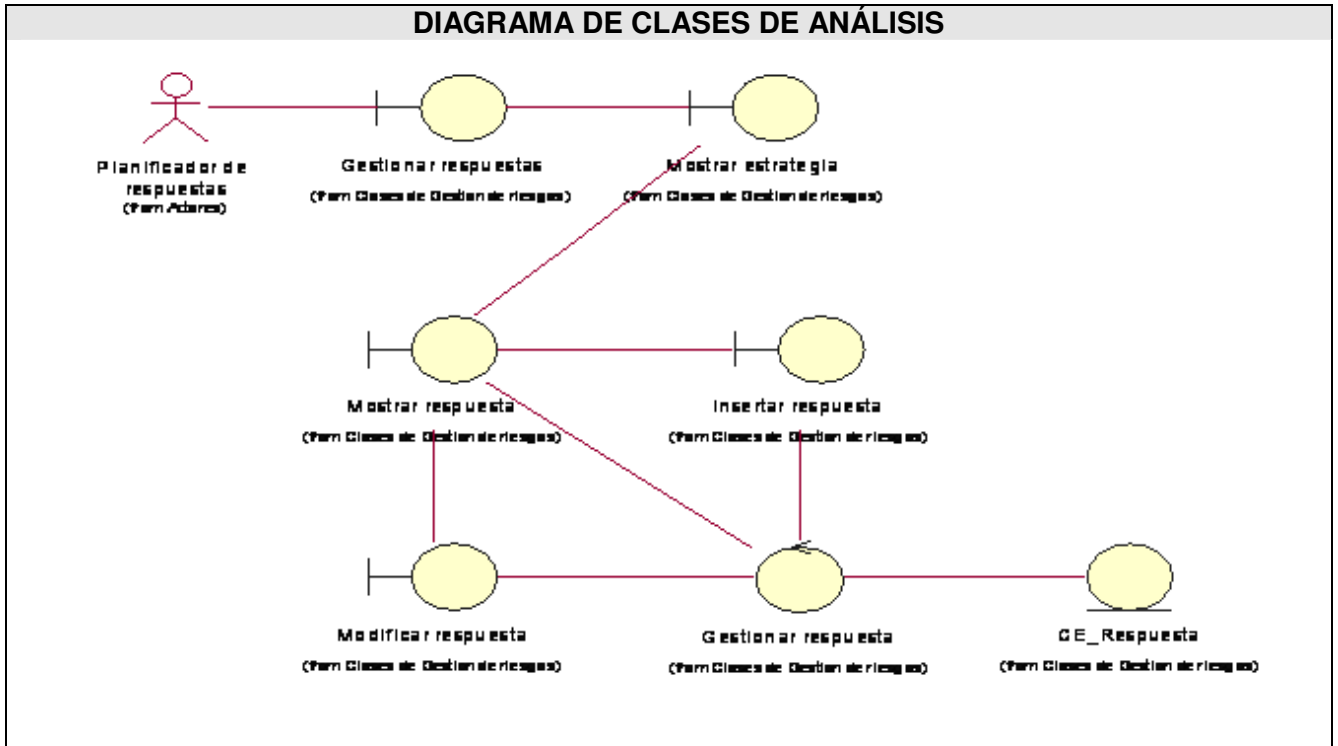


Figura 19. Diagrama de Clases del Análisis: CU Gestionar Respuestas.

**CU Gestionar Riesgos.**

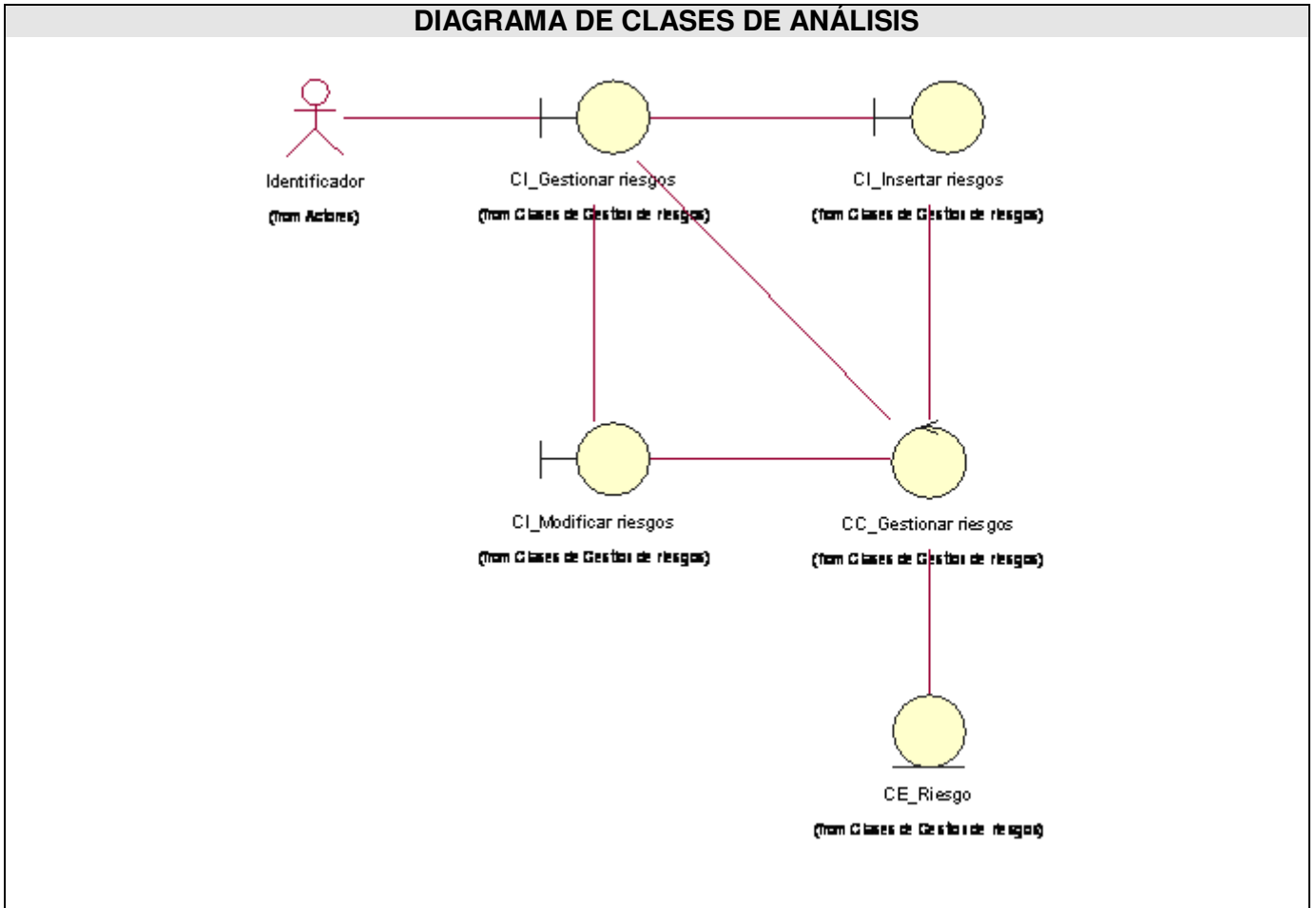


Figura 20. Diagrama de Clases del Análisis: CU Gestionar Riesgos

**CU Mostrar Riesgos Priorizados.**

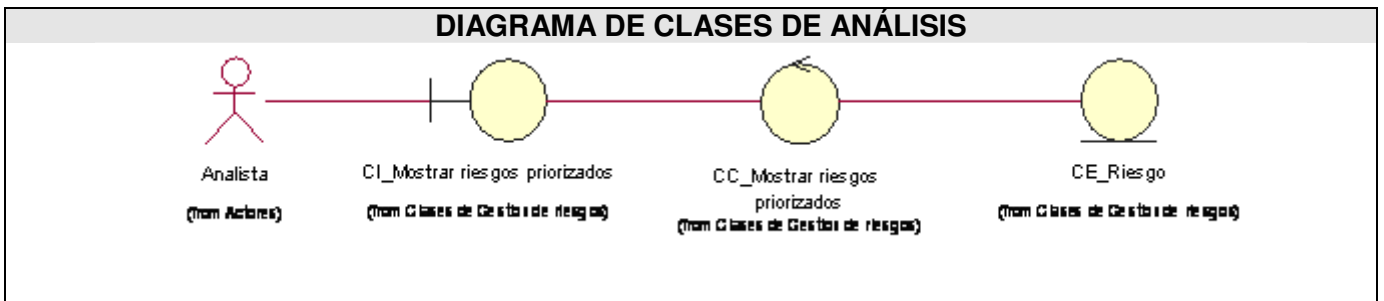


Figura 21. Diagrama de Clases del Análisis: CU Mostrar Riesgos Priorizados.



**CU Realizar análisis.**

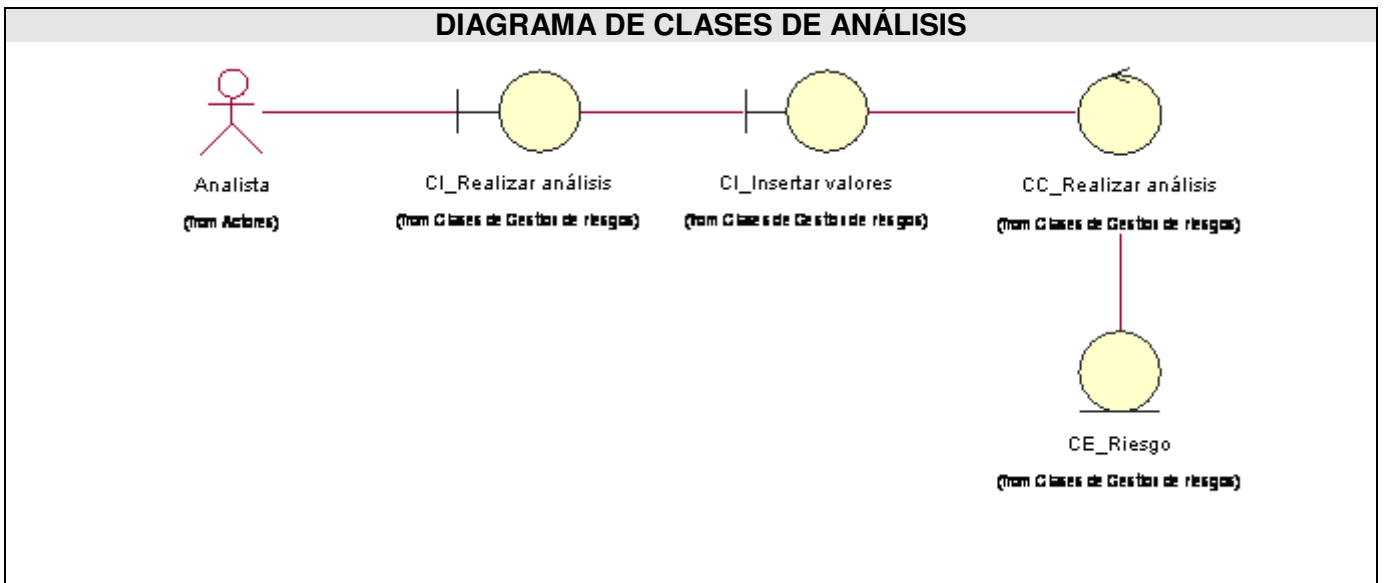


Figura 22. Diagrama de Clases del Análisis: CU Realizar análisis.

**4.2.1.3 Seguridad.**

**CU Autenticarse.**

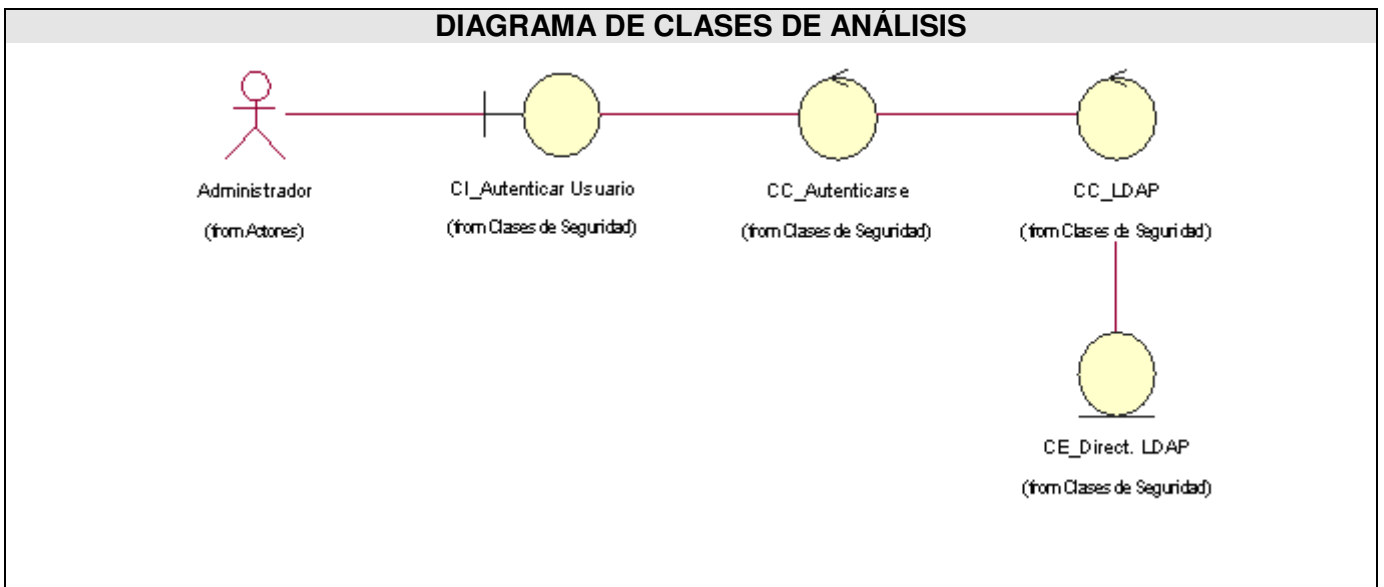
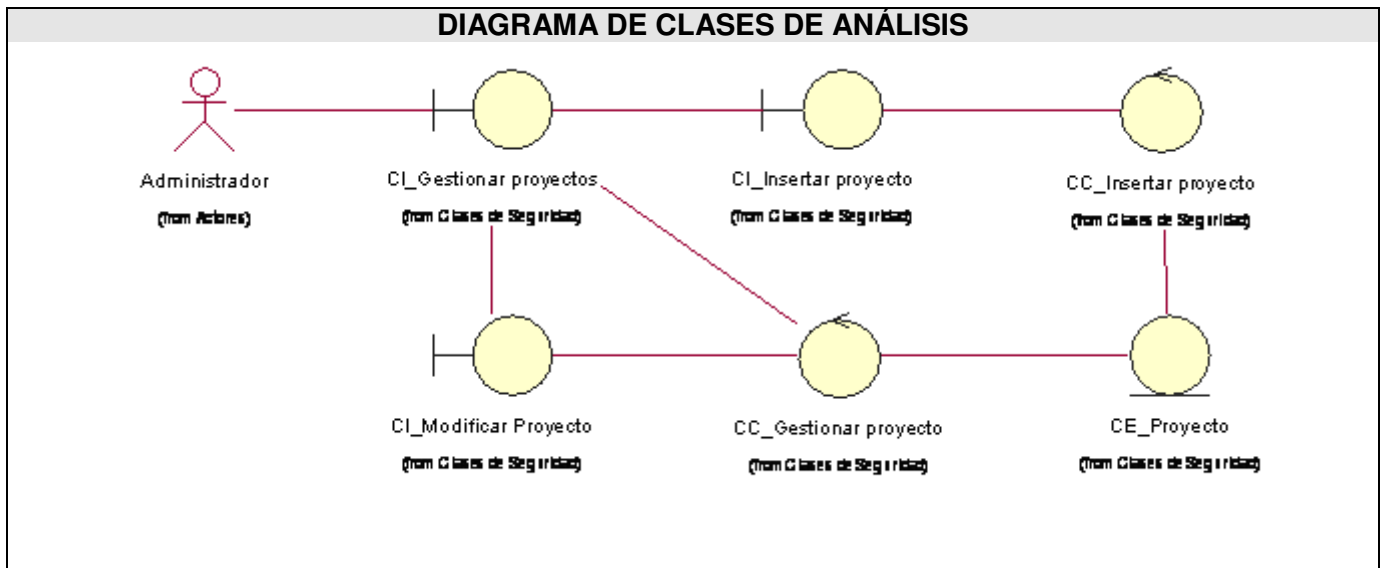


Figura 23. Diagrama de Clases del Análisis: CU Autenticarse.

**CU Gestionar proyecto.**



**Figura 24. Diagrama de Clases del Análisis: CU Gestionar proyectos.**

4.3 Modelo del Diseño.

4.3.1 Diagramas de clases y de interacción del diseño.

4.3.1.1 Documentar Experiencias.

CU Documentar Experiencias.

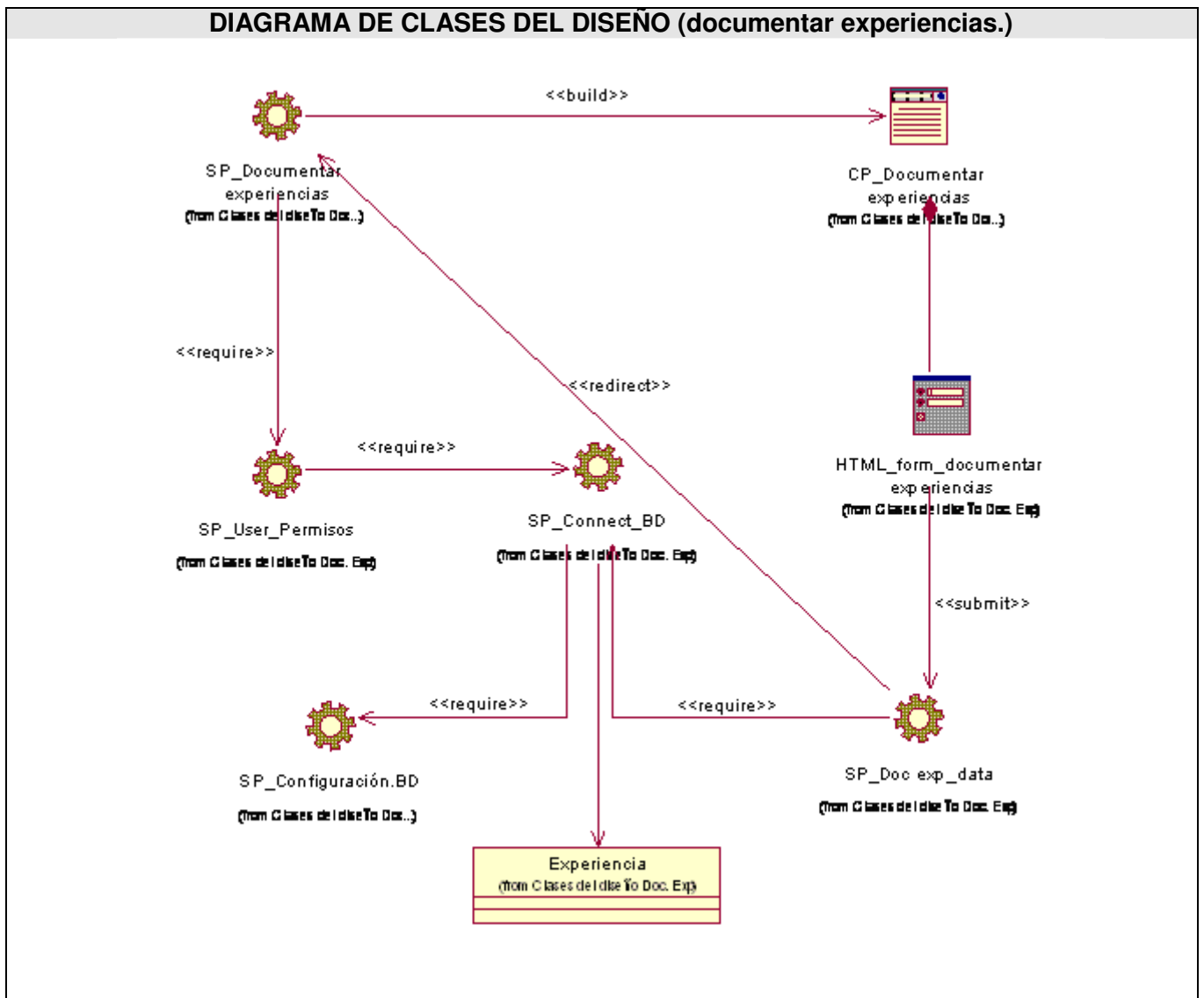


Figura 25. Diagrama de Clases del Diseño: CU Documentar Experiencias.

CU Mostrar Experiencias.

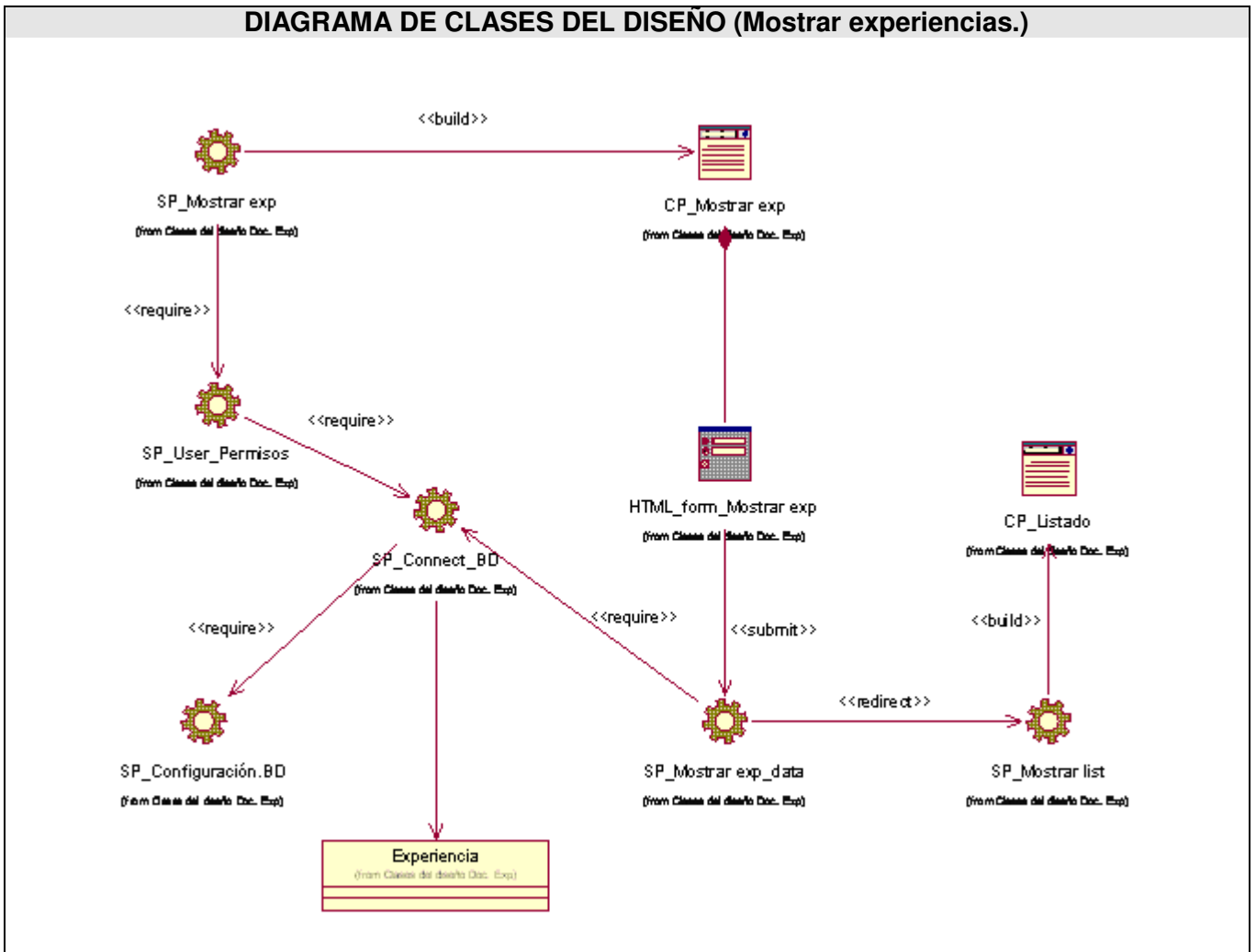


Figura 26. Diagrama de Clases del Diseño: CU Mostrar Experiencias.

**CU Mostrar Registro de Riesgos.**

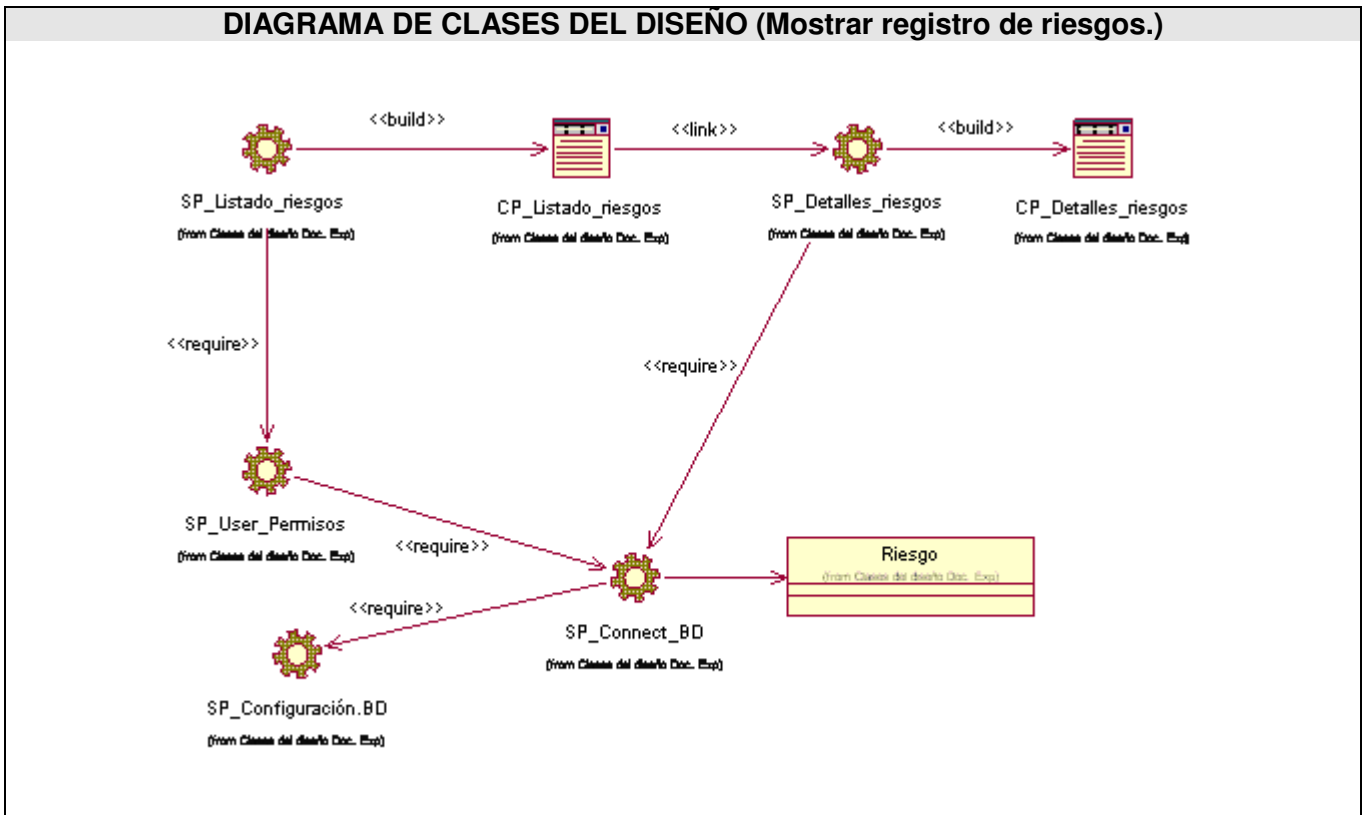


Figura 27. Diagrama de Clases del Diseño: CU Mostrar Registro de Riesgos.

4.3.1.2 Paquete Gestión de Riesgos.

CU Calcular Métricas.

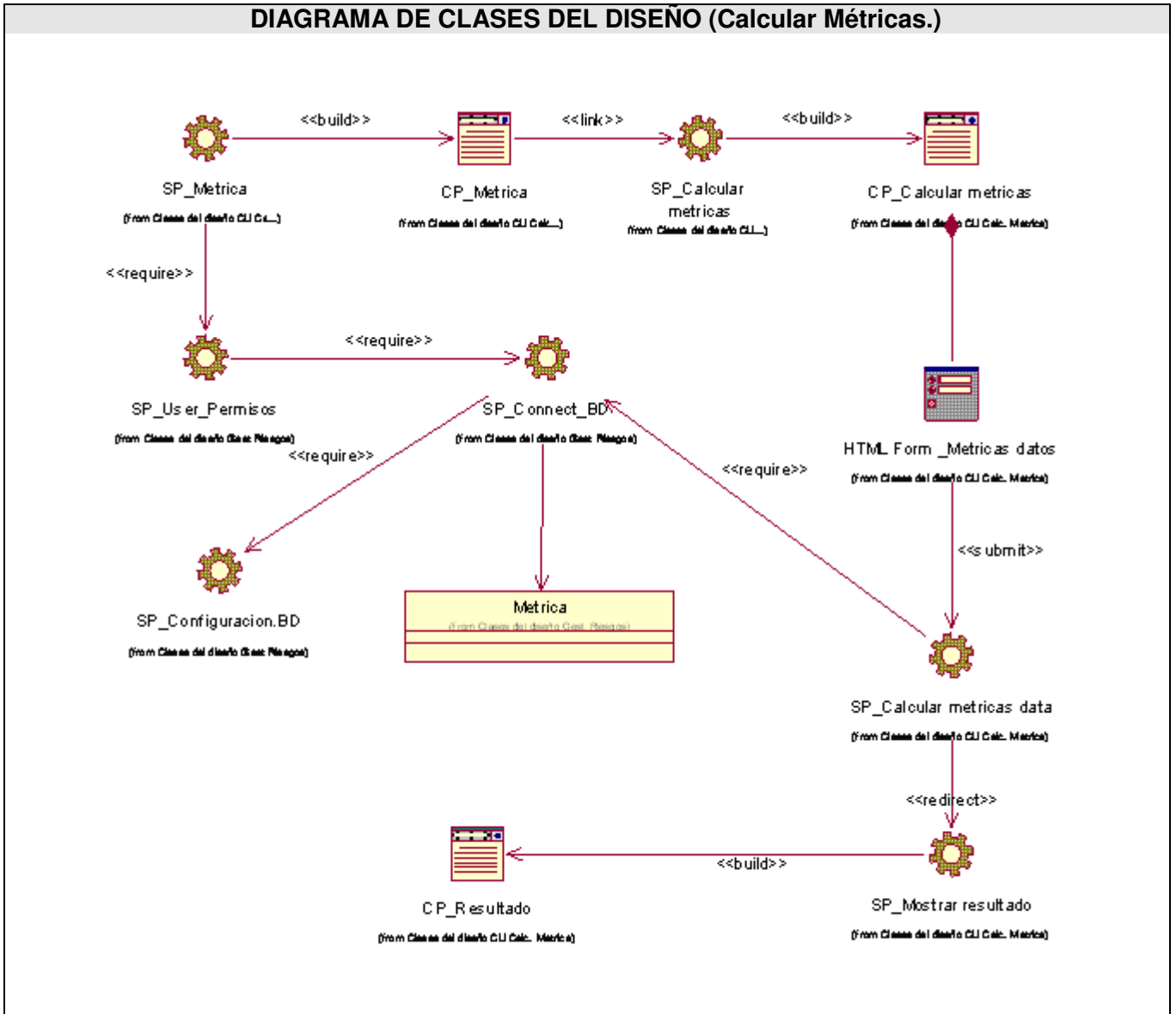


Figura 28. Diagrama de Clases del Diseño: CU Calcular Métricas.

CU Gestionar Equipo

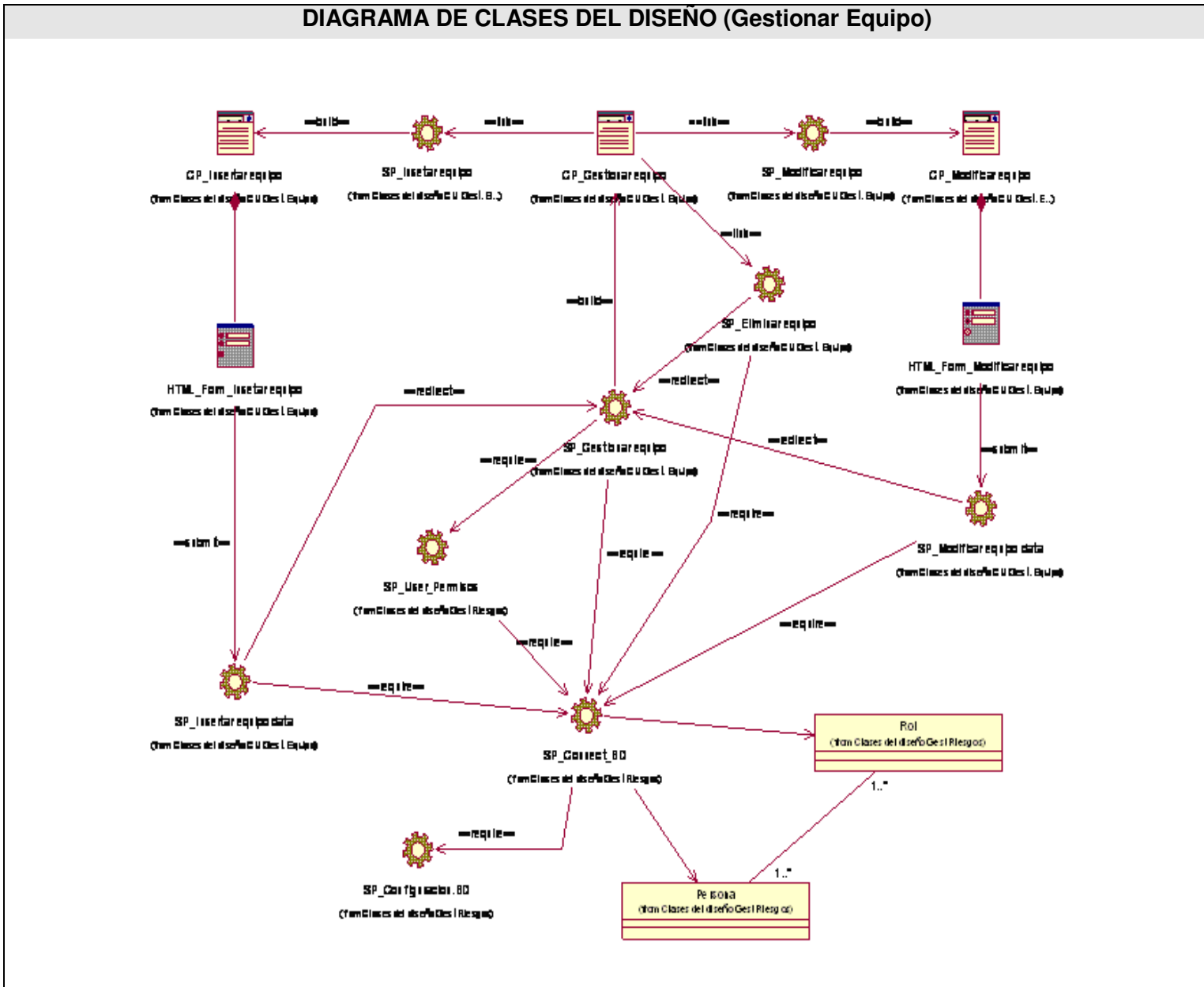


Figura 29. Diagrama de Clases del Diseño: CU Gestionar Equipo.

CU Gestionar Estrategia.

DIAGRAMA DE CLASES DEL DISEÑO (Gestionar Estrategias.)

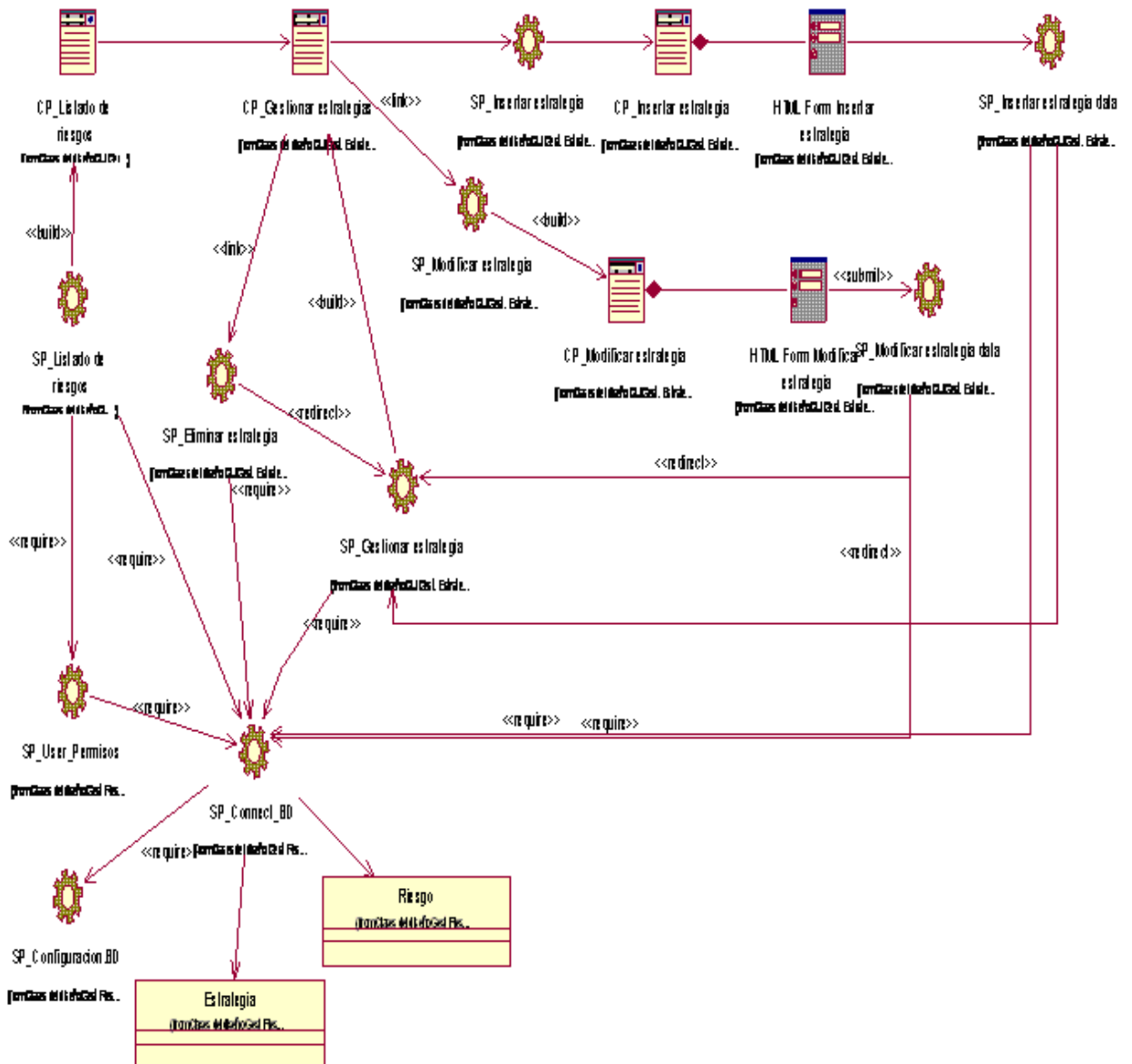


Figura 30. Diagrama de Clases del Diseño: CU Gestionar Estrategias.



CU Gestionar Respuestas.

DIAGRAMA DE CLASES DEL DISEÑO (Gestionar Respuestas.)

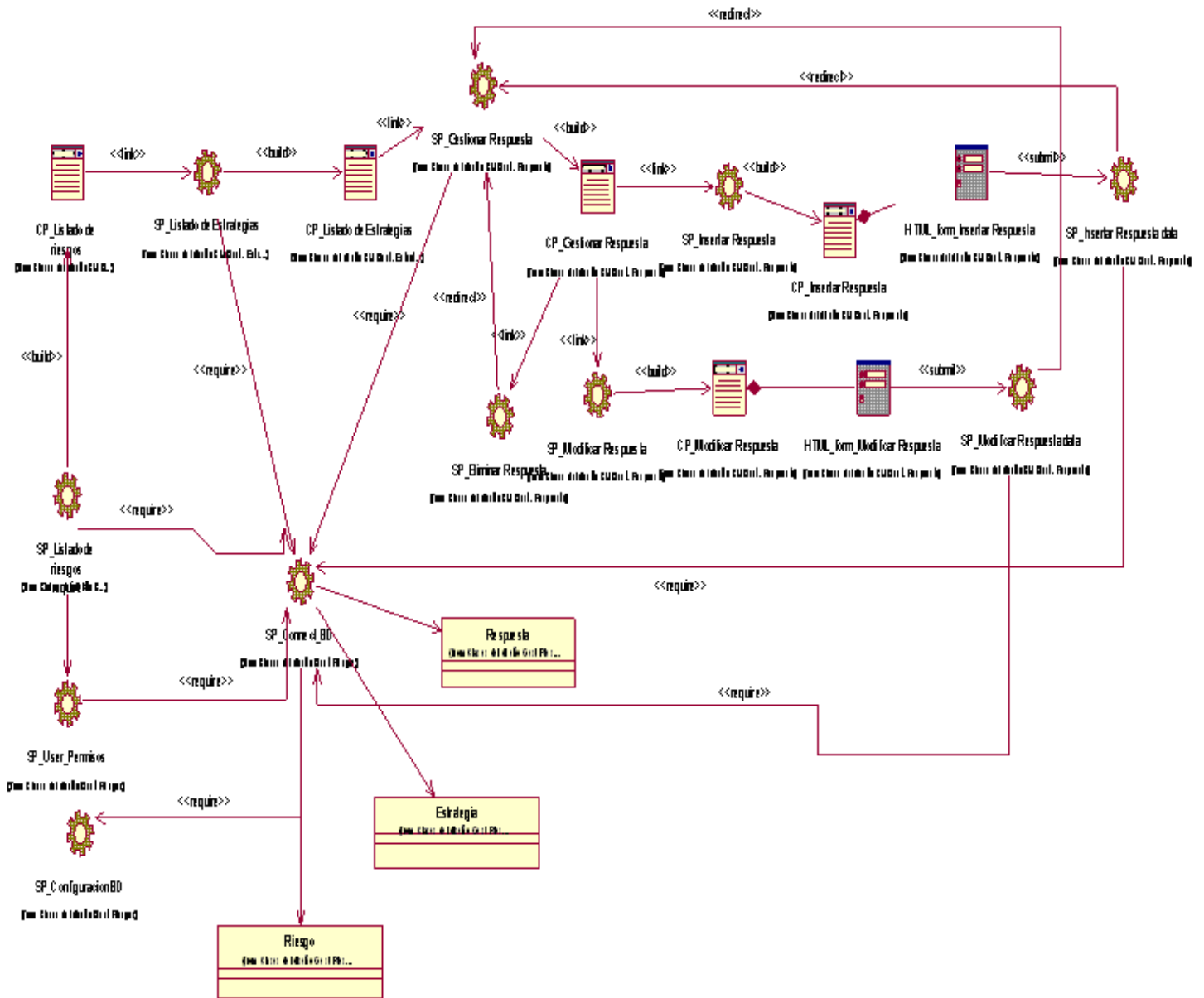


Figura 31. Diagrama de Clases del Diseño: CU Gestionar Respuestas.

CU Gestionar Riesgos.

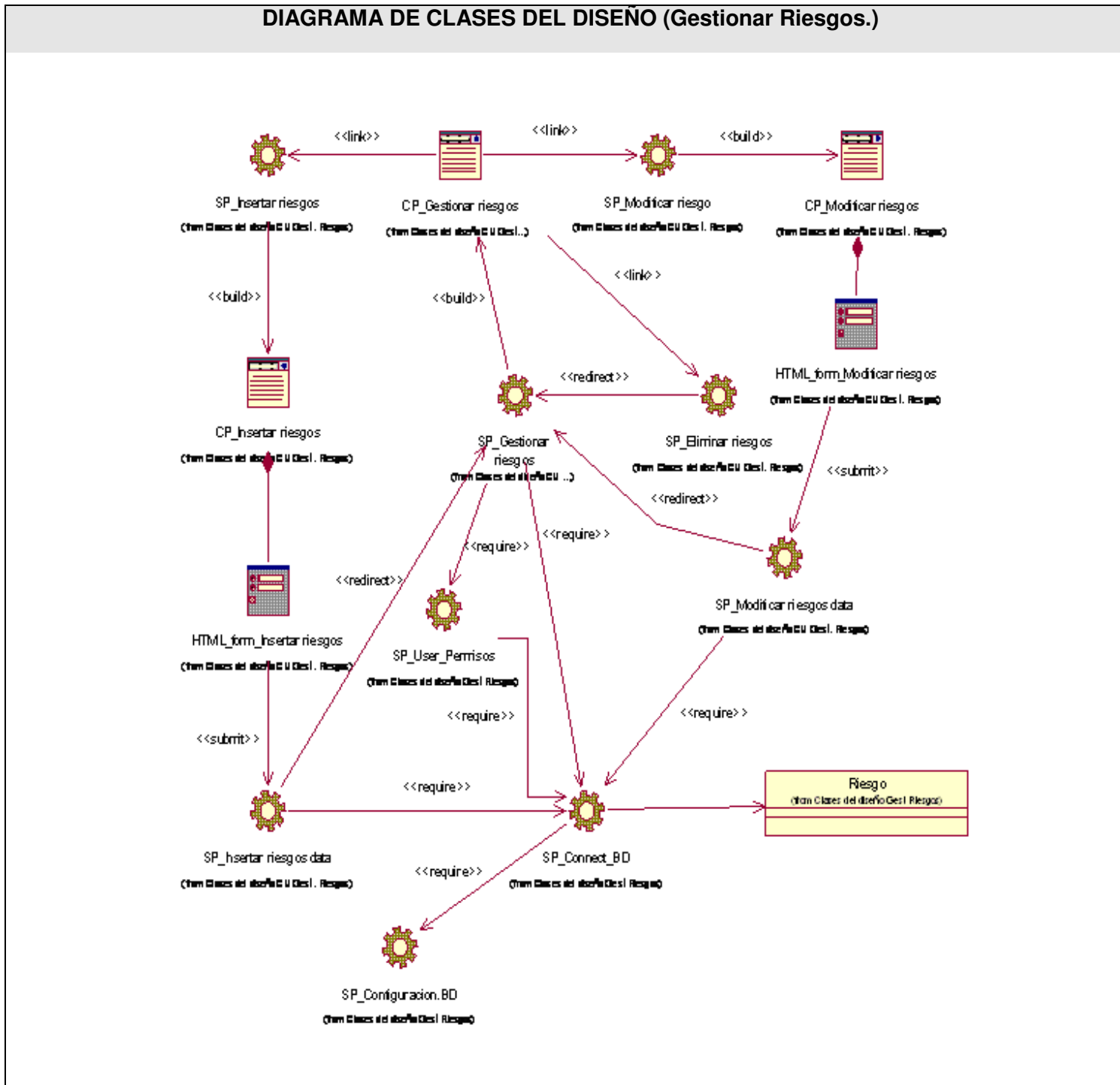


Figura 32. Diagrama de Clases del Diseño: CU Gestionar Riesgos.

CU Mostrar Riesgos Priorizados.

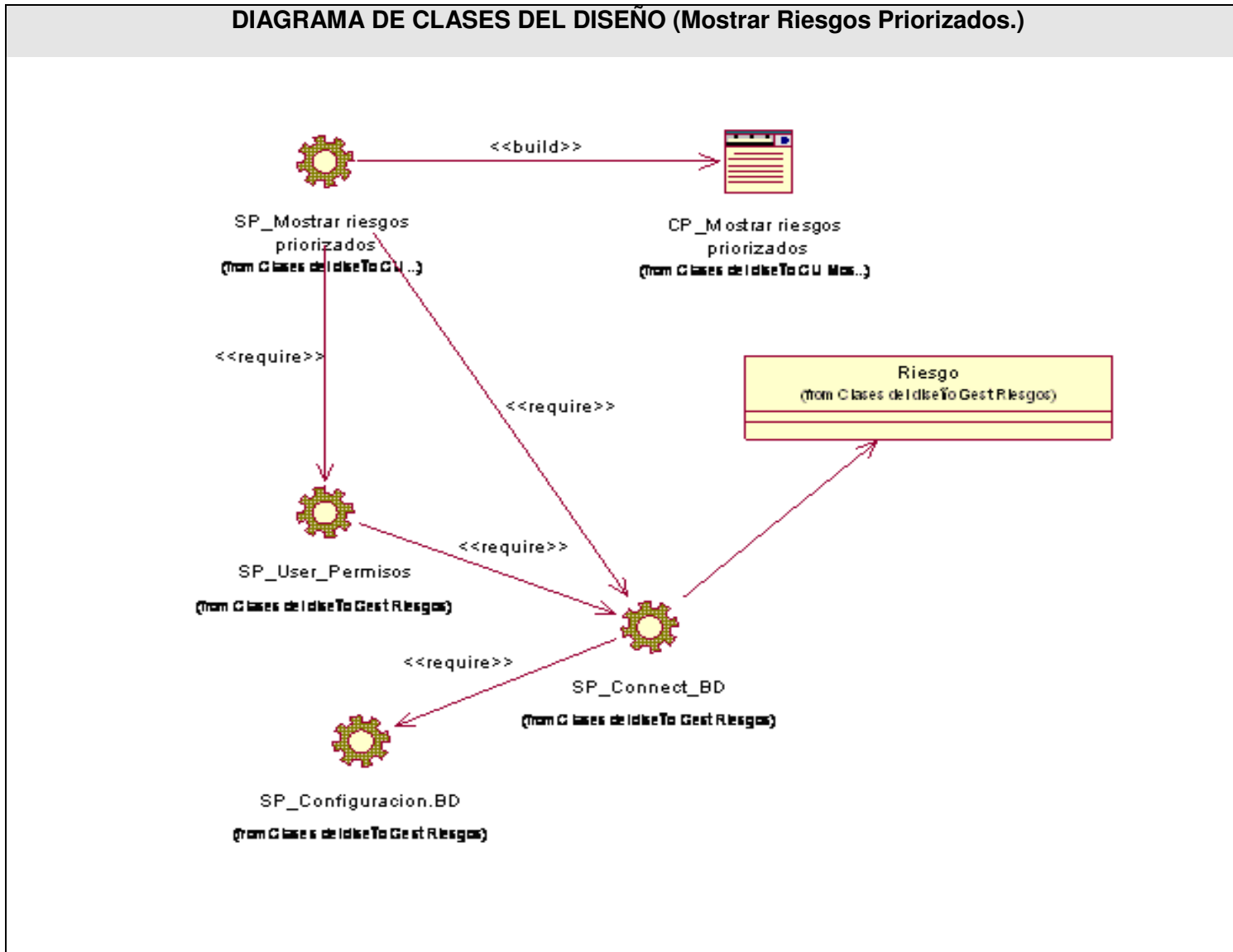


Figura 33. Diagrama de Clases del Diseño: CU Mostrar Riesgos Priorizados.

CU Realizar Análisis.

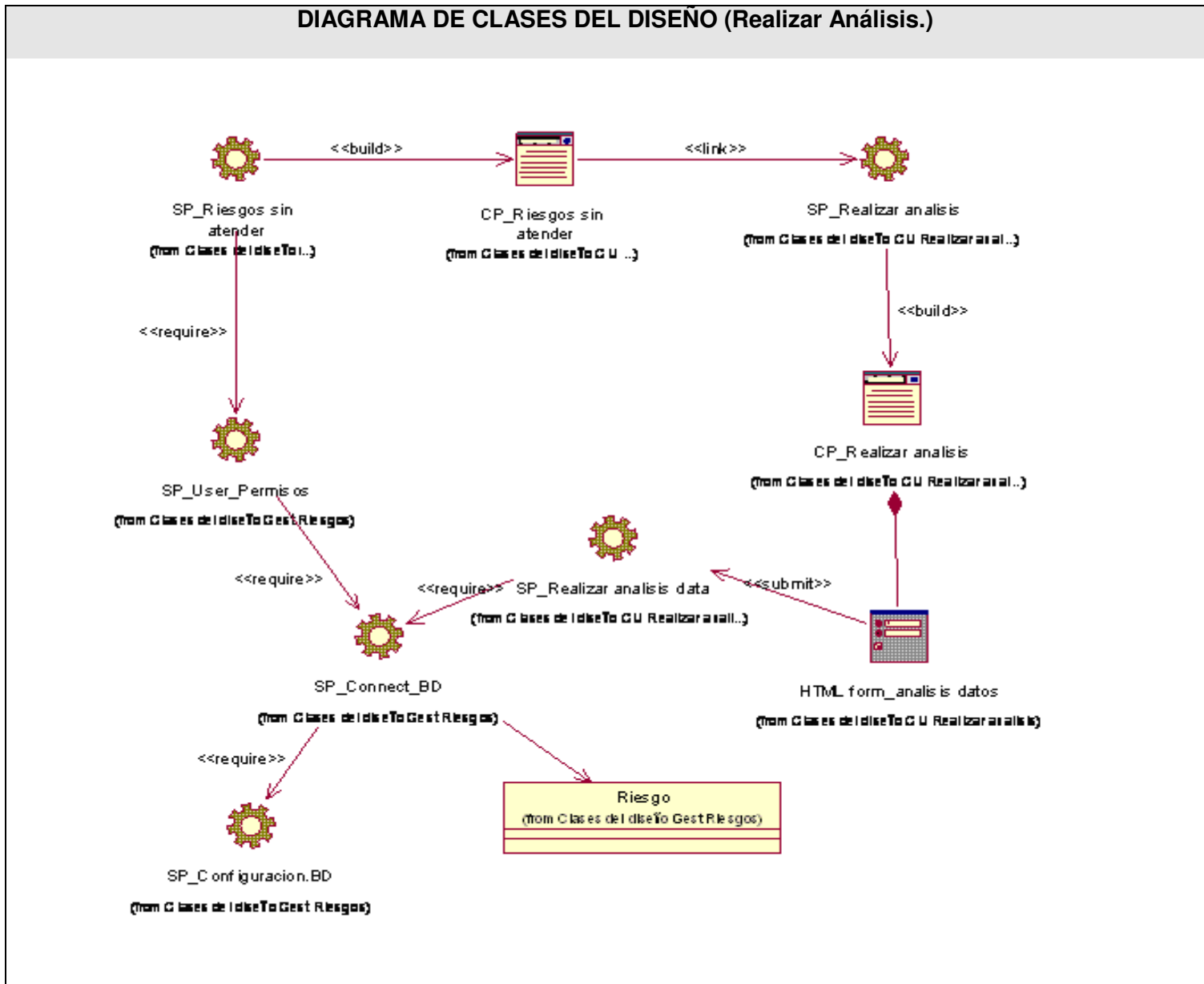


Figura 34. Diagrama de Clases del Diseño: CU Realizar Análisis.

### 4.3.1.3 Seguridad.

#### CU Autenticarse.

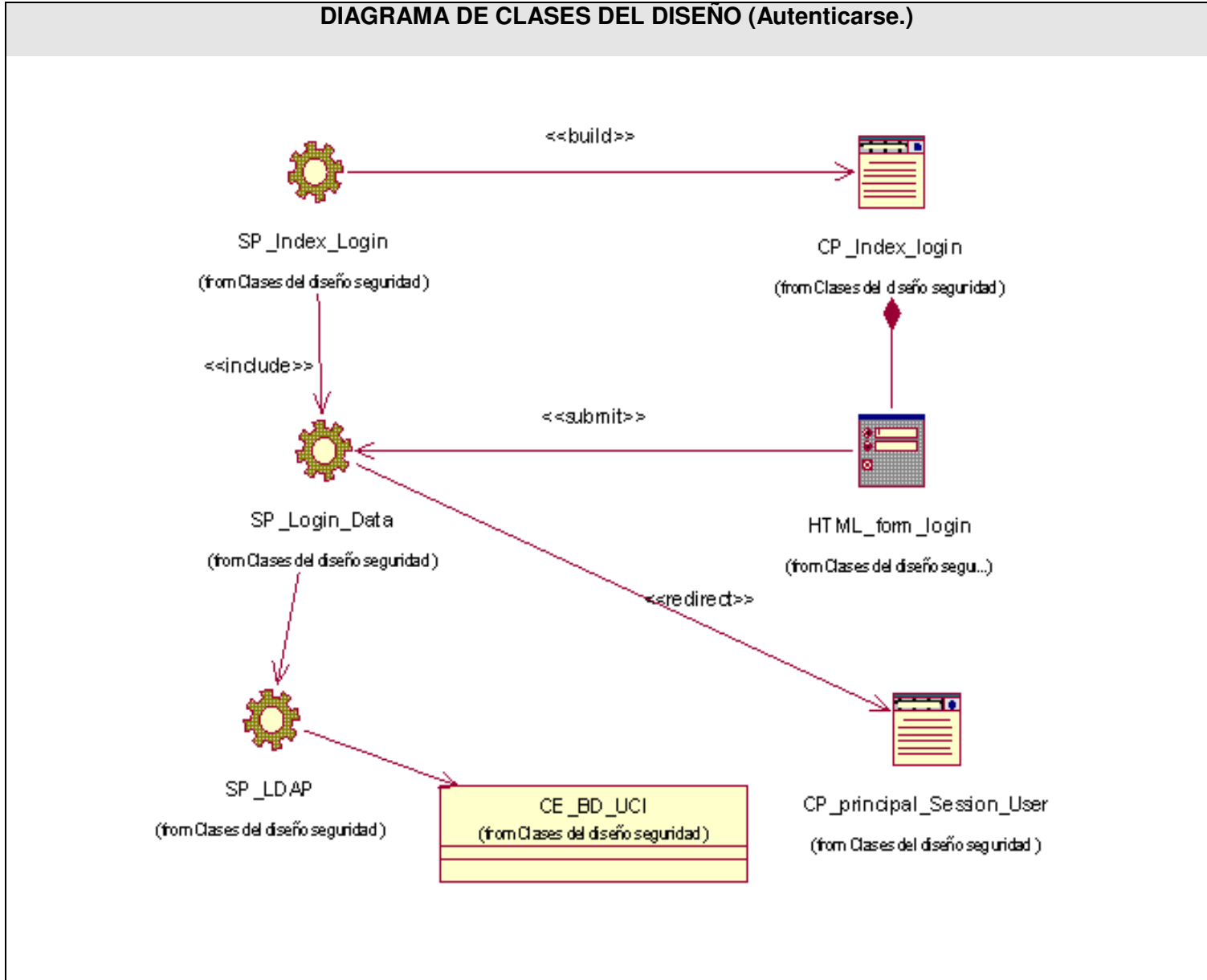


Figura 35. Diagrama de Clases del Diseño: CU Autenticarse.

CU Gestionar Proyectos.

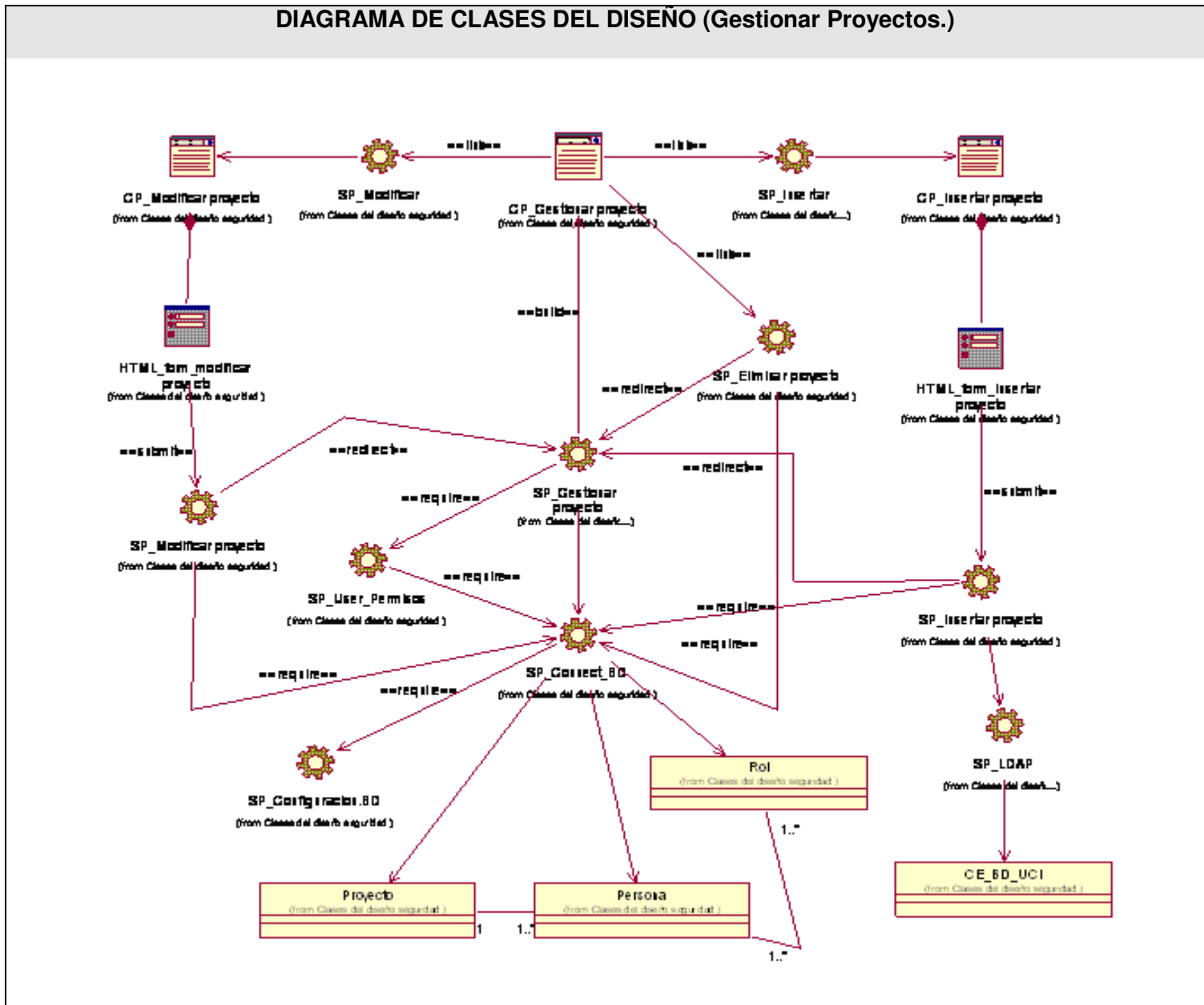


Figura 36. Diagrama de Clases del Diseño: CU Gestionar Proyectos.

#### **4.4 Definiciones de diseño.**

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

##### **4.4.1 Tratamiento de errores.**

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

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

##### **4.4.2 Seguridad**

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

##### **4.4.3 Interfaz**

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

Si la Interfaz de Usuario está bien diseñada, el usuario encontrará la respuesta que espera a su acción. Debe permitirse libertad al usuario para que elija el modo de interacción que más se adecúe a sus objetivos en cada momento. La mayoría de los programas y sistemas operativos ofrecen varias formas de interacción al usuario.

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

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



4.5 Diseño de la Base de Datos.

4.5.1 Modelo Lógico de Datos.

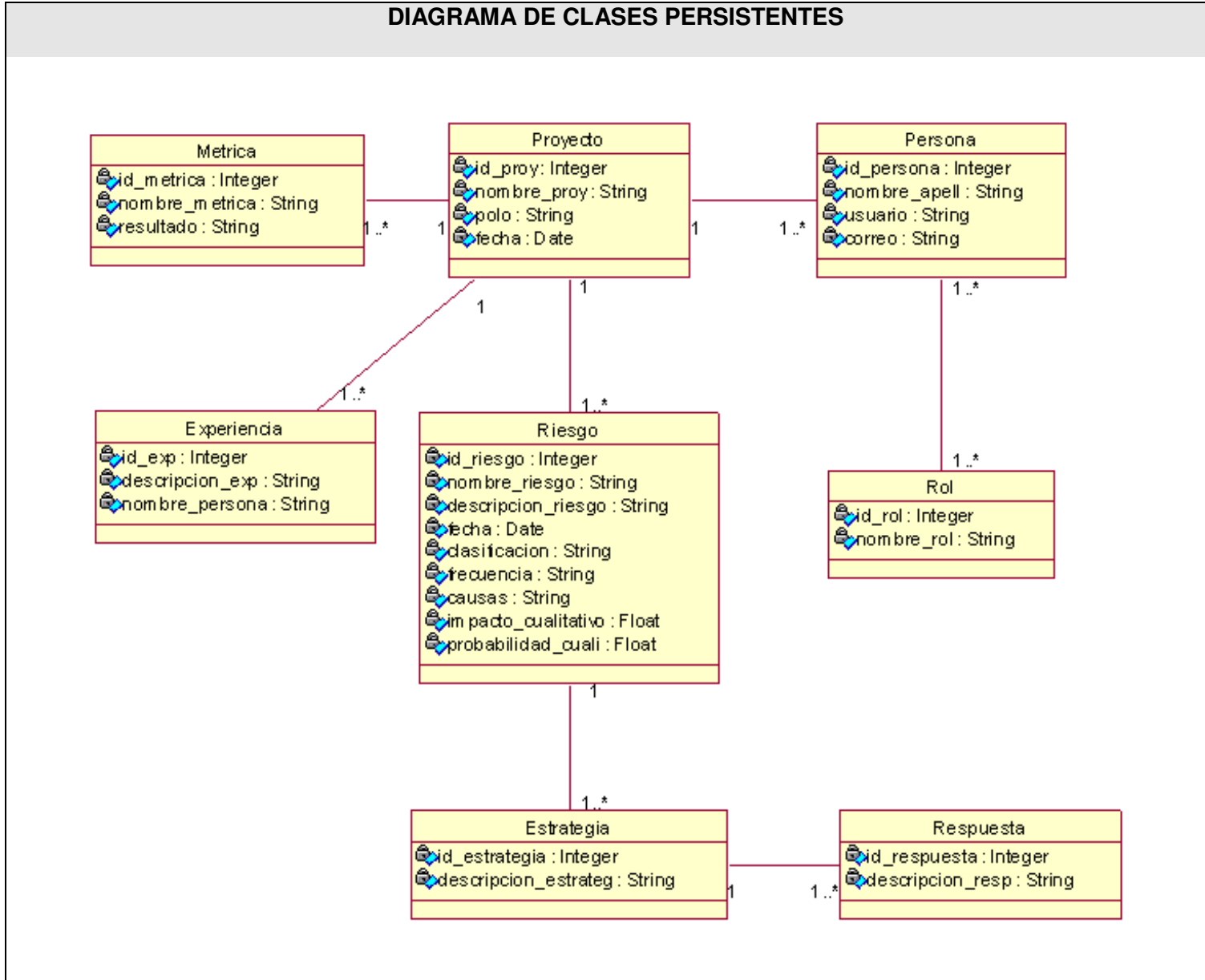


Figura 37. Diagrama de Clases Persistentes

Modelo Físico de Datos.

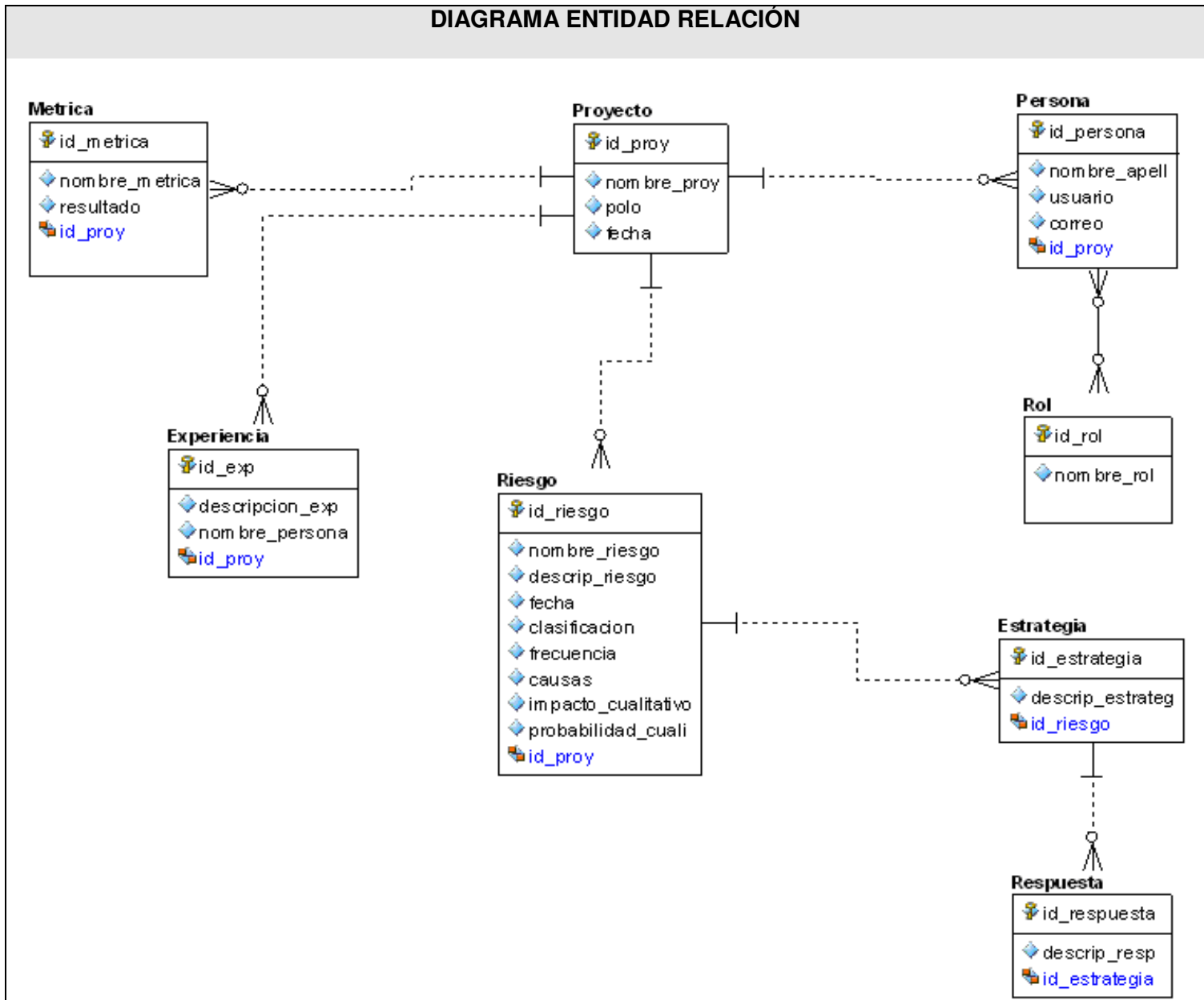


Figura 38. Modelo Físico de Datos.

MODELO DE DATOS

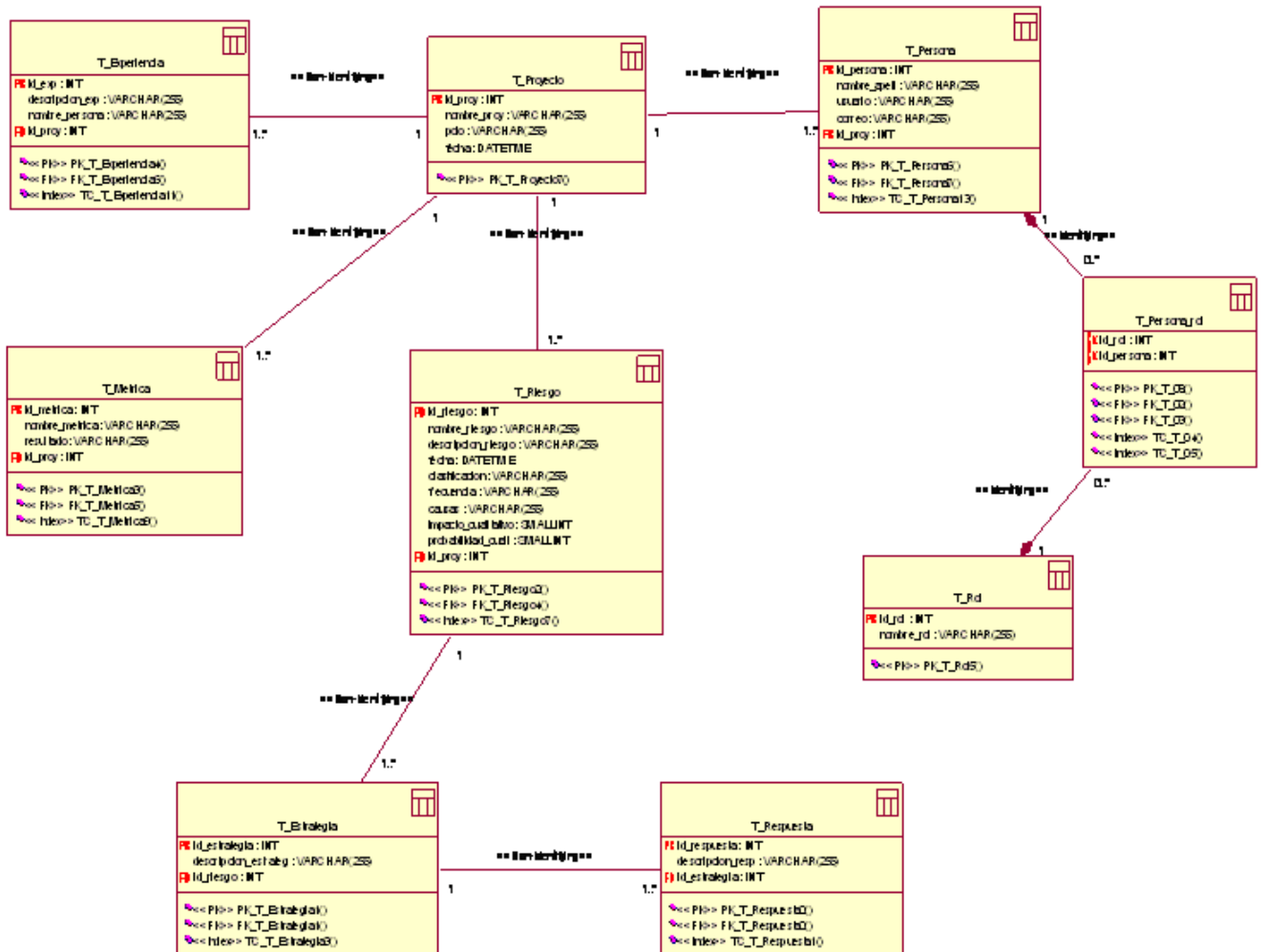


Figura 39. Modelo de Datos.

4.6 Diagrama de Despliegue.

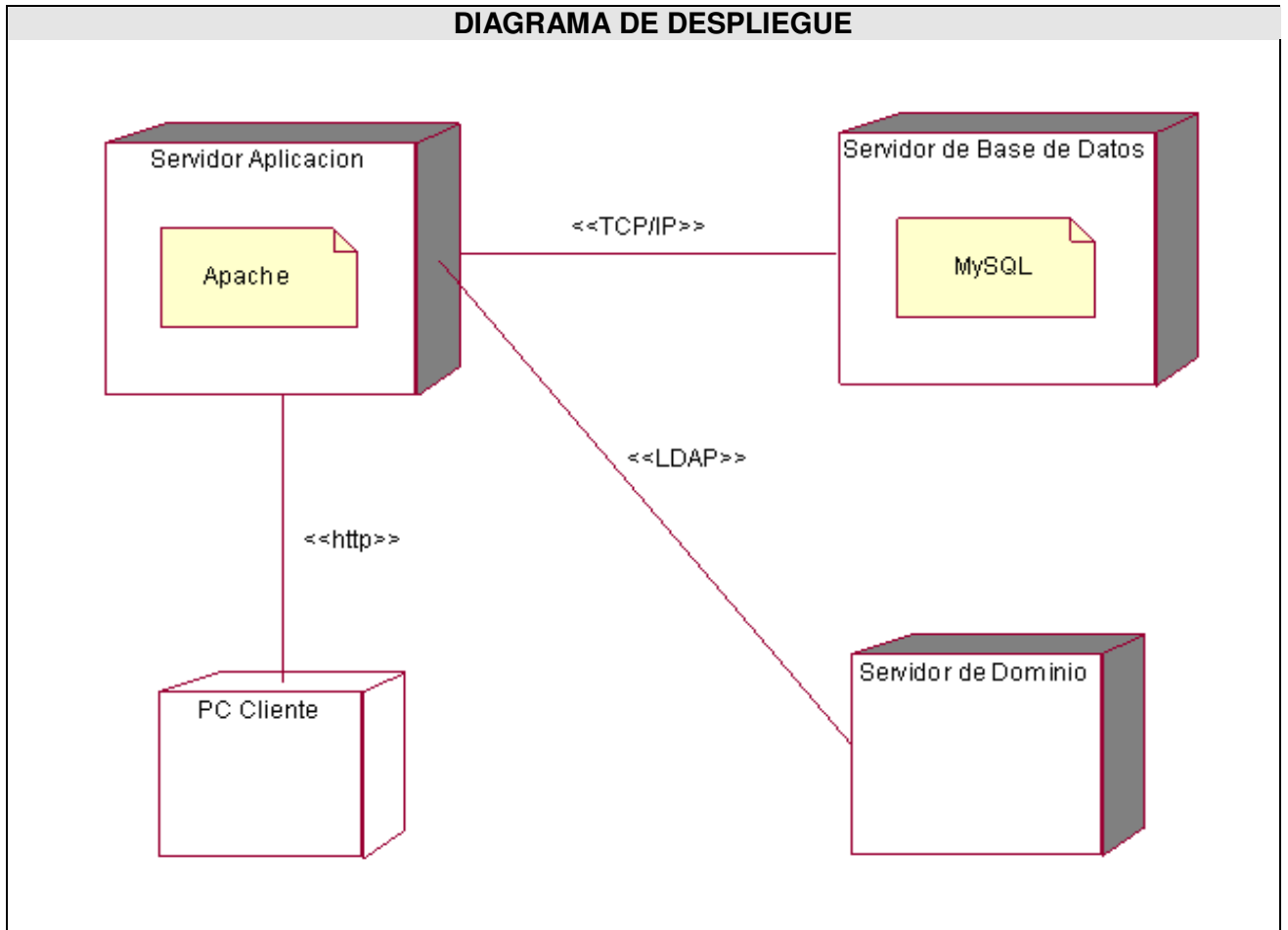


Figura 40. Diagrama de Despliegue.

#### 4.7 Modelo de Implementación.

##### 4.7.1 Diagramas de Componentes.

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

##### 4.7.1.1 Diagrama de componentes Base de Datos.

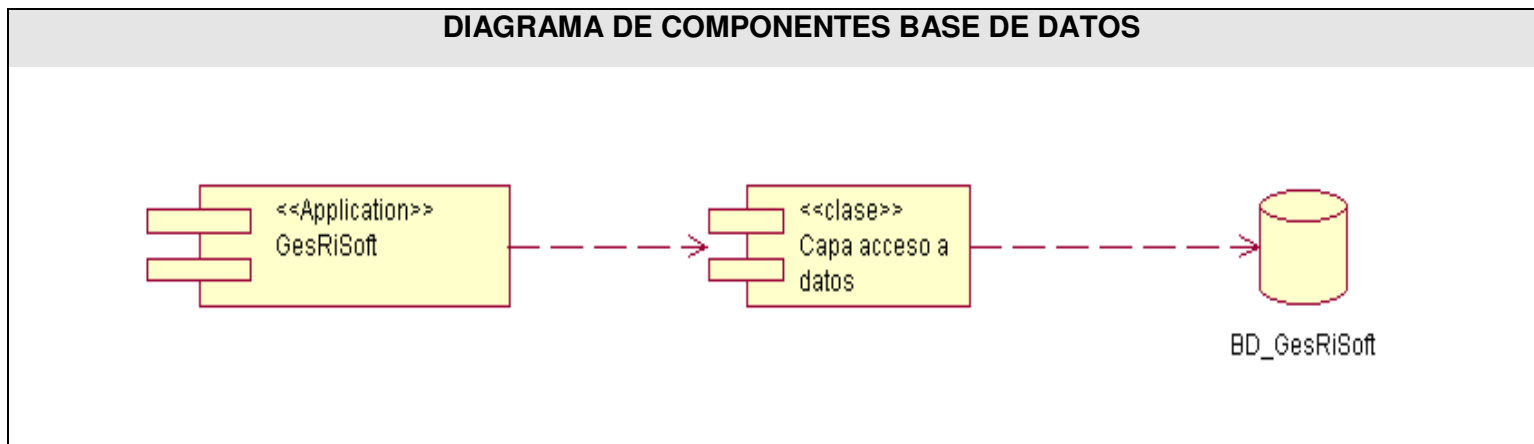


Figura 41. Diagrama de Componentes de Base de Datos

4.7.1.2 Diagrama de componentes Código Fuente.

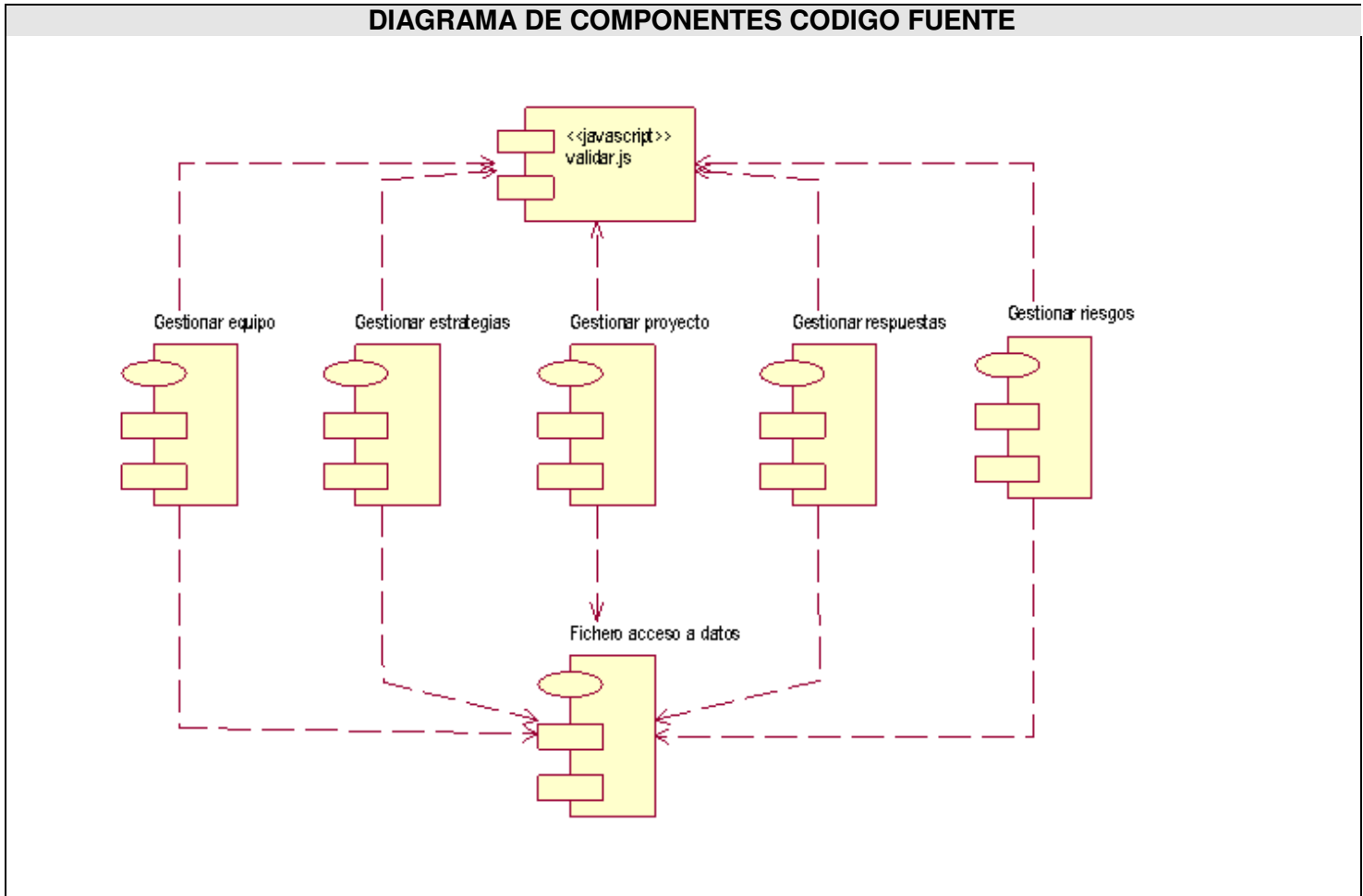


Figura 42. Diagrama de componentes Código Fuente.

4.7.1.3 Diagrama de componentes Código Ejecutable.

DIAGRAMA DE COMPONENTES WEB O CODIGO EJECTUBLE(I)

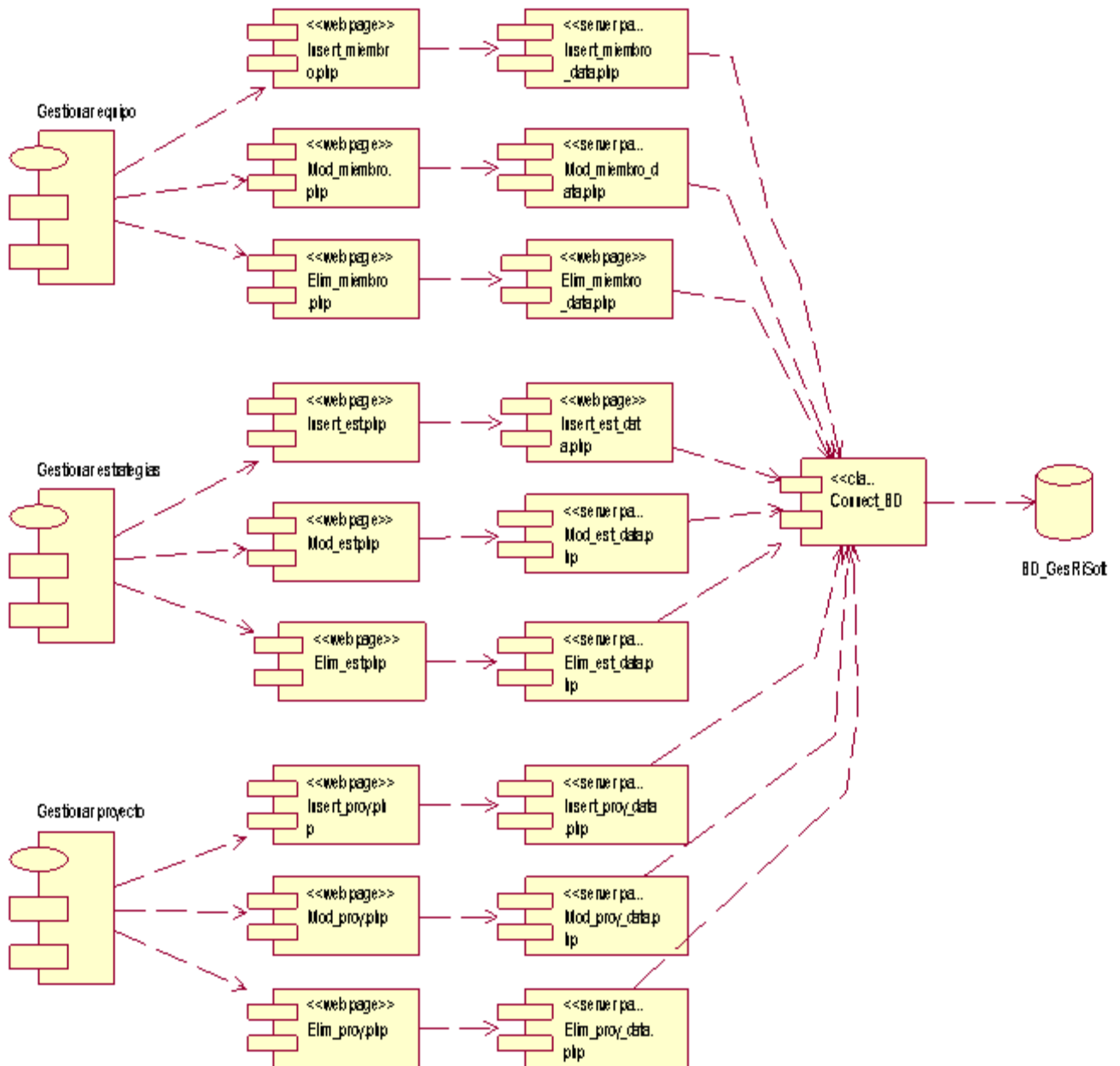


Figura 43: Diagrama de Componentes Web o código ejecutable (1)

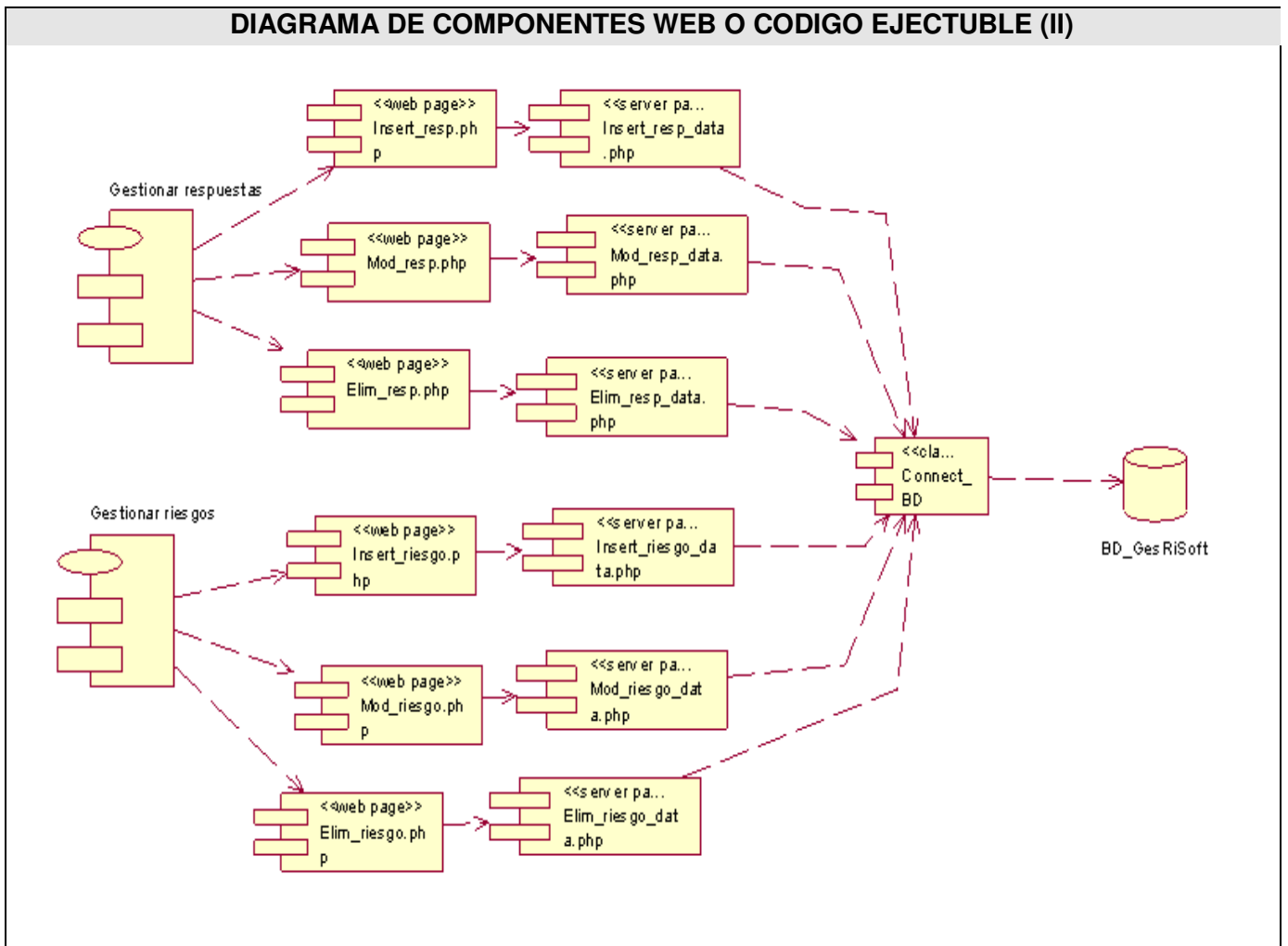


Figura 44. Diagrama de Componentes Web o código ejecutable (II)



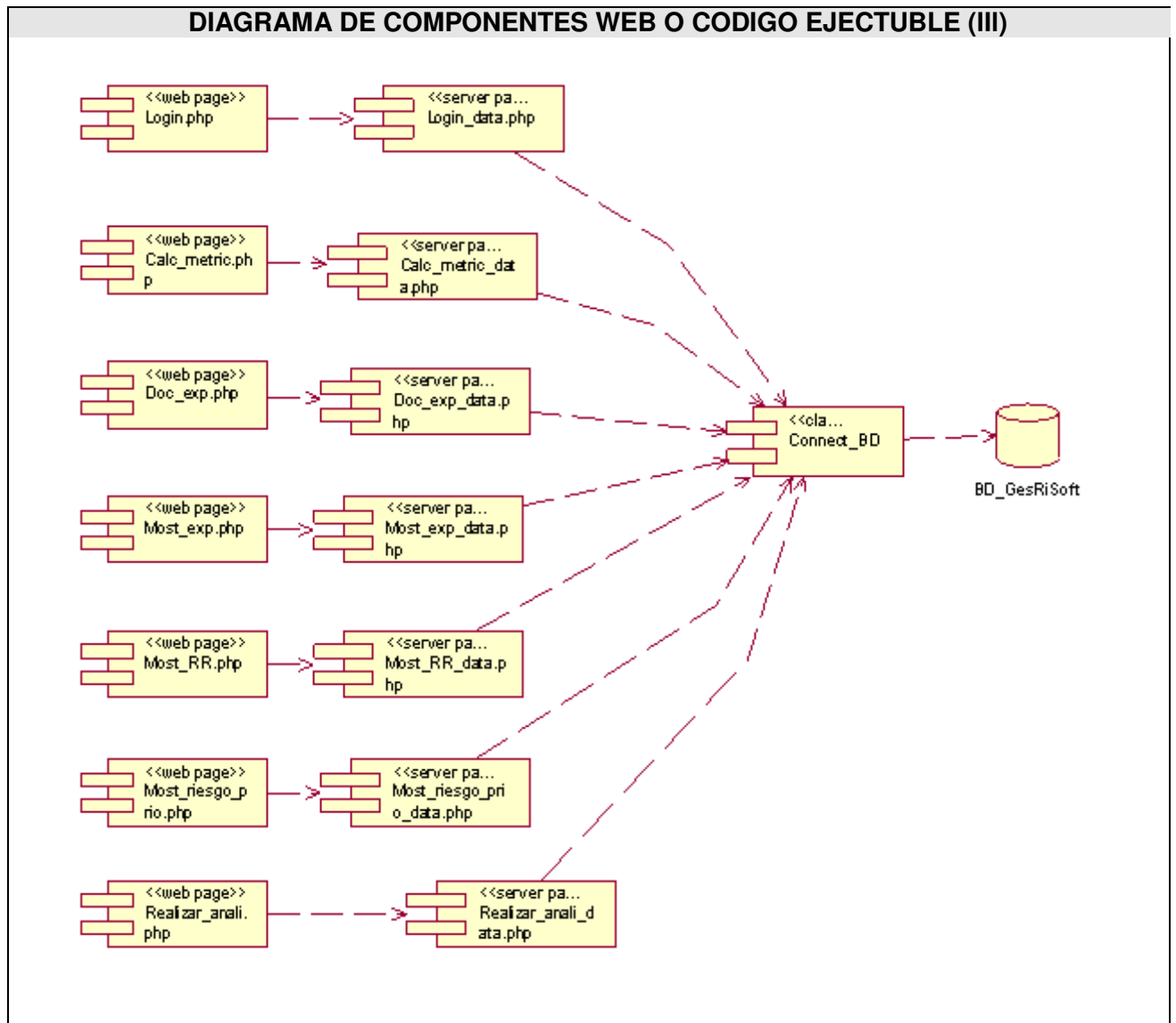


Figura 45. Diagrama de Componentes Web o código ejecutable (III)

#### 4.8 Conclusiones.

En este capítulo se realizaron los modelos del análisis y el diseño para cada uno de los casos de uso del sistema, así como también el modelo de datos. Se describieron los principios de diseño utilizados como por ejemplo los estándares de interfaz, tratamiento de errores y la forma en que se tratara la seguridad en la aplicación Web. Se desarrolló además el Modelo de Implementación en el cual se describieron los componentes a construir y su organización y dependencia entre nodos físicos en los que funcionará la aplicación.

## Capítulo 5. Análisis de Factibilidad.

### 5.1 Introducción.

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

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

### 5.2 Planificación basada en casos de uso.

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

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

$$UUCP=UAW+UUCW$$

UUCP: Puntos de casos de uso sin ajustar.

UAW: Factor de peso de los actores sin ajustar.

UUCW: Factor de peso de los casos de uso sin ajustar.

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

Tipo de actor	Descripción	Factor de peso	Actores	Total
Simple	Sistema con sistema a través de interfaz de programación.	1	0	0
Medio	Sistema con sistema mediante protocolo de interfaz basada en texto.	2	0	0
Complejo	Persona que interactúa	3	8	24

	con el sistema mediante interfaz gráfica.			
--	---	--	--	--

UAW = S (Factor \* Actores)

**UAW = 24**

Tipo de CU	Descripción	Peso	Cantidad de CU	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	11	55
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	1	10
Complejo	El caso de uso tiene más de 8 transacciones.	15	0	0

UUCW = Sumatoria (Factor \* Cant. CU)

UUCW =65

UUCP = UAW + UUCW

UUCP = 24 + 65

**UUCP= 89**

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

UCP = UUCP \* TCF \* EF

Donde:

UCP: Puntos de casos de uso ajustados.

UUCP: Puntos de casos de uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

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

Factor	Descripción	Peso	Valor asignado	Total
T1	Sistema distribuido	2	2	4

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

Sumatoria 29

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

$$TCF = 0.6 + 0.01 * 29$$

**TCF = 0.89**

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

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

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

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

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

T6: El sistemas requiere un Servidor Web Apache, con MySQL como gestor de bases de datos, PHP como lenguaje de programación.

T7: El sistema tendrá una interfaz amigable y fácil de usar.

T8: Será una aplicación con una gran capacidad para ser ejecutada en diferentes sistemas informáticos.

T9: Estará disponible a realizarle cambios necesarios.

T10: El sistema presenta una concurrencia normal.

T11: Los usuarios tendrán determinados accesos y permisos en el sistema.

T12: El sistema brindará acceso a los diferentes módulos.

T13: Su fácil uso permite que el personal no tenga que alcanzar un alto nivel de capacitación.

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

Factor	Descripción	Peso	Valor asignado	Total
E1	Familiaridad con el modelo de proyecto utilizado	1.5	5	7.5
E2	Experiencia en la aplicación	0.5	4	2
E3	Experiencia en la orientación a objetivos.	1	4	4
E4	Capacidad del analista líder.	0.5	5	2.5
E5	Motivación.	1	5	5
E6	Estabilidad de requerimientos	2	5	10
E7	Personal Part-Time	-1	3	-3
E8	Dificultad del lenguaje de programación	-1	2	-2
Sumatoria				26

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

$$EF = 1.4 - 0.03 * 26$$

$$EF = 0.60$$

$$UCP = UUCP * TCF * EF$$

$$UCP = 89 * 0.89 * 0.60$$

$$UCP = 46.05$$

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

$$E = UCP * CF$$

Donde:

E: Esfuerzo estimado en horas hombres.

UCP: Punto de casos de usos ajustados.

CF: Factor de conversión.

Para obtener el factor de conversión (CF) se cuentan cuantos valores de los que afectan el factor ambiente (E1...E6) están por debajo de la media (3), y los que están por arriba de la media para los

restantes (E7, E8). Si el total es 2 o menos se utiliza el factor de conversión 20 Horas-Hombre / Punto de Casos de uso. Si el total es 3 o 4 se utiliza el factor de conversión 28 Horas-Hombre / Punto de Casos de uso. Si el total es mayor o igual que 5 se recomienda efectuar cambios en el proyecto ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

En este caso se puede decir que:

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

$$E = 46.05 * 20$$

$$E = 921.6$$

Actividad	Porcentaje %	Horas-Hombres
Análisis	10	230.4
Diseño	20	460
Implementación	40	921.6
Pruebas	15	345.6
Sobrecarga (otras actividades)	15	345.6
<b>Total</b>	<b>100</b>	<b>2304</b>

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

Si  $ET = 2304$  horas-hombre y por cada 192 horas se tiene 1 mes eso daría un  $ET = 12$  mes-hombre. Esto significa que 1 persona puede realizar el problema analizado en un margen de 12 meses aproximadamente.

-Costo del Proyecto.

Se asume como salario promedio mensual \$50.00

CH: Cantidad de hombres.

Tiempo: Tiempo total del proyecto.

$$CH = 2 \text{ hombres}$$

$$CHM = 2 * \text{Salario Promedio}$$

$CHM = 100.00 \text{ \$/mes}$

$\text{Costo} = CHM * ET / CH$

$\text{Costo} = 100.00 * 12 / 2$

$\text{Costo} = \$ 600.00$

$\text{Tiempo} = ET / CH$

$\text{Tiempo} = 12 / 2$

Tiempo = 6 meses.

De los resultados obtenidos se interpreta que con 2 hombres trabajando en el proyecto el mismo se desarrolla en 6 meses y su costo total se estima que sea \$600.00.

### **5.3 Beneficios tangibles e intangibles.**

Los beneficios que se obtendrán con el desarrollo de este software serán fundamentalmente intangibles, ya que permitirá gestionar de forma segura los posibles riesgos que deben enfrentar los proyectos informáticos de la Facultad 9 de la Universidad de las Ciencias Informáticas. Además brindará un mayor aumento de la eficiencia en la Gestión de Riesgos, una mejor planificación del tiempo de realización y entrega de las etapas del proyecto, así como una respuesta eficiente para la toma de decisiones en la planificación del plan de actividades.

### **5.4 Análisis de costos y beneficios.**

La utilización del sistema de gestión desarrollado permitirá gestionar los riesgos informáticos que ocurran en la facultad y tener un control sobre los mismos de forma organizada. Este nuevo recurso del que dispondrá la facultad le permitirá un mayor cumplimiento de las actividades planificadas y ayudará a perfeccionar la Gestión de Riesgos de los proyectos informáticos en la Facultad 9.

### **5.5 Conclusiones.**

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

## Conclusiones.

Con la realización de este trabajo se propuso una solución al problema de la gestión de los Riesgos Informáticos en los proyectos de Software en la facultad 9 (UCI). Se hicieron estudios, investigaciones y comparaciones con otros sistemas existentes en el mundo que gestionan riesgos informáticos los cuales sirvieron como punto de partida y guía para comenzar a desarrollar la aplicación.

El desarrollo de este software dio cumplimiento al objetivo trazado: *Desarrollar un Sistema que sea capaz de gestionar los diferentes riesgos a los que se enfrenta un proyecto informático*, así como la influencia de sus respectivas ocurrencias en el desarrollo del plan trazado debido a que los requerimientos soportan al sistema y los casos de uso satisfacen las necesidades funcionales.

Se realizó el Análisis y Diseño de una aplicación capaz de gestionar no solo los riesgos informáticos en un proyecto sino que brinda la posibilidad de Gestionar también los Equipos de Proyectos, Estrategias, Proyectos y Respuestas de los mismos. Se construyó una base de datos, donde se almacena toda la información necesaria, para de esta forma garantizar la veracidad de la misma.

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

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

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

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



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

### **Recomendaciones.**

Se recomienda:

1. Poner a prueba el sistema durante un período de tiempo significativo, para comprobar su desempeño y que las funcionalidades del sistema se correspondan con la realidad de lo que se desea.
2. Continuar el estudio con el objetivo de añadir nuevas funcionalidades al sistema.
3. Lograr que los Equipos de Proyecto utilicen la herramienta desarrollada y verificar así la utilidad del mismo.

## Glosario de términos y siglas.

**RUP:** Rational Unified Process, Proceso Unificado de Desarrollo de Software en español.

**UML:** Lenguaje Unificado de Modelado, usado para modelar sistemas de software.

**CASE:** Computer Aided Software Engineering. Son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

**PHP:** Hypertext Pre-Processor. Es un ambiente script del lado del servidor que permite crear y ejecutar aplicaciones Web dinámicas e interactivas.

**MySQL:** Es un sistema de gestión de bases de datos relacional que cuentan con todas las características de un motor de BD comercial: transacciones atómicas, triggers, replicación, llaves foráneas entre otras. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar.

**SQL:** Structured Query Language. Es un lenguaje declarativo de acceso a bases de datos que permite especificar diversos tipos de operaciones sobre las mismas. Aún características del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar información de interés de una base de datos.

**HTML:** HyperText Markup Language. Lenguaje usado para escribir documentos para servidores World Wide Web.

**FTP:** File Transfer Protocol, es un protocolo de transferencia de ficheros entre sistemas conectados a una red TCP basado en la arquitectura cliente-servidor, de manera que desde un equipo cliente nos podemos conectar a un servidor para descargar ficheros desde él o para enviarle nuestros propios archivos independientemente del sistema operativo utilizado en cada equipo.

**XML:** Siglas en inglés de eXtensible Markup Language, es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.

**IMAP:** Acrónimo inglés de Internet Message Access Protocol, es un protocolo de red de acceso a mensajes electrónicos almacenados en un servidor.

**ODBC:** Open DataBase Connectivity, es un estándar de acceso a Bases de Datos desarrollado por Microsoft Corporation, su objetivo es hacer posible el acceder a cualquier dato de cualquier aplicación, sin importar qué Sistema Gestor de Bases de Datos los almacene.

## Referencias Bibliográficas.

1. TOVAR, E. C. *Curso Básico” La Gestión de Riesgos en los sistemas de información”*. 2006,
2. CHARETTE, R. N. *Software Engineering Risk Analysis and Management*. 1989,
3. VELIZ, Y. Z. *Modelo de Gestión de Riesgos en Proyectos de Desarrollo de Software*. Universidad de las Ciencias Informáticas, 2007.
4. SENA, L. y TENZER, S. *Introducción a Riesgo Informático*. 2004, Disponible en: <http://www.ccee.edu.uy/ensenian/catcomp/material/riesgo.pdf>.
5. SALAZAR, C. *RIESGOS DEL SOFTWARE*. Disponible en: <http://www.sitios.uach.cl/caminosfor/CristianSalazar/SIE/RS.html>.
6. ROMERO., A.; LOVERA., D., et al. *Gestion de riesgos con CMMI, RUP e ISO en Ingenieria de Software Minero*. 2007, Disponible en: [http://sisbib.unmsm.edu.pe/BibVirtualData/publicaciones/geologia/vol10\\_n19/a05.pdf](http://sisbib.unmsm.edu.pe/BibVirtualData/publicaciones/geologia/vol10_n19/a05.pdf).
7. MURCIA, U. D. *Gestión de riesgos en ingeniería del software*. 2006, Disponible en: <http://www.um.es/docencia/barzana/IAGP/lagp05.pdf>.
8. RESTREPO, G. G. *El Concepto y Alcance de la Gestión Tecnológica*. 2004, Disponible en: [http://ingenieria.idea.edu.co/producciones/guillermo\\_r/concepto.html](http://ingenieria.idea.edu.co/producciones/guillermo_r/concepto.html).
9. MAP *El proyecto Eurométodo. Ejercicio de validación de EM v0*. 1996,
10. MAÑAS, J. A. *PILAR.Herramientas para el Análisis y Gestión de riesgos*. 2004,
11. ROPPONEN, J. y LYYTINEN, K. *Components of Software Development Risk: Hot to address them? IEEE transactions on software engineering*. 2000,
12. PRESSMAN, R. S. *Ingeniería del Software. Un enfoque práctico*. . Mc Graw-Hill/Interamericana de España, S.A., 2002.
13. *Gestión de Riesgos* Disponible en: <http://www.di.uniovi.es/~aquilino/Asignaturas/ProyectosInformatica/Documentos/07-GestionRiesgos.pdf>.
14. INFORMÁTICA, C. S. D. *EUROMÉTODO v1*. 1997, Disponible en: [http://www.csi.map.es/csi/pdf/em\\_v1.pdf](http://www.csi.map.es/csi/pdf/em_v1.pdf).
15. BASIC, E. P. *Análisis de riesgos* Disponible en: <http://www.ar-tools.com/tools/pilar/index.html>.

16. SIGEA. *SISTEMAS DE PROTECCIÓN DE LA INFORMACIÓN*, S.L. 2007, Disponible en: [http://www.sigea.es/index.php?option=com\\_content&task=view&id=60&Itemid=1](http://www.sigea.es/index.php?option=com_content&task=view&id=60&Itemid=1).
17. EUROPEAS, C. D. L. C. Tecnologías de la información y de la comunicación en el ámbito del desarrollo. El papel de las TIC en la política comunitaria de desarrollo. En *Bruselas*. 2001.
18. GUDIÑO., P. F.; BARBOSA., O. D. H., et al. *Tutorial de Fundamentos de Bases de Datos*. 2007, Disponible en: [http://labredes.itcolima.edu.mx/fundamentosbd/sd\\_u1\\_6.htm](http://labredes.itcolima.edu.mx/fundamentosbd/sd_u1_6.htm).
19. *Apache Software Foundation*  
Disponible en: <http://www.apache.org>.
20. BUCHMANN. *Introducción a Patrones*. Disponible en: <http://www.mcc.unam.mx/~cursos/Algoritmos/javaDC99-2/patrones.html>.
21. JUNIOR, A. y ALTAMIRANDA, P. *Técnicas Emergentes para la Automatización Integrada de Procesos de Producción*. 2006, Disponible en: <http://sistemas.fsl.fundacite-merida.gob.ve/docman/view.php/16/22/GFORGE.pdf>.
22. ENTERSOFT. Disponible en: <http://www.entersoft.com.mx/Sitio/plataformaTecnologica.asp>.
23. WORSLEY, J. y DRAKE, J. *PostgreSQL Práctico*. Commandprompt, Inc 2001
24. SALNET. Disponible en: [http://www.salnet.com.ar/inv\\_mysql/pag01\\_intro.htm](http://www.salnet.com.ar/inv_mysql/pag01_intro.htm).
25. JACOBSON, I.; BOOCH, G., et al. *El Proceso Unificado de Desarrollo de Software*. Madrid: Addison Wesley, 2000.
26. ---. *El Lenguaje Unificado de Modelado. Manual de Referencia*. Madrid: Addison Wesley, 2000.

## Bibliografía.

*Apache Software Foundation*. Disponible en: <http://www.apache.org>.

*MAP El proyecto Eurométodo. Ejercicio de validación de EM v0*. 1996,

*Informática Aplicada a la Gestión Pública. Gestión de riesgos en ingeniería del software*. 2002, Disponible en: <http://www.um.es/docencia/barzana/IAGP/IAGP.html>.

*SIGEA SISTEMAS DE PROTECCIÓN DE LA INFORMACIÓN, S.L.* 2007, Disponible en: [http://www.sigea.es/index.php?option=com\\_content&task=view&id=60&Itemid=1](http://www.sigea.es/index.php?option=com_content&task=view&id=60&Itemid=1).

*Cursos Optativos de PHP y MySQL*. . 2007-2008, Disponible en: <http://teleformacion.uci.cu/>.

*Gestión de Riesgos* Disponible en: <http://www.di.uniovi.es/~aquilino/Asignaturas/ProyectosInformatica/Documentos/07-GestionRiesgos.pdf>.

*Conferencias y Clases Prácticas de Ingeniería del Software I y II*. 2007-2008, Disponible en: <http://teleformacion.uci.cu/>.

APARICIO, F. *Análisis y Gestión de Riesgos*. En *FIST Conference*. 2005.

AUTORES, C. D. *Manual de PHP*. 2005, Disponible en: <http://www.php.net/docs.php>.

BASIC, E. P. *Análisis de riesgos* Disponible en: <http://www.ar-tools.com/tools/pilar/index.html>.

BOEHM, B. *Software Risk Management*. 1989,

BUCHMANN. *Introducción a Patrones*. Disponible en: <http://www.mcc.unam.mx/~cursos/Algoritmos/javaDC99-2/patrones.html>.

CHARETTE, R. N. *Software Engineering Risk Analysis and Management*. 1989,

CONNELL, S. *Desarrollo y Gestión de Proyectos Informáticos*. McGraw Hill Iberoamericana, 1997.

ENTERSOFT. Disponible en: <http://www.entersoft.com.mx/Sitio/plataformaTecnologica.asp>.

ESPAÑA, C. S. D. I. E. *MAGERIT (v1.0) Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información*. Disponible en: <http://www.csi.map.es/csi/pg5m20.html>.

- EUROMÉTODO. *El proyecto eurométodo (v1.0)*. 1997,
- EUROPEAS, C. D. L. C. Tecnologías de la información y de la comunicación en el ámbito del desarrollo. El papel de las TIC en la política comunitaria de desarrollo. En *Bruselas*. 2001.
- FERNÁNDEZ, L. y GARCÍA, L. *Gestión del riesgo en la fase de ingeniería de requisitos de un proyecto software*. 2007.
- GÓMEZ., G. E. *Técnicas para evaluar el riesgo en el presupuesto del Capital*. .
- GUDIÑO., P. F.; BARBOSA., O. D. H., et al. *Tutorial de Fundamentos de Bases de Datos*. 2007, Disponible en: [http://labredes.itcolima.edu.mx/fundamentosbd/sd\\_u1\\_6.htm](http://labredes.itcolima.edu.mx/fundamentosbd/sd_u1_6.htm).
- IAGP. *Gestión de Riesgos en Ingeniería del Software*. 2005-2006.
- INFORMÁTICA, C. S. D. *EUROMÉTODO v1*. 1997, Disponible en: [http://www.csi.map.es/csi/pdf/em\\_v1.pdf](http://www.csi.map.es/csi/pdf/em_v1.pdf).
- INSTITUTE, P. M. *A guide to the Project Management Body of knowledge*. 2000,
- JACOBSON, I.; BOOCH, G., et al. *El Proceso Unificado de Desarrollo de Software*. Madrid: Addison Wesley, 2000.
- . *El Lenguaje Unificado de Modelado. Manual de Referencia*. Madrid: Addison Wesley, 2000.
- JUNIOR, A. y ALTAMIRANDA, P. *Técnicas Emergentes para la Automatización Integrada de Procesos de Producción*. 2006, Disponible en: <http://sistemas.fsl.fundacite-merida.gob.ve/docman/view.php/16/22/GFORGE.pdf>.
- MAÑAS, J. A. PILAR. *Herramientas para el Análisis y Gestión de riesgos*. 2004,
- MENÉNDEZ, R. y ASENSIO., B. *Gestión de Riesgos en Ingeniería del Software- Reducción, supervisión y gestión del Riesgo*. Disponible en: <http://www.um.es/docencia/barzana>.
- MURCIA, U. D. *Gestión de riesgos en ingeniería del software*. 2006, Disponible en: <http://www.um.es/docencia/barzana/IAGP/lagp05.pdf>.
- NUCHERA, A. H. *Una introducción a la gestión de riesgos tecnológicos. Gestión de la innovación y de la tecnología*. 2004.
- PRESSMAN, R. S. *Ingeniería del Software. Un enfoque práctico*. . Mc Graw-Hill/Interamericana de España, S.A., 2002.

- RESTREPO, G. G. *El Concepto y Alcance de la Gestión Tecnológica*. 2004, Disponible en: [http://ingenieria.idea.edu.co/producciones/guillermo\\_r/concepto.html](http://ingenieria.idea.edu.co/producciones/guillermo_r/concepto.html).
- ROMERO., A.; LOVERA., D., *et al. Gestión de riesgos con CMMI, RUP e ISO en Ingeniería de Software Minero*. 2007, Disponible en: [http://sisbib.unmsm.edu.pe/BibVirtualData/publicaciones/geologia/vol10\\_n19/a05.pdf](http://sisbib.unmsm.edu.pe/BibVirtualData/publicaciones/geologia/vol10_n19/a05.pdf).
- ROPPONEN, J. y LYYTINEN, K. *Components of Software Development Risk: Hot to address them? IEEE transactions on software engineering*. 2000,
- SALAZAR, C. *RIESGOS DEL SOFTWARE*. Disponible en: <http://www.sitios.uach.cl/caminosfor/CristianSalazar/SIE/RS.html>.
- SALNET. Disponible en: [http://www.salnet.com.ar/inv\\_mysql/pag01\\_intro.htm](http://www.salnet.com.ar/inv_mysql/pag01_intro.htm).
- SCHMULLER J. *Aprendiendo UML en 24 horas*. México: 2000.
- SENA, L. y TENZER, S. *Introducción a Riesgo Informático*. 2004, Disponible en: <http://www.ccee.edu.uy/ensenian/catcomp/material/riesgo.pdf>.
- SIGEA. *SISTEMAS DE PROTECCIÓN DE LA INFORMACIÓN, S.L.* 2007, Disponible en: [http://www.sigea.es/index.php?option=com\\_content&task=view&id=60&Itemid=1](http://www.sigea.es/index.php?option=com_content&task=view&id=60&Itemid=1).
- TOVAR, E. C. *Curso Básico” La Gestión de Riesgos en los sistemas de información”*. 2006,
- VELIZ, Y. Z. *Modelo de Gestión de Riesgos en Proyectos de Desarrollo de Software*. Universidad de las Ciencias Informáticas, 2007.
- WELLING, L. y THOMSON, L. *Desarrollo Web con PHP y MySQL*. 2005.
- WORSLEY, J. y DRAKE, J. *PostgreSQL Práctico*. Commandprompt, Inc 2001