



Universidad de las Ciencias Informáticas

Título: “Herramienta para la Identificación de Riesgos en proyectos de Software Educativo y Multimedia”

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

Autores:

Adriana Ludeiro Montero

Juan Miguel Martínez Vargas

Tutor:

Lic. Héctor Matías González

Ciudad de la Habana

Junio de 2009

Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber.

Albert Einstein

A mis padres por ser la fuente de mis fuerzas, por haberme ayudado a construir mi futuro, por todo su amor, dedicación y sobre todo, por tener en ustedes mi más preciado tesoro.

A mi hermanita por lo mucho que la quiero, por cada expresión de amor y añoranza, cada palabra de aliento y confianza en estos cinco años.

A mi tía Dulce por ser como una madre y ayudarme en todo, por ser mi ejemplo.

A la memoria de mi abuela Teófila, por educarme y brindarme su amor. Aunque ya no estés a mi lado, siempre te tengo presente y te llevo en el corazón.

Adriana

A mi hermosa familia, en especial a mis padres y a mi hermano querido que no solo se merecen esto, sino mucho más, a ellos les doy gracias por educarme, por amarme, por adorarme y acompañarme en cada segundo, en cada minuto, en cada hora del día, sin más su hijo que los quiere con todo su corazón.

Juan Miguel

De Adriana:

Quisiera agradecerles en este momento a muchas personas que de una forma u otra han hecho posible que este sueño se haga realidad, a aquellas que confiaron en mí y me ayudaron en el desarrollo de este trabajo, de forma especial:

- *A mi mamá por su preocupación constante y por darme fuerzas para levantarme en los momentos más difíciles.*
- *A mi papá por su forma especial de quererme y por estar siempre pendiente de lo que necesitara.*
- *A mi hermanita por todo su cariño, por compartir conmigo los buenos y malos momentos y sobre todo, por ser tan especial conmigo.*
- *A mis tías Dulce, Delta, Mirta, Marta que siempre me apoyaron en todo y por confiar en que lo lograría.*
- *A toda mi familia, que aunque lejos no dejaron de preocuparse por mis estudios.*
- *A mi novio Jorge Carlos por todo el amor y cariño que me brinda y por soportar con paciencia todas mis obsesiones durante estos 5 años.*
- *A Armando Ortiz Cabrera por sus consejos y por el apoyo infinito que me brindó para lograr este sueño.*
- *A nuestro tutor Héctor Matías por guiarnos en este largo camino para lograr nuestro objetivo.*
- *A mis compañeros de aula que han aportado su granito de arena en la realización de este trabajo.*
- *A las niñas del cuarto por tener que aguantarme y ser parte de estos años dignos de recordar.*
- *A Yasnier, Carlos Lima, Ismael, Liana Isabel, Osdalme por su aporte y por siempre estar dispuestos a ayudarme ante cualquier problema.*
- *A todas aquellas personas que contribuyeron a que este sueño se hiciera realidad.*

¡A todos muchas gracias!

De Juan Miguel:

No sé cómo empezar a agradecerles a todas aquellas personas que me apoyaron y me ayudaron en el desarrollo de este trabajo, y que de una forma u otra han hecho posible no solo mi sueño sino el de toda mi familia y porque no, el de ustedes mismos que son los principales protagonistas de mi graduación.

- *En primer lugar agradecerles a los profesores de Multisaber en especial a: Liana Isabel, Ismael y Osdalme que siempre estuvieron presentes ante cualquier duda e inconveniente que surgiera.*
- *Al Ing. Armando Ortiz Cabrera que nos supo guiar por el camino correcto, por lo que fue posible terminar la aplicación a tiempo.*
- *A nuestro tutor el Lic. Héctor Matías González que nos guió y ayudó en el desarrollo del trabajo.*
- *A todos mis compañeros de grupo, en especial a los de mi cuarto por haberme aguantado estos cinco años.*
- *A todos mis amigos KDT y a Darian Zamora López por aportar su granito de arena.*
- *A mi queridísima novia por aguantarme y estar conmigo en las buenas y las malas.*
- *A la Revolución y a nuestro comandante en jefe Fidel Castro Ruz por facilitarnos todo lo que nos ha hecho falta y permitirnos formarnos como mejores profesionales.*
- *Y nunca puede faltar la familia, que está de más decir lo que pueden hacer por ti, este agradecimiento es muy importante y muy especial para mí, pues mi familia lo es todo en mi vida, solo decir que los quiero con todo mi corazón y que de no haber sido por ustedes hoy no estaría aquí, especialmente por mis padres queridos, por mis cuatro abuelos que aunque hay dos de ellos que no están presente es como si lo estuvieran, por mis tíos y por mi hermanito del alma.*

¡Les doy las gracias a todos desde el fondo de mi corazón!

DECLARACIÓN DE LA AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Adriana Ludeiro Montero
Autor

Juan Miguel Martínez Vargas
Autor

Lic. Héctor Matías González
Tutor

RESUMEN

La identificación de los riesgos en los proyectos de software es una importante fase en el proceso de Gestión de Riesgos, ésta permite identificar los elementos que afectan al proyecto y tomar las medidas necesarias para evadirlos.

En el presente trabajo se describe el análisis, diseño e implementación de la primera versión de una herramienta que sirve de apoyo al proceso de Identificación de Riesgos en los proyectos de Software Educativo y Multimedia en la Universidad de las Ciencias Informáticas. El mismo está basado en la metodología y conceptos propuestos en el campo de la Gestión de Riesgos por el Instituto de Ingeniería de Software y el Instituto de Gestión de Proyectos.

La herramienta está desarrollada sobre tecnologías Web, implementada con PHP como lenguaje de programación y MySQL como gestor de Base de Datos. Las funcionalidades principales se centran en la gestión de los riesgos, de los factores que los provocan y propiedades del entorno de trabajo de los proyectos que propician la aparición de riesgos y factores, además se almacenan las relaciones entre los elementos antes mencionados. El desarrollo del software estuvo guiado por los estándares de la metodología de desarrollo de software RUP.

Palabras clave: Riesgos, Gestión de Riesgos, Identificación de Riesgos, Software Educativo y Multimedia.

Tabla de contenido

INTRODUCCIÓN 1

 Situación problemática 1

CAPÍTULO1 FUNDAMENTACIÓN TEÓRICA 5

 1.1 Introducción 5

 1.2 Gestión de Proyectos 5

 1.3 Gestión de Riesgos 6

 1.3.1 Historia de la Gestión de Riesgos 6

 1.3.2 Modelo de Gestión de Riesgos 7

 1.4 Identificación de Riesgos 8

 1.5 Herramientas, Técnicas y Métodos para la Identificación de Riesgos 9

 1.5.1 Revisiones de Documentación 9

 1.5.2 Técnicas de Recopilación de Información 10

 1.5.3 Análisis mediante Lista de Control 11

 1.5.4 Análisis de Asunciones 11

 1.5.5 Técnicas de Diagramación 11

 1.6 Salidas del proceso de Identificación de Riesgos 11

 1.6.1 Registro de Riesgos 11

 1.7 Taxonomía del SEI para la Identificación de Riesgos 12

 1.8 Aplicaciones informáticas de apoyo a la Gestión de Proyectos de Software 14

 1.9 Identificación de Riesgos en proyectos de Software Educativo y Multimedia en la UCI 15

 1.10 Propuesta de solución 16

 1.11 Tendencias y tecnologías a utilizar 17

 1.11.1 Metodología de Desarrollo de Software 17

 1.11.2 Lenguaje Unificado de Modelado (UML) 20

 1.11.3 Herramienta CASE de Desarrollo de Software 21

 1.11.4 Fundamentación del lenguaje a utilizar 22

 1.11.5 Sistemas Gestores de Bases de Datos 23

 1.11.6 Entorno de Desarrollo 25

 1.12 Conclusiones parciales 25

CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA	27
2.1 Introducción.....	27
2.2 Modelo de Dominio	27
2.2.1 Descripción del Modelo de Dominio	27
2.2.2 Diagrama del Modelo de Dominio	28
2.3 Requisitos del sistema	29
2.3.1 Definición de los requisitos funcionales.....	29
2.3.2 Definición de los requisitos no funcionales.....	31
2.4 Propuesta de solución.....	32
2.4.1 Actores del sistema	32
2.4.2 Diagrama de casos de uso del sistema.....	33
2.4.3 Descripción de los casos de uso del sistema.....	35
2.5 Conclusiones parciales	59
CAPÍTULO 3 ANÁLISIS Y DISEÑO DEL SISTEMA.....	60
3.1 Introducción.....	60
3.2 Arquitectura	60
3.3 Modelo de análisis.....	61
3.3.1 Diagrama de clases del análisis	62
3.3.2 Diagrama de interacción.....	63
3.4 Modelo de diseño	64
3.4.1 Diseño de la Base de Datos	64
3.4.2 Diagrama de clases del diseño.....	67
3.4.3 Principios del diseño.....	67
3.5 Estándares en la interfaz de la aplicación.....	67
3.6 Tratamiento de excepciones	68
3.7 Conclusiones parciales	68
CAPÍTULO 4 IMPLEMENTACIÓN Y PRUEBAS	69
4.1 Introducción.....	69
4.2 Modelo de implementación	69
4.2.1 Diagramas de componentes.....	69

4.2.2 Diagrama de despliegue.....	75
4.3 Pruebas	75
4.3.1 Pruebas de Caja Negra para el caso de uso <Autenticar usuario>.....	76
4.3.2 Pruebas de Caja Negra para el caso de uso <Gestionar riesgos>	77
4.3.3 Pruebas de Caja Negra para el caso de uso <Adicionar factor de riesgo>	80
4.3.4 Pruebas de Caja Negra para el caso de uso <Gestionar factor de riesgo>	81
4.3.5 Pruebas de Caja Negra para el caso de uso <Gestionar relación riesgo_factor>.....	82
4.4 Conclusiones parciales	85
CONCLUSIONES GENERALES.....	86
RECOMENDACIONES.....	87
REFERENCIAS BIBLIOGRÁFICAS	88
GLOSARIO DE TÉRMINOS.....	90
ANEXOS.....	93

Índice de tablas

Tabla 1: Resumen de herramientas de Gestión de Riesgos.	15
Tabla 2: Comparación de herramientas de soporte de Identificación de Riesgos.....	16
Tabla 3: Tipos de modelos en UML	20
Tabla 4: Descripción de los actores del sistema.....	33
Tabla 5: Caso de uso del sistema<Gestionar riesgos>	35
Tabla 6: Caso de uso del sistema<Adicionar factor de riesgo>.....	37
Tabla 7: Caso de uso del sistema<Gestionar factor de riesgo>	39
Tabla 8: Caso de uso del sistema <Adicionar entorno de trabajo>	40
Tabla 9: Caso de uso del sistema<Gestionar entorno de trabajo>.....	41
Tabla 10: Caso de uso del sistema< Gestionar valor entorno de trabajo>	43
Tabla 11: Caso de uso del sistema<Gestionar relación riesgo_factor>	45
Tabla 12: Caso de uso del sistema<Gestionar relación riesgo_entorno>	47
Tabla 13: Caso de uso del sistema<Gestionar relación factor_entorno>	50
Tabla 14: Caso de uso del sistema<Gestionar usuarios>	52
Tabla 15: Caso de uso del sistema<Generar reportes>	55
Tabla 16: Caso de uso del sistema<Exportar a PDF>.....	56
Tabla 17: Caso de uso del sistema<Notificar disparo>	57
Tabla 18: Caso de uso del sistema<Autenticar usuario>	58

Índice de figuras

Figura 1: Disciplinas a través de las iteraciones..... 18

Figura 2: Modelo de Dominio..... 28

Figura 3: Diagrama de casos de uso del sistema..... 34

Figura 4: Diagrama de clases del análisis <Gestionar riesgos>..... 63

Figura 5: Diagrama de colaboración<Insertar riesgos>..... 64

Figura 6: Diagrama de clases persistentes..... 65

Figura 7: Modelo de datos. 66

Figura 8: Diagrama de componentes general del sistema. 70

Figura 9: Diagrama de componentes del paquete Interfaz..... 71

Figura 10: Diagrama de componente del paquete PagServidoras..... 72

Figura 11: Diagrama de componentes del Paquete Negocio. 73

Figura 12: Diagrama de componentes del Paquete Clases. 74

Figura 13: Diagrama de despliegue..... 75

INTRODUCCIÓN

Desde hace algunos años la industria del software ha sufrido un gran número de cambios, dados esencialmente por el auge en el uso de las tecnologías, lo que ha implicado el aumento de proyectos informáticos a nivel mundial, así como las dimensiones de los mismos. Factores como la supervisión, el control, la dirección, la competencia de mercado, entre otros, han convertido a la Gestión de Proyectos de Software en una compleja profesión, en la que se han tenido que formular teorías que ayuden al buen desenvolvimiento del novedoso campo.

Cuba y particularmente la Universidad de las Ciencias Informáticas (UCI) no están ausentes de tales cambios, por lo que la Universidad ha organizado la producción a través de los Polos, en los cuales se encuentran agrupados los proyectos con una misma línea de desarrollo. La creación de los Polos ha beneficiado en gran medida el proceso productivo permitiendo una mejor organización del trabajo. Uno de estos Polos es el de Software Educativo y Multimedia que concentra todos aquellos proyectos que se vinculan de una forma u otra a este perfil.

Los proyectos de Software Educativo y Multimedia desempeñan un papel importante en el sistema educacional, pero como es natural la confección de los mismos tiene asociado riesgos. Un riesgo es un evento o condición incierto que, si se produce, tiene un efecto positivo o negativo sobre al menos un objetivo del proyecto, como tiempo, coste, alcance o calidad (PMI, 2004). Es muy importante comprenderlos y tomar las medidas proactivas para evitarlos o gestionarlos, siendo todo esto, un elemento clave de una buena Gestión de Proyectos de software.(PRESSMAN, 2002)

La Gestión de Proyectos tiene varias áreas de procesos y dentro de ella contiene la Gestión de Riesgos, que es la práctica compuesta por procesos, métodos y herramientas que posibilitan la gestión de los mismos en un proyecto, desde su inicio y a lo largo de todo su ciclo de vida. Esta actividad procura formalizar conocimiento orientado a la identificación, minimización y evitación de los riesgos más importantes en proyectos de desarrollo de software.

Por tanto, para lograr un producto con calidad que cumpla con las necesidades y exigencias del cliente, todo proyecto informático debe proponerse realizar correctamente la gestión de los riesgos y dentro de ésta, la identificación, que es el proceso que consiste en la determinación de elementos de riesgos

potenciales mediante la utilización de algún método consistente y estructurado; considerándose el paso más importante entre todos aquellos que componen las actividades de Administración de Riesgos, ya que sin la correcta determinación de los mismos, no es posible desarrollar e implementar anticipadamente respuestas apropiadas a los problemas que puedan surgir en el proyecto.(FUTRELL, 2002)

Situación problemática:

Actualmente la Gestión de Proyectos y en especial la Gestión de Riesgos en la Universidad y particularmente en los Proyectos de Software Educativo y Multimedia, no se está desarrollando de forma adecuada. Muchos de los desarrolladores y directivos que intervienen en la actividad productiva coinciden en un 100% en que los riesgos no son debidamente identificados (ZULUETA VÉLIZ y DESPAIGNE HERRERA, 2007), a pesar de la importancia que reviste este proceso para lograr que los proyectos tengan una buena calidad y que sean entregados según lo planificado.

En los últimos años la producción de software en la UCI ha centrado su atención en la utilización de herramientas y tecnologías libres, buscando bajar los costos de producción, así como la contribución de software a la industria nacional que no posean dependencias o restricciones de licencias. Las aplicaciones que desarrolla la Universidad en el campo del Software Educativo y Multimedia no han quedado exentos de dicha política, y han tenido que comenzar a enfrentar nuevos riesgos, antes desconocidos para los equipos que desarrollan este tipo de aplicaciones.

La insuficiente identificación de los riesgos se debe fundamentalmente a la inexperiencia del equipo de desarrollo y de los líderes de proyectos, que en su mayoría son jóvenes, sin dejar atrás el proceso de utilización de nuevas tecnologías y herramientas. Todo esto trae problemas como: cambios de requerimientos, incumplimientos con las fechas del cronograma, gasto de recursos humanos, pérdida de tiempo, entre otros, que pueden provocar cuantiosos daños en una amplia gama de servicios informáticos y en la economía.

Dados los elementos planteados en la situación problemática, el **problema a resolver** queda formulado de la siguiente forma:

¿Cómo contribuir a la correcta Identificación de Riesgos en los proyectos de Software Educativo y Multimedia en la UCI?

El **objeto de estudio** es la Gestión de Proyectos de Software, fundamentalmente las herramientas informáticas que ayudan a la Gestión de Riesgos, siendo éste el **campo de acción**.

La idea a defender que se plantea es que con el desarrollo de una herramienta, se podrá mejorar el proceso de Identificación de Riesgos en los proyectos de Software Educativo y Multimedia.

El **objetivo general** que persigue este trabajo, consiste en realizar el Análisis, Diseño e Implementación de la primera versión de una herramienta que ayude a la identificación de los riesgos en proyectos de Software Educativo y Multimedia. Y a raíz del mismo se derivaron los siguientes **objetivos específicos**:

- Realizar una revisión del estado del arte en cuanto a herramientas similares.
- Recopilar la mayor cantidad de información referente a los problemas que presentan los proyectos de Software Educativo y Multimedia.
- Definir las herramientas y lenguajes de programación adecuados para el desarrollo de la aplicación.
- Realizar el Análisis y Diseño de la aplicación.
- Realizar la Implementación del sistema.
- Realizar pruebas a la aplicación.

Para dar cumplimiento a lo antes planteado se definieron las siguientes **tareas**:

- Caracterizar las principales herramientas existentes que ayudan a la Gestión de Riesgos en proyectos informáticos.
- Comparar las herramientas estudiadas de acuerdo a parámetros definidos.
- Definir los datos y la forma en que se almacenará la experiencia en cuanto a los riesgos y problemas que sufren los proyectos de Software Educativo y Multimedia.
- Realizar entrevistas a los equipos de desarrollo de proyectos de Software Educativo y Multimedia.
- Definir la metodología que servirá para guiar el desarrollo y documentación del software.
- Seleccionar las herramientas y lenguajes de programación precisos para el desarrollo de la aplicación.
- Realizar el análisis y diseño de la aplicación.

- Definir el alcance de la implementación de acuerdo al tiempo del cronograma dedicado a esta tarea y a los módulos definidos en el diseño del software.
- Implementar el software, según el alcance definido.
- Validar cada funcionalidad del sistema basado en el método de prueba Caja Negra.

Para realizar dichas tareas de investigación se ponen en práctica los siguientes **métodos científicos**:

Empíricos:

- *Entrevistas* a los líderes de proyecto, para recopilar toda la información necesaria sobre el proceso de Identificación de Riesgos.

Teóricos:

- *Método Analítico-Sintético*, para analizar toda la bibliografía utilizada.
- *Modelación*, que ayudará a la modelación de diagramas para lograr un mejor entendimiento de lo que se va a implementar.
- *Método Histórico*, que sirve de apoyo para investigar si existen herramientas informáticas que apoyen al proceso de Identificación de Riesgos.

El presente documento está estructurado por capítulos, a continuación se expone brevemente una descripción de cada uno.

Capítulo 1: Se tratan conceptos específicos del dominio del problema y una breve descripción de algunas de las aplicaciones informáticas de Gestión de Riesgos existentes. Además se abordan temas relacionados con las herramientas, lenguaje y gestor de base de datos a utilizar, así como el lenguaje de modelado y la metodología de desarrollo de software.

Capítulo 2: Se describe el Modelo de Dominio como alternativa del Modelo de Negocio. Se muestran las funcionalidades y restricciones que posee el sistema por medio de los requisitos funcionales y no funcionales, así como la propuesta de solución mediante el diagrama de casos de uso del sistema.

Capítulo 3: Se muestran los diagramas de clases del análisis, lo cual influye a la hora de concebir el diseño del software, los diagramas de clases Web que reflejan de una forma más clara cómo va a

funcionar dicho sistema y qué clases están presentes en el mismo, además se muestra el diseño de la Base de Datos.

Capítulo 4: Se describe cómo está implementado el sistema a través de los diagramas de componentes y el diagrama de despliegue, además se exponen y detallan las diferentes pruebas que se le realizan al mismo.

Capítulo 1

Fundamentación Teórica

1.1 Introducción

En el presente capítulo se definen conceptos relacionados con la Gestión de Riesgos, profundizando en el proceso de Identificación de Riesgos en proyectos de software, que constituye el eje central de esta investigación. Se exponen además las principales características de algunas herramientas de Gestión de Riesgos que existen. Por último se analizan las tecnologías y tendencias actuales a nivel mundial que permiten el desarrollo de la propuesta de solución.

1.2 Gestión de Proyectos

Según la guía del PMBOK¹, la dirección de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto para satisfacer los requisitos del mismo. La dirección de proyectos se logra mediante la aplicación e integración de los procesos de inicio, planificación, ejecución, seguimiento y control, y cierre. Esta guía organiza todos los procesos que intervienen en la Gestión de Proyectos en 9 áreas del conocimiento:

- Gestión de la Integración del Proyecto.
- Gestión del Alcance del Proyecto.
- Gestión del Tiempo del Proyecto.
- Gestión de los Costes del Proyecto.
- Gestión de la Calidad del Proyecto.
- Gestión de los Recursos Humanos del Proyecto.
- Gestión de las Comunicaciones del Proyecto.
- Gestión de los Riesgos del Proyecto.
- Gestión de las Adquisiciones del Proyecto.

¹ Guía de los Fundamentos de la Dirección de Proyectos (Guía del PMBOK)

1.3 Gestión de Riesgos

Los autores Juan Fuente y Cueva Lovelle, definieron el término **riesgo** como cualquier suceso que pueda afectar negativamente a la marcha del proyecto en el futuro, es asociado de manera inexorable a cualquier actividad que se lleve a cabo y que imponga la decisión entre varias alternativas, por tanto, acompaña todo cambio y está presente en cada decisión. (FUENTE y LOVELLE, 2006)

El SEI (*Software Engineering Institute*) define al Riesgo como “*la posibilidad de sufrir una pérdida*” y a la Gestión de Riesgos como la práctica compuesta de procesos, métodos y herramientas que posibilita la gestión de los riesgos en un proyecto y que provee de un entorno disciplinado para la toma de decisiones pro-activas en base a determinar constantemente que puede ir mal (riesgos), identificar cuáles son los riesgos más importantes en los cuales enfocarse e implementar estrategias para gestionarlos.(SEI, 2004)

La Gestión de Riesgos es importante debido a que ayuda a evitar desastres y exceso de trabajo (ROSENBERG *et al.*, 1999). Una correcta Gestión de Riesgos posibilita, por tanto, el aprovechamiento óptimo de recursos y provoca como consecuencia, el aumento de ganancias y la disminución de pérdidas.

1.3.1 Historia de la Gestión de Riesgos

Muchos autores han trabajado en temas como la evolución de las teorías sobre Riesgos y la comparación de modelos y métodos. Por ejemplo, se ha dividido la Gestión de Riesgos en generaciones para exponer mejor las características y evolución de cada una.

En su estudio exploratorio sobre los métodos de Gestión de Proyectos de alto riesgo, Marcelo, Rodenes y Torralba (MARCELO y RODENES, 2003) plantean la evolución de los modelos de gestión de los riesgos en forma de sucesivas generaciones:

Primera Generación: G1

Es la generación casuística o tradicional, donde se limitaban las tareas a la Identificación de Riesgos en los proyectos con técnicas basadas en cuestionarios, listas de incidencias y de las medidas para contrarrestarlas. Se identificaban casos de riesgos y se extrapolaban a otros proyectos. No había una planificación específica. En esta generación se definen los riesgos tecnológicos y las listas de comprobación de riesgos.

Segunda Generación: G2

Es la generación taxonómica que está dada a principios de los años 90 y está basada en modelos de procesos y eventos. Dentro de esta generación se pueden incluir:

- Modelo de Boehm.
- Modelo de Hall.
- Modelo del SEI.
- Modelo SPR de mejora de capacidad en la Gestión de Riesgos.
- Modelo SERIM (Software Engineering Risk Management) de Karolak: IEEE.
- Modelo del PMI.
- Modelo de McFarlan (adelantos de G3).

Tercera Generación: G3

Es la generación causal o emergente, 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. Los principales modelos de Gestión de Riesgos propuestos son:

- Modelo Eurométodo.
- Modelo MAGERIT de Gestión de Riesgos en Sistemas adaptados a Proyectos (Transición).
- Modelo de eventos de MAGERIT – Proyectos (Transición).
- Modelo RiskMan e iniciativa RiskDriver.
- Modelo DriveSPI.

1.3.2 Modelo de Gestión de Riesgos

Existen en el mundo varios modelos de Gestión de Riesgos, pero la mayoría coinciden en los procesos que proponen. Según la Guía de los Fundamentos de la Dirección de Proyecto, la Gestión de los Riesgos del Proyecto incluye los procesos relacionados con la planificación de la Gestión de Riesgos, la identificación y el análisis de riesgos, las respuestas a los riesgos y el seguimiento y control de riesgos de un proyecto; la mayoría de estos procesos se actualizan durante el proyecto. Los objetivos de la Gestión de los Riesgos del Proyecto son aumentar la probabilidad y el impacto de los eventos positivos, y

disminuir la probabilidad y el impacto de los eventos adversos para el proyecto (PMI, 2004). En el **Anexo 1** se presentan los procesos mencionados anteriormente.

A continuación se describe cada fase del proceso de Gestión de Riesgos:

Planificación de la Gestión de Riesgos: decidir cómo enfocar, planificar y ejecutar las actividades de Gestión de Riesgos para un proyecto.

Identificación de Riesgos: determinar qué riesgos pueden afectar al proyecto y documentar sus características.

Análisis Cualitativo de Riesgos: priorizar los riesgos para realizar otros análisis o acciones posteriores, evaluando y combinando su probabilidad de ocurrencia y su impacto.

Análisis Cuantitativo de Riesgos: analizar numéricamente el efecto de los riesgos identificados en los objetivos generales del proyecto.

Planificación de la Respuesta a los Riesgos: desarrollar opciones y acciones para mejorar las oportunidades y reducir las amenazas a los objetivos del proyecto.

Seguimiento y Control de Riesgos: realizar el seguimiento de los riesgos identificados, supervisar los riesgos residuales, identificar nuevos riesgos, ejecutar planes de respuesta a los riesgos y evaluar su efectividad a lo largo del ciclo de vida del proyecto.(PMI, 2004)

Todos estos pasos ayudan a realizar una buena Gestión de Riesgos, sin embargo la etapa de la Identificación de Riesgos se considera como una de las más importantes, ya que al identificar correctamente los mismos se pueden buscar soluciones a los problemas que puedan surgir durante la vida de un proyecto.

1.4 Identificación de Riesgos

La Identificación de Riesgos determina qué riesgos pueden afectar al proyecto y documenta sus características. Entre las personas que participan en actividades de Identificación de Riesgos se pueden incluir las siguientes:

- Director del proyecto.
- Miembros del equipo del proyecto.

- Equipo de Gestión de Riesgos (expertos en la materia).
- Ajenos al equipo del proyecto.
- Clientes y usuarios finales.
- Otros directores de proyectos.
- Interesados y expertos en Gestión de Riesgos.

La Identificación de Riesgos es un proceso iterativo porque se pueden descubrir nuevos riesgos a medida que el proyecto avanza a lo largo de su ciclo de vida. La frecuencia de la iteración y quién participará en cada ciclo variará de un caso a otro. El equipo del proyecto debe participar en el proceso para poder desarrollar y mantener un sentido de pertenencia y responsabilidad por los riesgos y las acciones asociadas con la respuesta a los riesgos.(PMI, 2004)

Se pueden diferenciar dos tipos de riesgos en la identificación, los genéricos y los específicos del producto. Los **riesgos genéricos** son una amenaza potencial para todos los proyectos de software, independientemente de lo que traten éstos. Los **riesgos específicos** sólo se pueden identificar por parte de los responsables del proyecto, los cuales tienen una clara visión de la tecnología, el personal y el entorno específico del proyecto.(PRESSMAN, 2002)

Tanto los riesgos genéricos como los específicos del producto se deberían identificar sistemáticamente. Tom Gilb tiene toda la razón cuando cita: "Si no atacas activamente a los riesgos. Ellos te atacarán activamente a ti".(MURCIA, 2006)

1.5 Herramientas, Técnicas y Métodos para la Identificación de Riesgos

La guía del PMBOK propone algunas herramientas y técnicas que son necesarias para lograr una satisfactoria Identificación de Riesgos. Algunas de éstas se mencionan a continuación:

1.5.1 Revisiones de Documentación

Se puede realizar una revisión estructurada de la documentación del proyecto, incluidos planes, asunciones, archivos de proyectos anteriores y otra información. La calidad de los planes, así como la consistencia entre esos planes y con los requisitos y asunciones del proyecto, pueden ser indicadores de riesgos en el proyecto.

1.5.2 Técnicas de Recopilación de Información

Algunos ejemplos de técnicas de recopilación de información utilizadas para identificar los riesgos son:

- **Tormenta de ideas:** la meta de la tormenta de ideas es obtener una lista completa de los riesgos del proyecto. El equipo del proyecto suele realizar tormentas de ideas, a menudo con un grupo multidisciplinario de expertos que no pertenecen al equipo. Se generan ideas acerca de los riesgos del proyecto bajo el liderazgo de un facilitador. Pueden utilizarse como marco categorías de riesgo tales como una estructura de desglose del riesgo. Los riesgos luego son identificados y categorizados por tipo de riesgo y sus definiciones son refinadas.
- **Técnica Delphi:** la técnica Delphi es una forma de llegar a un consenso de expertos. Los expertos en riesgos de proyectos participan en esta técnica de forma anónima. Un facilitador emplea un cuestionario para solicitar ideas acerca de los riesgos importantes del proyecto. Las respuestas son resumidas y luego enviadas nuevamente a los expertos para que realicen comentarios adicionales. En pocas rondas de este proceso se puede lograr el consenso. La técnica Delphi ayuda a reducir sesgos en los datos y evita que cualquier persona ejerza influencias impropias en el resultado.
- **Entrevistas:** entrevistar a participantes experimentados del proyecto, interesados y expertos en la materia puede servir para identificar riesgos. Las entrevistas son una de las principales fuentes de recopilación de datos para la Identificación de Riesgos.
- **Identificación de la causa:** es una investigación de las causas esenciales de los riesgos de un proyecto. Refina la definición del riesgo y permite agrupar los riesgos por causa. Se pueden desarrollar respuestas efectivas a los riesgos si se aborda la causa del riesgo.
- **Análisis de debilidades, amenazas, fortalezas y oportunidades (DAFO):** esta técnica asegura el examen del proyecto desde cada una de las perspectivas del análisis DAFO, para aumentar el espectro de los riesgos considerados.

1.5.3 Análisis mediante Lista de Control

Las Listas de Control para la Identificación de Riesgos pueden ser desarrolladas basándose en información histórica y en el conocimiento que ha sido acumulado de proyectos anteriores similares y de otras fuentes de información. Si bien una Lista de Control puede ser rápida y sencilla, es imposible elaborar una que sea exhaustiva, por tanto debe tenerse cuidado de explorar elementos que no aparecen en la misma. La Lista de Control debe revisarse durante el cierre del proyecto, a fin de mejorarla para su uso en futuros proyectos.

1.5.4 Análisis de Asunciones

Todos los proyectos se conciben y desarrollan sobre la base de un grupo de hipótesis, escenarios o asunciones. El análisis de asunciones es una herramienta que explora la validez de las asunciones según su aplicación en el proyecto. Identifica los riesgos del proyecto debido al carácter inexacto, inconsistente o incompleto de las asunciones.

1.5.5 Técnicas de Diagramación

Las técnicas de diagramación de riesgos pueden incluir:

- **Diagramas de causa y efecto:** estos diagramas también se conocen como diagramas de Ishikawa o de espina de pescado y son útiles para identificar las causas de los riesgos.
- **Diagramas de flujo o de sistemas:** estos diagramas muestran cómo se relacionan los diferentes elementos de un sistema y el mecanismo de causalidad.
- **Diagramas de influencias:** estos diagramas son representaciones gráficas de situaciones que muestran las influencias causales, la cronología de eventos y otras relaciones entre variables y resultados.

1.6 Salidas del proceso de Identificación de Riesgos

Estas técnicas dan como resultado algunas salidas, que se encuentran en un documento llamado Registro de Riesgos.

1.6.1 Registro de Riesgos

Las principales salidas de la Identificación de Riesgos son las entradas iniciales en el Registro de Riesgos, que se convierte en un componente del plan de gestión del proyecto. Éste al final contiene los resultados de los demás procesos de Gestión de Riesgos a medida que se llevan a cabo. La preparación del Registro

de Riesgos comienza en el proceso de Identificación de Riesgos y luego está disponible para la gestión de otros proyectos y otros procesos de Gestión de los Riesgos del Proyecto.

- **Lista de riesgos identificados:** se describen los riesgos identificados, incluidas las causas y las asunciones inciertas del proyecto.
- **Lista de posibles respuestas:** se pueden identificar posibles respuestas a un riesgo durante el proceso de Identificación de Riesgos. Estas respuestas, si son identificadas, pueden ser útiles como entradas al proceso Planificación de la Respuesta a los Riesgos.
- **Causas de los riesgos:** son las condiciones o eventos fundamentales que pueden dar lugar al riesgo identificado.
- **Categorías de riesgo actualizadas:** el proceso de identificar riesgos puede llevar a que se añadan nuevas categorías de riesgo a la lista de categorías. Es posible que la RBS (Estructura de Desglose del Riesgo) desarrollada en el proceso Planificación de la Gestión de Riesgos tenga que ser mejorada o modificada, basándose en los resultados del proceso Identificación de Riesgos.

1.7 Taxonomía del SEI para la Identificación de Riesgos

Las Taxonomías son las clasificaciones ordenadas de elementos de acuerdo a sus relaciones presumidas; pueden emplearse como herramientas de suma utilidad en diferentes ramas de la ciencia y la industria donde se pretende organizar y facilitar el acceso a un número importante de elementos que se encuentran mutuamente relacionados de alguna manera relevante.(WEBSTER, 2004)

La Gestión de Riesgos en base a Taxonomías implica utilizar una estructura agrupadora de los mismos de acuerdo a sus diferentes clases como una lista de consulta durante la actividad de Identificación de los Riesgos, esta lista estructurada puede obtenerse tanto de la misma organización como de cualquier otra fuente disponible.

El SEI es una de las instituciones que más aportes ha realizado al campo de la Gestión y la Identificación de Riesgos en proyectos de software. En (SEI, 2004) se plantea una taxonomía para la clasificación de riesgos según su afinidad, que posibilita que el proceso de identificación sea más organizado. Basado en la jerarquía y los grupos, el SEI propone también un cuestionario, que respondiendo a las clasificaciones presentadas, ayuda a los equipos a identificar riesgos potenciales en sus proyectos.

La taxonomía del SEI está estructurada en clases taxonómicas que se dividen en elementos y cada elemento se caracteriza por sus atributos, los cuales permiten clasificar el riesgo identificado. Los atributos serán seleccionados dependiendo del enfoque, estrategia de acción y la solución del problema. Las taxonomías del software están organizadas en tres clases.(MARVIN *et al.*, 1993)

Producto de la ingeniería: se refiere a los aspectos técnicos de los trabajos que deben realizarse, es decir, consiste en las actividades intelectuales y físicas necesarias que se necesitan para construir el producto deseado por el cliente teniendo en cuenta los siguientes elementos:

- **Requisitos:** definir lo que el producto de software debe hacer, satisfacer las necesidades de los clientes, además de tener en cuenta la viabilidad de desarrollo del producto y del esfuerzo realizado en el mismo.
- **Diseño:** la traducción de las necesidades en un diseño eficaz dentro del proyecto y las limitaciones operacionales.
- **Código y pruebas de unidad:** la traducción de los diseños de software en código que cumple los requisitos asignados a unidades individuales.
- **Integración y prueba:** la integración de las unidades en un sistema de trabajo y la validación que realiza el producto de software según las necesidades.
- **Especialidades de ingeniería:** los requisitos de los productos o las actividades de desarrollo que podrían requerir conocimientos especializados tales como la seguridad, disponibilidad y confiabilidad.

Entorno de desarrollo: se refiere a los métodos, procedimientos y herramientas utilizadas para producir el producto. Este entorno cuenta con los siguientes elementos:

- **Proceso de desarrollo:** la definición, planificación, documentación, la idoneidad, la observación y la comunicación de los métodos y procedimientos utilizados para desarrollar el producto.
- **Sistema de desarrollo:** las herramientas y el apoyo a los equipos utilizados en los productos de desarrollo, como la ayuda de la computadora, de las herramientas de ingeniería de software, simuladores, compiladores y sistemas de computación.

- **Gestión de procesos:** la planificación, el seguimiento y el control de los presupuestos; el control de los factores que intervienen en la definición, ejecución y pruebas del producto, la experiencia del líder de proyecto en el desarrollo del software.
- **Método de gestión:** los métodos, herramientas y equipo de apoyo que se utilizará para gestionar y controlar el desarrollo de productos, tales como instrumentos de supervisión, gestión de personal, control de calidad y gestión de la configuración.
- **Entorno de trabajo:** el ambiente general en el que se realiza el trabajo, incluyendo las actitudes de las personas, los niveles de cooperación, comunicación y la moral.

Limitaciones de programas: se refiere a los factores que están fuera del control directo del proyecto, pero pueden tener efectos importantes en su éxito. Se incluyen los siguientes elementos:

- **Recursos:** las limitaciones externas impuestas en el plazo previsto, el personal, presupuesto o instalaciones.
- **Contratos:** los términos y condiciones del contrato de proyecto.
- **Interfaces de programas:** las interfaces externas a los clientes, otros contratistas, de gestión de la empresa y los proveedores.

1.8 Aplicaciones informáticas de apoyo a la Gestión de Proyectos de Software

Desde los inicios de la Gestión de Proyectos surgieron aplicaciones que ayudaron a los directivos a la gestión de las tareas del personal del proyecto y a la contabilidad del mismo.

Muchas de estas herramientas gestionan proyectos de manera general y contienen módulos que permiten gestionar los riesgos. Un ejemplo de ello es el **DotProject**, que ofrece un marco completo para la planificación, gestión y seguimiento de múltiples proyectos para diferentes clientes, quienes pueden disponer de acceso para monitorizar la evolución del desarrollo.

Además hay otras herramientas de software que se ocupan solamente de la Gestión de Riesgos, específicamente, algunas de ellas orientan su funcionalidad al soporte de la **Identificación de Riesgos** y parcialmente, algunas pocas lo hacen empleando **Taxonomías**.

La siguiente tabla resume las características principales de las herramientas de software que se encuentran disponibles actualmente y que están relacionadas con la Identificación de Riesgos en proyectos:

Tabla 1: Resumen de herramientas de Gestión de Riesgos.

Producto	Proveedor	Descripción
Active Risk Manager (ARM)	Strategic Thought	Herramienta integrada de Gestión de Riesgos que brinda una solución para la Identificación de Riesgos mediante la utilización de la información contenida en la Estructura de Desglose del Trabajo(EDT) del proyecto.
Technical Risk Identification and Mitigation System (TRIMS)	Best Manufacturing Practices	Herramienta integrada de Gestión de Riesgos que emplea ingeniería de conocimientos y que se enfoca en la identificación y medición de riesgos técnicos de proyectos.
RiskTrak	Risk Services & Technology	Herramienta integrada de Gestión de Riesgos que brinda una solución para la Identificación de Riesgos mediante el empleo de bases de datos.
WelcomRisk	Welcom	Herramienta integrada de Gestión de Riesgos que brinda una solución para la identificación sistemática de riesgos mediante la utilización de una base de datos con categorías de riesgos propias.

1.9 Identificación de Riesgos en proyectos de Software Educativo y Multimedia en la UCI

Desde que la Universidad comenzó a dar sus primeros pasos en el desarrollo de proyectos ha tratado de indagar en el tema de la Gestión de Riesgos, en especial la Identificación de Riesgos, etapa en la cual se debe identificar y describir los riesgos de manera que se comprendan las situaciones de los mismos.

Como es natural, los proyectos de Software Educativo y Multimedia presentan riesgos particulares, genéricos a todo el Polo. El proceso de Identificación de Riesgos se realiza fundamentalmente siguiendo las Técnicas de Recopilación de Información. En la mayoría de estos proyectos no se utilizan herramientas que puedan almacenar los riesgos que han sido identificados, lo que trae como

consecuencia que no se tenga un registro genérico que permita ser utilizado en proyectos similares. Por tanto, la búsqueda de soluciones que ayuden a la Identificación de Riesgos es una prioridad. La utilización de una herramienta informática que ayude a mejorar este proceso pudiera ser muy útil y ventajosa.

Las herramientas estudiadas no brindan un soporte adecuado y estandarizado a los líderes de proyectos, específicamente durante la fase de Identificación de Riesgos para los proyectos de Software Educativo y Multimedia. Esto se debe fundamentalmente a que algunas no son libres, no usan taxonomías y además en su mayoría no son aplicaciones Web, lo que no favorece un ambiente colaborativo y de fácil acceso.

Estas aplicaciones o sólo se enfocan en una categoría de riesgo (Technical Risk Identification and Mitigation System), o tienen una amplia Base de Datos (Risk Trak y WelcomRisk) que les permite generar información de categorías propias de riesgos, o utilizan un mecanismo que no se orienta al uso de taxonomías (Active Risk Manager).

La siguiente tabla permite realizar una comparación entre las principales características de las herramientas analizadas.

Tabla 2: Comparación de herramientas de soporte de Identificación de Riesgos

Parámetros	ARM	TRIMS	Risk Trak	WelcomRisk
Uso de taxonomías	No	Si	No	No
Modelo completo	Si	Si	Si	Si
Interfaz gráfica	Si	Si	Si	Si
Reportes	Si	Si	Si	Si
Multiplataforma	Si	No	No	No

1.10 Propuesta de solución

Luego de un análisis de las herramientas que existen para la Gestión de Riesgos y las entrevistas realizadas a los grupos de desarrollo de los proyectos de Software Educativo y Multimedia, se comprobó la necesidad de desarrollar una aplicación Web que permita almacenar la experiencia de los proyectos con el objetivo de apoyar al proceso de Identificación de Riesgos. La aplicación empleará la Taxonomía del SEI para clasificar los riesgos.

1.11 Tendencias y tecnologías a utilizar

A continuación se describen algunas de las tendencias y tecnologías actuales útiles para el desarrollo de la propuesta de solución de manera eficiente, teniendo en cuenta las necesidades existentes y el entorno donde se aplicará el sistema que se va a construir.

1.11.1 Metodología de Desarrollo de Software

En la actualidad, con el avance tecnológico, es imprescindible el uso de metodologías de desarrollo, pues permiten a los desarrolladores un buen ambiente de trabajo y evitan de cierto modo que el proyecto fracase. Con el empleo de las metodologías, los equipos pueden elaborar un producto de alta calidad, que cumpla con los requisitos establecidos por el cliente, en el tiempo acordado y con el coste esperado.

Proceso Unificado de Desarrollo de Software (RUP): James Rumbaugh, Grady Booch e Ivar Jacobson, autores de “El proceso unificado de desarrollo de software”, opinan que “El problema del software se reduce a la dificultad que afrontan los desarrolladores para coordinar las múltiples cadenas de trabajo de un gran proyecto de software”.(RUMBAUHG *et al.*, 2000b)

El proceso unificado de desarrollo, RUP, es el resultado de la evolución e integración de diferentes metodologías de desarrollo de software. Permite sacar el máximo provecho de los conceptos asociados a la orientación a objetos y al modelado visual. Cuenta con las mejoras prácticas del modelo de desarrollo de un software en particular.(PRESSMAN, 2002; RUMBAUHG *et al.*, 2000a)

- Desarrollo de software de forma iterativa.
- Manejo de requerimientos.
- Utiliza arquitectura basada en componentes.
- Modela el software de forma visual, usando UML.
- Verifica la calidad del software.
- Controla los cambios.
- Dirige las tareas de cada desarrollador por separado y del equipo como un todo.
- Especifica los artefactos que deben desarrollarse en cada fase de desarrollo del software.

RUP se repite a lo largo de una serie de ciclos de desarrollo que constituyen la vida de un sistema, donde cada ciclo concluye con una versión del producto y consta de cuatro fases: Inicio, Elaboración,

Construcción y Transición. Todas las fases se subdividen en n-iteraciones y terminan con un hito. Está dividido en nueve flujos de trabajo, los cuales se pueden apreciar en la **Figura 1**.

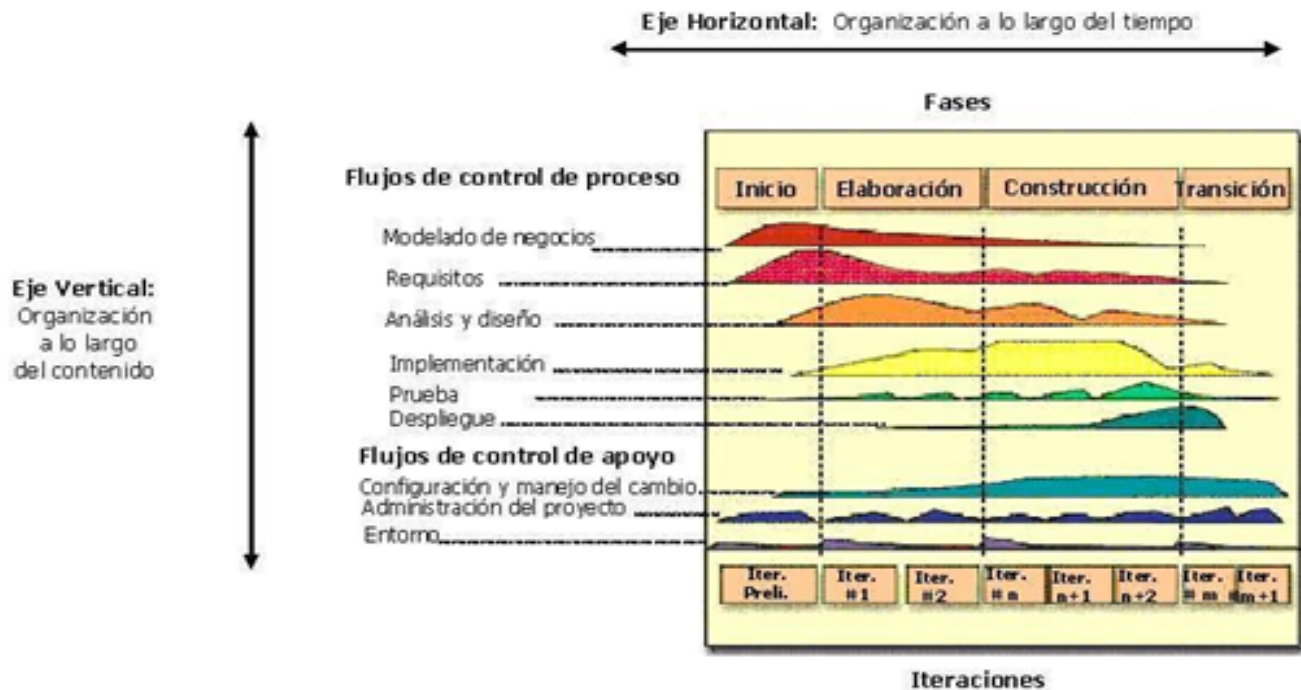


Figura 1: Disciplinas a través de las iteraciones.

En el caso específico de las aplicaciones Web, las actividades establecidas por dicho proceso son suficientes para cubrir todos los aspectos de los entornos de este tipo de aplicación. El Proceso Unificado Rational presenta una serie de características que lo convierten en único. (RUMBAUHG *et al.*, 2000a)

- Dirigido por casos de uso.
- Centrado en la Arquitectura.
- Iterativo e Incremental.

Esta metodología utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema de software. Es una parte esencial del Proceso Unificado de desarrollo de software y fueron desarrollados paralelamente por las mismas personas, haciendo que su integración sea un éxito.

Programación Extrema (XP): a diferencia de RUP como metodología pesada, la Programación Extrema (XP) es una metodología ágil que intenta reducir la complejidad del software por medio de un trabajo orientado directamente al objetivo, basado en las relaciones interpersonales y la velocidad de reacción. Esta metodología intenta minimizar el riesgo de fallo del proceso por medio de la disposición permanente de un representante competente del cliente a disposición del equipo de desarrollo. Dicho representante debe estar en condiciones de contestar rápida y correctamente cualquier pregunta del equipo de desarrollo de forma que no se retrase la toma de decisiones.(MOLPECERES, 2003)

XP define Historia de Usuario como base del software a desarrollar. A partir de las Historias de Usuarios y de la arquitectura perseguida se crea un plan de release entre el equipo de desarrollo y el cliente. Para cada release se discutirán los objetivos de la misma con el representante del cliente y se definirán las iteraciones (de pocas semanas de duración).(MOLPECERES, 2003)

Cada iteración tendrá como resultado un programa que se le entrega al cliente para que lo juzgue, si el cliente no está de acuerdo se definen las próximas iteraciones del proyecto y se someten a la aprobación del cliente, hasta que esté completamente de acuerdo y el software cumpla con todos sus requisitos. A diferencia de otros métodos, en XP la codificación pertenece al equipo completo, de forma que el conocimiento de la aplicación lo posea el equipo entero y no unos pocos miembros. (MOLPECERES, 2003)

En esta metodología se sigue un diseño evolutivo con la siguiente premisa: conseguir la funcionalidad deseada de la forma más sencilla posible. Este diseño hace que apenas se le dé importancia al análisis como fase independiente, debido a que se trabaja exclusivamente en función de las necesidades del momento.(MOLPECERES, 2003)

Después de realizar un estudio de estas metodologías, se ha llegado a la conclusión de que la metodología a utilizar para el desarrollo de la aplicación Web propuesta es RUP, debido a que abarca todo el ciclo de vida del software de manera organizada, dividiéndolo en fases, en las cuales se realizan varias iteraciones que traen consigo que se vaya desarrollando el producto de forma incremental, genera gran cantidad de artefactos y se centra en la arquitectura. Además posibilita reducir los riesgos y hace que el proyecto se torne más predecible.

1.11.2 Lenguaje Unificado de Modelado (UML)

El Lenguaje Unificado de Modelado (UML) fue adquirido en 1997 por la Object Management Group (OMG) para luego convertirse en un estándar de facto, que permite visualizar, especificar y documentar modelos que se crean durante la aplicación de un proceso de software.

UML nació como una notación estándar de la construcción de modelos. Es un sistema notacional (que entre otras cosas, incluye el significado de sus notaciones) destinado a los sistemas de modelado que utilizan conceptos orientados a objetos (LARMAN, 1999). El mismo no es una metodología, sino que está asociado a un proceso de desarrollo en particular. Los modelos que se representan a través de UML se pueden agrupar en tres tipos diferentes:

Tabla 3: Tipos de modelos en UML

Modelo	Descripción
Funcional	Diagramas de casos de uso (CU), que describen el sistema desde la perspectiva del usuario.
Objeto	Diagramas de clases, que describen la estructura de un sistema en términos de objetos, atributos, asociaciones y operaciones.
Dinámico	Diagramas de interacción y de estados, que describen el comportamiento del sistema.

Los elementos de UML se visualizan por medio de diagramas que muestran diferentes vistas y cada uno en particular representa un aspecto del sistema. Es un lenguaje consolidado, fácil de aprender y permite una comunicación fluida entre los diferentes actores del modelo.

UML permite a los usuarios entender la realidad desde un punto de vista de la tecnología y la posibilidad de que reflexione antes de invertir y gastar grandes cantidades de dinero en proyectos que no estén seguros en su desarrollo, reduciendo el costo y el tiempo empleado en la construcción de los módulos que construirán el software. (RUMBAUHG *et al.*, 2000a)

El propósito general de este lenguaje es el modelado orientado a objetos (OO). Además, combina notaciones provenientes desde el Modelado OO, Modelado de componentes, Modelado de datos, Modelado de Workflows. Este lenguaje de modelado ha causado un gran impacto en las comunidades de

software gracias al conjunto de características que presenta, además permite ser utilizado en la elaboración de cualquier dominio y con diferentes tamaños.

Se utilizará como notación el Lenguaje Unificado de Modelado para lograr un mayor entendimiento ya que permite modelar y describir secuencialmente por pasos todos los procesos que se llevan a cabo. Es considerado muy útil porque es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema de software.

1.11.3 Herramienta CASE de Desarrollo de Software

Actualmente, se ha hecho habitual que cada vez que se desarrolle un software por pequeño o grande que sea, se utilicen las herramientas CASE, con el fin de automatizar los aspectos claves de todo el proceso de desarrollo de un sistema.

Estas herramientas son de gran utilidad pues intervienen en todos los aspectos del ciclo de vida del desarrollo del software, permitiendo a los desarrolladores menos complejidad a la hora de realizar diferentes actividades del proceso de software como son la realización del diseño del proyecto, la documentación o detección de errores. Algunos de los objetivos de las herramientas CASE son:

- Mejorar la productividad en el desarrollo y mantenimiento del software.
- Aumentar la calidad del software en construcción.
- Perfeccionar la planificación del proyecto.

Rational Rose: es la herramienta CASE que comercializan los desarrolladores de UML (Booch, Rumbaugh y Jacobson) y que soporta de forma completa la especificación del UML. Esta herramienta propone la utilización de cuatro tipos de modelos para realizar un diseño del sistema, utilizando una vista estática, otra dinámica de los modelos del sistema, una lógica y otra física; que permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software (LARMAN, 1999). A continuación se listan los tipos de modelos:

- Desarrollo Iterativo.
- Generador de Código.
- Ingeniería Inversa.
- Trabajo en Grupo.

Visual Paradigm: es una herramienta multiplataforma para el modelado de aplicaciones, sobre todo en proyectos donde se vaya a aplicar intensivamente los conceptos avanzados de orientación a objeto (CANALES MORA, 2004). Además facilita a los ingenieros de software diseñar, integrar y modelar visualmente los distintos diagramas que se generan a lo largo del desarrollo del software. Presenta un generador de código que soporta más de 10 lenguajes y proporciona la ingeniería inversa (PARADIGM, 2007), ejemplos de lenguajes que soporta se mencionan a continuación:

- Java
- C++
- PHP

Después de hacer un estudio de estas dos herramientas, se llegó a la conclusión de que la herramienta a utilizar para un buen desarrollo ingenieril de esta investigación es Rational Rose Enterprise Edition, teniendo en cuenta sus potencialidades para el modelado visual. Utiliza la notación estándar en la arquitectura de software (UML), la cual permite a los arquitectos de software y desarrolladores visualizar el sistema completo utilizando un lenguaje común para comprender y comunicar la estructura y la funcionalidad del sistema en construcción. Además permite la ingeniería inversa y brinda soporte a la metodología de software seleccionada, por lo que se obtendrá una mayor calidad en el desarrollo del software.

1.11.4 Fundamentación del lenguaje a utilizar

Con el nacimiento de la Word Wide Web y la necesidad de intercambiar información con otras partes del mundo, surgieron una serie de lenguajes de programación Web para darle un mayor dinamismo y funcionalidad a las expectativas de diferentes tipos de clientes. Existen clasificaciones según los tipos de lenguajes, libres y propietarios. Para el desarrollo de la aplicación fue seleccionado el lenguaje PHP, el cual es libre y fácil de usar, además se ha convertido en estos últimos años en uno de los pilares dentro de los lenguajes de programación y a su vez, en uno de los más explotados por los programadores a nivel mundial.

PHP (acrónimo “Hipertext Pre-processor”) es un lenguaje de alto nivel embebido en páginas HTML y ejecutado en un servidor (MOORE y FOX, 2006). La mayor parte de su sintaxis ha sido tomada de C, Java

y Perl con algunas características específicas de sí misma. La meta del lenguaje es permitir rápidamente a la comunidad de desarrolladores la generación dinámica de páginas Web.(ADRFORMACIÓN.COM, 2004)

Características del lenguaje a utilizar

Al ser un lenguaje libre, dispone de una gran cantidad de características que lo convierte en la herramienta ideal para la creación de páginas Web dinámicas. (RODAS HINOSTROZA, 2007)

- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQLServer, Sybase mSQL, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF y analizar código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación.
- Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.

Ventajas que proporciona la utilización del lenguaje

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de Base de Datos que se utilizan en la actualidad, se destaca su conectividad con MySQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos llamados ext's o extensiones.
- Posee una amplia documentación de ayuda para un mayor entendimiento del lenguaje.
- Soporte sólido para Programación Orientada a Objetos.

1.11.5 Sistemas Gestores de Bases de Datos

Los Sistemas de Gestión de Bases de Datos (SGBD) se pueden definir como el conjunto de herramientas que suministra a todos los administradores, analistas, programadores y usuarios los medios necesarios para describir, recuperar y manipular los datos almacenados en la Bases de Datos(BD), manteniendo la seguridad, integridad y confidencialidad de los mismos.

Fundamentación del Sistema Gestor de Base de Datos a utilizar (MySQL)

MySQL es un sistema administrador de bases de datos relacionales (SGBDR), creado por la compañía MySQL AB fundada en 1995. Se trata de un SGBD multihilo y multiusuario, cuyo principal objetivo de diseño fue la velocidad (ofrece alto rendimiento debido a su rapidez de respuesta).

Provee múltiples motores de almacenamiento (MyISAM, Merge, InnoDB, BDB, Memory/heap, MySQL Cluster, Federated, Archive, CSV, Blackhole. Además, permite la agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar su número por segundo. Posibilita conexiones entre diferentes computadoras con distintos sistemas operativos y una integración perfecta con PHP.

La licencia GPL de MySQL obliga a distribuir cualquier producto derivado (aplicación) bajo esa misma licencia. Por tanto MySQL tiene sus restricciones: sólo es gratis si se está dispuesto a distribuir la aplicación que se quiere desarrollar bajo esa misma licencia GPL. Si se desea distribuir la aplicación comercialmente, entonces se debe pagar la licencia comercial de MySQL. (GILFILLAN, 2003)

Características del Sistema Gestor de Base de Datos a utilizar

- Un amplio subconjunto de ANSI SQL 99 y varias extensiones.
- Soporte a multiplataforma.
- Procedimientos almacenados.
- Vistas actualizables.
- Soporte a VARCHAR21.
- Motores de almacenamiento independientes (MyISAM²² para lecturas rápidas, InnoDB para transacciones e integridad referencial).
- Transacciones con los motores de almacenamiento InnoDB, BDB y Cluster; puntos de recuperación (savepoints) con InnoDB.

Ventajas que proporciona la utilización del Sistema Gestor de Base de Datos

- Lo mejor de MySQL es su velocidad a la hora de realizar las operaciones, que lo convierte en uno de los gestores que ofrecen mayor rendimiento.
- Su bajo consumo de recursos del sistema lo declara apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.

- Las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación.
- Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en que los errores no se produzcan en el propio gestor, sino en el sistema en el que está.

1.11.6 Entorno de Desarrollo

Un Entorno de Desarrollo Integrado (IDE, en inglés, Integrated Development Environment) no es más que un programa compuesto por un conjunto de herramientas para un programador, este puede ser un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los mismos proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación y cabe la posibilidad que un mismo IDE pueda funcionar con varios lenguajes de programación.

Zend Studio como IDE de Desarrollo

Este es un software empleado por los programadores para la programación Web, 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. Consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. Permite hacer depuraciones simples de scripts (ALVAREZ, 2006). Dentro de sus ventajas se encuentran las siguientes:

- Contiene una ayuda contextual con todas las librerías de funciones del lenguaje que asiste en todo momento ofreciendo nombres de las funciones y parámetros que deben recibir.
- Completamiento de código.
- Identificación de errores en el código.

1.12 Conclusiones parciales

- Los conceptos asociados al objeto de estudio son indispensables para comprender realmente el problema descrito y tener una mayor claridad para orientar la investigación.
- La Identificación de Riesgos es un proceso importante y necesario que todo proyecto debe tener en cuenta para evitar cuantiosos daños.

- Las aplicaciones informáticas existentes para la Gestión de Riesgos no brindan el soporte adecuado, específicamente durante la fase de Identificación de Riesgos en los proyectos de Software Educativo y Multimedia.
- El lenguaje de programación PHP es viable para la implementación de la aplicación Web.
- La metodología RUP representa una guía óptima para el desarrollo de la herramienta propuesta.

Capítulo 2

Características del Sistema

2.1 Introducción

En el presente capítulo se presenta el Modelo de Dominio como alternativa al Modelo de Negocio debido a la poca estructuración de los procesos que describen el negocio. El objetivo principal es contribuir a la comprensión del contexto del sistema y por tanto de los requisitos funcionales que se desprenden del mismo. Se identifican cuáles son los requisitos funcionales y los no funcionales del sistema, además de obtener y describir los casos de uso que guiarán la solución propuesta, siendo de gran ayuda la utilización de la herramienta Case Rational Rose Enterprise Edition, que asiste al desarrollo del software para obtener mayor calidad.

2.2 Modelo de Dominio

2.2.1 Descripción del Modelo de Dominio

Realizando un análisis exhaustivo del problema, se llega a la conclusión de que el negocio tiene un bajo nivel de estructuración, donde no se definen concretamente los procesos del mismo, de ahí que se decide dar un nuevo enfoque a todo el proceso; para ello se utiliza un Modelo del Dominio, que permite de manera visual mostrar al usuario los principales conceptos que se manejan en el dominio de la aplicación en desarrollo. Este modelo contribuye posteriormente a identificar algunas clases que se utilizarán en el sistema, lo que ayuda a los clientes y desarrolladores a utilizar un vocabulario común con el que puedan entender el contexto en que se emplaza el sistema.

Por tanto es necesario tener un gran conocimiento del funcionamiento del objeto de estudio, con el objetivo de realizar una correcta captura de los requisitos y construir un sistema que cumpla con las necesidades del cliente. Para esto se exponen todos los conceptos que intervienen en el mismo:

Proyecto: es la entidad que contiene el conjunto de actividades que son realizadas por sus integrantes y que se encuentran interrelacionadas y coordinadas.

- Gestor de riesgos:** es el encargado de realizar el proceso de Identificación de Riesgos en los proyectos.
- Identificación de Riesgos:** Proceso en el cual se identifican los riesgos y factores de riesgos de un proyecto.
- Riesgos:** amenazas que afectan a los objetivos del proyecto.
- Factores de riesgos:** son los eventos o factores que incrementan la posibilidad de ocurrencia de un riesgo.
- Taxonomías:** son clasificaciones ordenadas de elementos.
- Impacto:** consecuencia que provoca el riesgo.
- Probabilidad de disparo:** posibilidad de que se dispare un factor de riesgo.
- Entorno:** condiciones de trabajo en las cuales se desarrolla el proyecto.

2.2.2 Diagrama del Modelo de Dominio

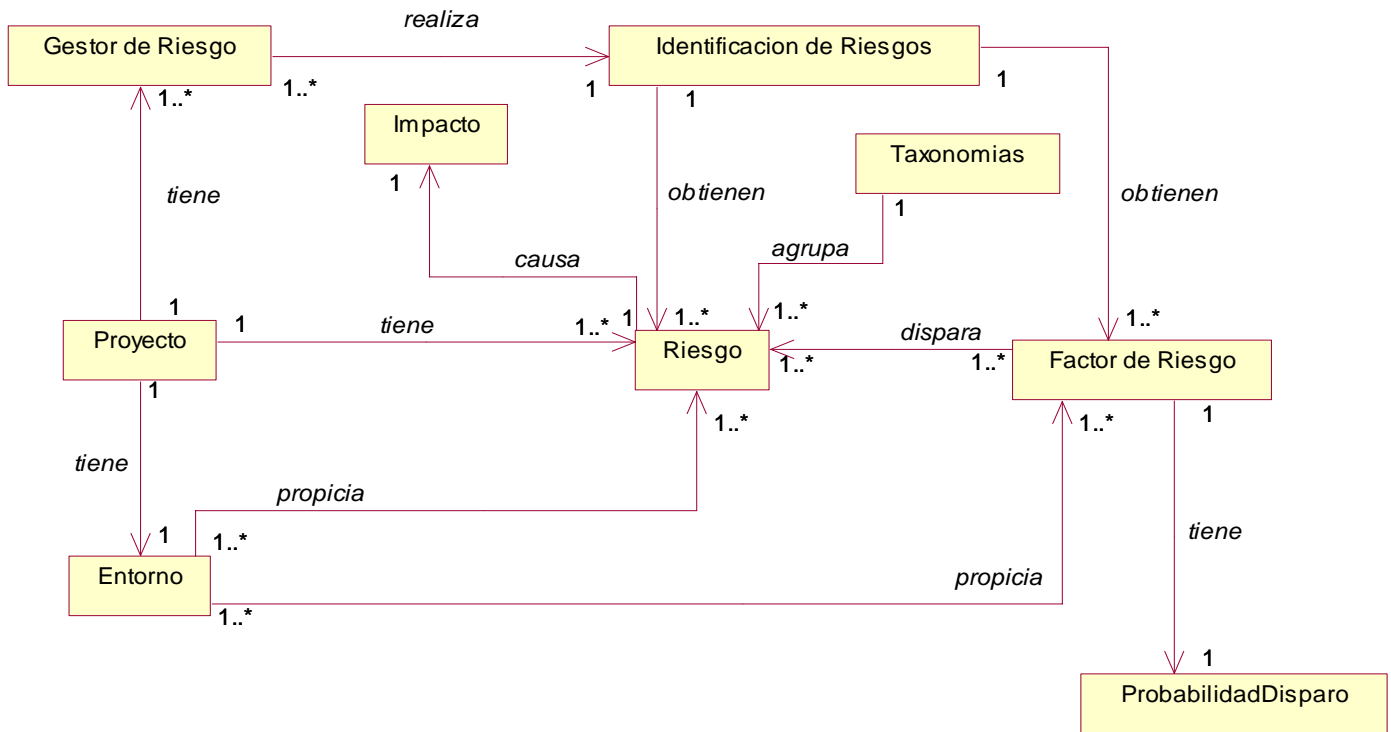


Figura 2: Modelo de Dominio.

2.3 Requisitos del sistema

Luego de conocer los conceptos asociados al objeto de estudio del problema se comienza a modelar el sistema que se va construir, además se analiza, ¿qué debe hacer el sistema para que se cumplan los objetivos planteados?. Para lo mismo se identifican los requisitos funcionales (RF), los no funcionales (RNF) y se modelan los RF en representaciones de casos de uso del sistema.

2.3.1 Definición de los requisitos funcionales

Los requerimientos o requisitos funcionales, son capacidades o condiciones que el sistema debe cumplir, es decir, define lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas. Con éstos se pretende determinar de manera más clara y concisa lo que debe hacer el sistema siguiendo un enfoque funcional. A continuación se presentan las funcionalidades que debe cumplir la aplicación:

RF1 Gestionar riesgos.

RF1.1 Insertar riesgos.

RF1.2 Eliminar riesgos.

RF1.3 Modificar riesgos.

RF2 Gestionar factor de riesgo.

RF2.1 Insertar factor de riesgo.

RF2.2 Modificar factor de riesgo.

RF2.3 Eliminar factor de riesgo.

RF3 Gestionar entorno de trabajo.

RF3.1 Adicionar entorno de trabajo.

RF3.2 Eliminar entorno de trabajo.

RF3.3 Modificar entorno de trabajo.

RF4 Gestionar valor del entorno de trabajo.

RF4.1 Insertar valor del entorno de trabajo.

RF4.2 Modificar valor del entorno de trabajo.

RF4.3 Eliminar valor del entorno de trabajo.

RF5 Gestionar relación entre riesgo y factor de riesgo.

RF5.1 Insertar relación riesgo_factor riesgo.

RF5.2 Eliminar relación riesgo_factor riesgo.

RF5.3 Modificar relación riesgo_factor riesgo.

RF6 Gestionar relación entre riesgo y entorno de trabajo.

RF6.1 Insertar relación riesgo_entorno trabajo.

RF6.2 Eliminar relación riesgo_entorno trabajo.

RF6.3 Modificar relación riesgo_entorno trabajo.

RF7 Gestionar relación entre factor de riesgo y entorno de trabajo.

RF7.1 Insertar relación factor riesgo_entorno trabajo.

RF7.2 Eliminar relación factor riesgo_entorno trabajo.

RF7.3 Modificar relación factor riesgo_entorno trabajo.

RF8 Gestionar usuarios.

RF8.1 Registrar usuarios.

RF8.2 Eliminar usuarios.

RF8.3 Modificar usuarios.

RF8.4 Asignar permisos a usuarios.

RF9 Mostrar reportes.

RF9.1 Mostrar reportes de riesgos.

RF9.2 Mostrar reportes de factores de riesgos.

RF9.3 Mostrar reportes del entorno de trabajo.

RF9.4 Exportar reportes a PDF.

RF10 Notificar disparo de un riesgo.

RF11 Notificar disparo de un factor de riesgo.

RF12 Autenticar usuario.

2.3.2 Definición de los requisitos no funcionales

Los requerimientos o requisitos no funcionales son propiedades o cualidades que el producto debe tener, que posibilitan que sea más atractivo, usable, rápido, confiable entre otras características. También, puede ser alguna restricción que éste debe tener para cumplir una determinada funcionalidad. A continuación se presentan los requisitos no funcionales:

Apariencia o interfaz externa:

- Diseño sencillo y fácil de usar, permitiendo que no sea necesario mucho entrenamiento para utilizar el sistema.

Usabilidad:

- La aplicación podrá ser utilizada por cualquier persona que posea conocimientos básicos de informática.

Portabilidad:

- La herramienta debe ser capaz de ejecutarse sobre la mayoría de los sistemas operativos que existen.

Diseño e Implementación:

- La aplicación será desarrollada utilizando lenguaje de programación PHP.
- La aplicación no deberá hacer uso de ningún framework del lado del servidor.
- La interfaz deberá ser separada de manera total del resto de la aplicación de forma tal que ésta última pueda ser reutilizada en otras aplicaciones con facilidad.

Seguridad:

Confidencialidad

- La información manejada por el sistema debe estar protegida de acceso no autorizado.
- Garantizar que la información sea vista únicamente por los usuarios autenticados en el sistema según su rol.

Integridad

- Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.

Software:

Cliente

- Navegador Mozilla Firefox (3.0.10 o superior) o Internet Explorer (7.0 o superior).

Servidor

- MySQL (mayor que 4.0).
- Apache Web Server (mayor que 2.0).
- Dotproject 2.x.

Hardware:

Servidor

- 60 MB de espacio libre en disco duro.

Escalabilidad:

- Se debe diseñar una arquitectura capaz de asimilar nuevos mecanismos encargados de ayudar a la identificación automatizada de los riesgos basada en la base de conocimientos.
- El sistema debe permitir su introducción de manera sencilla en otras herramientas de Gestión de Proyectos, con un mínimo de cambios.

2.4 Propuesta de solución

Haciendo uso de las habilidades y facilidades que brinda UML, luego de realizar la captura de los requisitos funcionales (RF) del sistema, se representarán a través del diagrama de casos de uso. Teniendo en cuenta lo definido anteriormente, se identifican cuáles serán los actores que van a interactuar con el sistema y los casos de uso que van a representar las diferentes funcionalidades.

2.4.1 Actores del sistema

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

Tabla 4: Descripción de los actores del sistema.

Actor	Descripción
Administrador	Es la persona que se encarga de administrar todo el sistema informático, creando cuentas de administradores. Además, tiene acceso pleno a la información que maneja el sistema.
Líder de proyecto	Esta persona es la que interactúa con la aplicación ya sea solicitando algún reporte o gestionando información, puede gestionar usuarios y líderes de su proyecto; es la mayor responsable de clasificar correctamente los riesgos.
Usuario	Es la persona que pretende ingresar al sistema. Si el proceso de autenticación es correcto, recibe permisos en dependencia del rol que desempeña. El ingreso al sistema es mediante el usuario y contraseña. Se beneficia de cierta forma con la información que está almacenada en el sistema, puede solicitar reportes generales o específicos y exportar estos reportes a formato PDF.

2.4.2 Diagrama de casos de uso del sistema

Los casos de uso del sistema (CUS) son un conjunto de secuencia de acciones que un sistema ejecuta y que produce un resultado observable para un actor. Con otras palabras, son fragmentos de funcionalidad que el sistema ofrece a los actores que interactúan con el mismo, reportándoles beneficios. A continuación se muestra el diagrama de casos de uso del sistema:

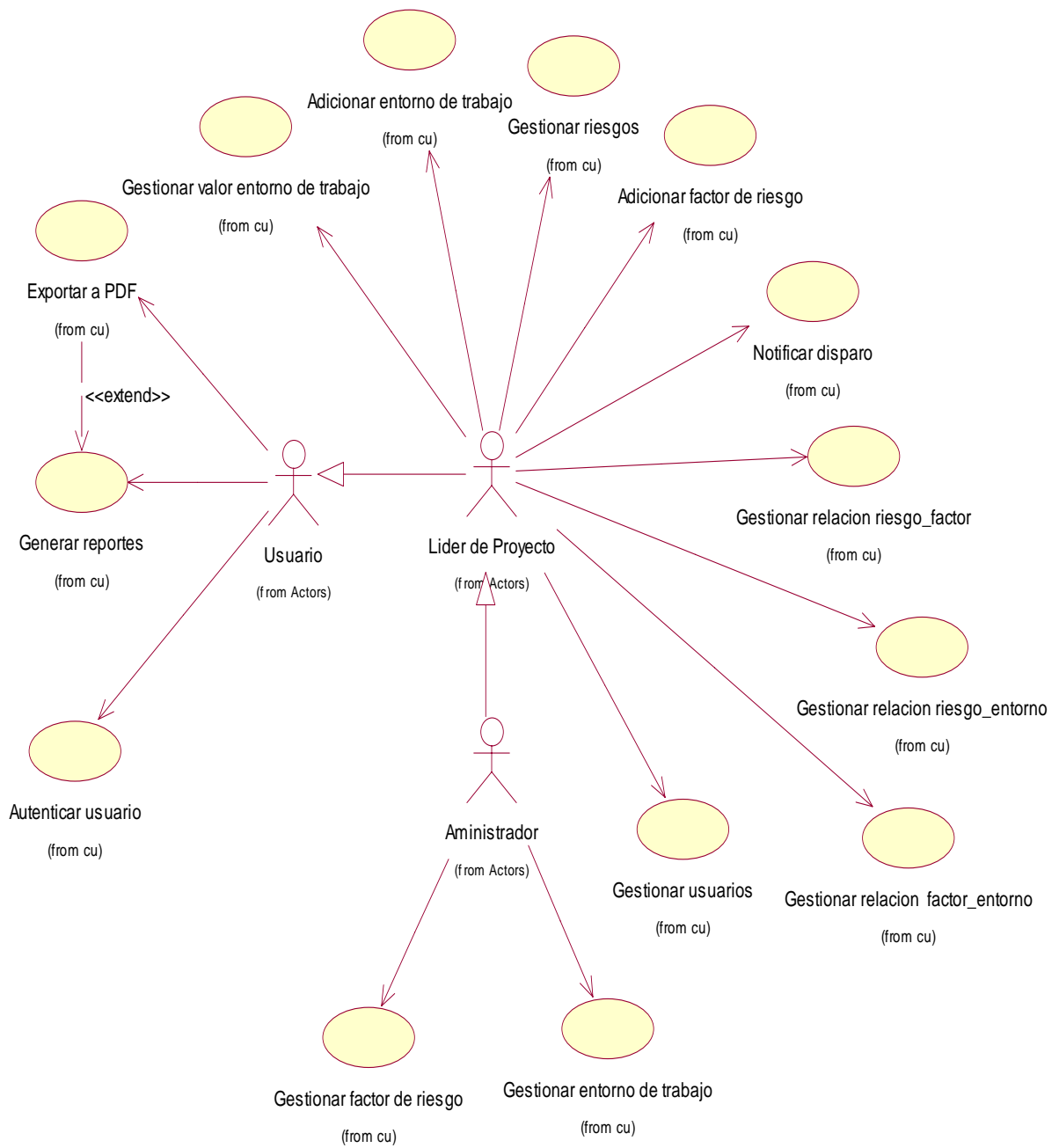


Figura 3: Diagrama de casos de uso del sistema.

2.4.3 Descripción de los casos de uso del sistema

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

Tabla 5: Caso de uso del sistema<Gestionar riesgos>

Caso de uso	Gestionar riesgos	
Actor	Líder de proyecto	
Propósito	Permite al líder de proyecto gestionar (Insertar, Modificar, Eliminar) los riesgos del proyecto.	
Resumen	El CUS se inicia cuando el líder de proyecto selecciona la opción Riesgos, luego se selecciona el tipo de gestión, se introducen los datos necesarios, el sistema realiza la acción seleccionada y termina el CU.	
Referencia	R1	
CU asociados		
Precondiciones	El usuario con el rol de líder de proyecto se debe haber autenticado con anterioridad.	
Poscondiciones	<ul style="list-style-type: none"> •Riesgo adicionado a la Base de Datos. •Riesgo modificado en la Base de Datos. •Riesgo eliminado de la Base de Datos. 	
Flujo Normal de Eventos		
Acción del actor	Respuesta del sistema	
1-El líder de proyecto selecciona la opción Riesgos.	1.1-El sistema muestra los riesgos en dependencia del proyecto que esté y las siguientes opciones: <ul style="list-style-type: none"> ▪ Insertar Riesgos ▪ Modificar Riesgos ▪ Eliminar Riesgos 	

Sección 1: “Insertar riesgos”	
Acción del actor	Respuesta del sistema
1-El líder de proyecto selecciona la opción Insertar Riesgos.	<p>1.1-El sistema muestra un formulario donde se muestran los campos necesarios para que el líder de proyecto registre los datos.</p> <ul style="list-style-type: none"> • Nombre • Descripción • Descripción del impacto • Clasificación • Probabilidad • Efecto
2-El líder de proyecto inserta los datos en el formulario.	<p>2.1-El sistema valida que todos los campos estén llenos.</p> <p>2.2-El sistema verifica que el riesgo no esté registrado en la Base de Datos.</p> <p>2.3-El sistema guarda todos los datos entrados por el líder de proyecto y termina el caso de uso.</p>
Flujos Alternos	
Cursos Alternos	<p>2.1-Se emite un mensaje “Debe introducir los datos”, con el objetivo de que llene los campos obligatorios y retorna a la acción 2.</p> <p>2.2-Si el riesgo existe el sistema muestra un mensaje de alerta y retorna a la acción 2.</p>
Sección 2: “Modificar riesgos”	
Acción del actor	Respuesta del sistema
1-El líder de proyecto selecciona el riesgo que va modificar.	1.1-El sistema localiza los datos del riesgo y los muestra en un formulario.

2-El líder de proyecto realiza las modificaciones deseadas.	2.1-Se verifica que todos los campos obligatorios estén llenos. 2.2-Se actualiza la información entrada por el líder de proyecto y finaliza el caso de uso.
Flujos Alternos	
Cursos Alternos	2.1-En caso de que no se llenen los campos que son obligatorios, se emite el mensaje: “Debe introducir los datos” y retorna a la acción 2.
Sección 3: “Eliminar riesgos”	
Acción del actor	Respuesta del sistema
1-El líder de proyecto selecciona el riesgo a eliminar.	1.1-El sistema localiza los datos del riesgo seleccionado y muestra un mensaje de advertencia para la acción a realizar.
2-El líder de proyecto confirma la eliminación del riesgo.	2.1-El sistema procede a la eliminación de los datos del riesgo seleccionado y termina el CU.
Flujos Alternos	
Cursos Alternos	2.1-Si el líder de proyecto cancela la acción se culmina el CU sin ejecutar ninguna acción.
Prioridad	Crítico

Tabla 6: Caso de uso del sistema<Adicionar factor de riesgo>

Caso de uso	Adicionar factor de riesgo
Actor	Líder de proyecto
Propósito	Permite al líder de proyecto adicionar los factores de riesgos.
Resumen	El CUS se inicia cuando el líder de proyecto selecciona la opción de Factor de Riesgo, se introducen los datos necesarios, el sistema realiza la acción seleccionada y termina el CU.

Referencia	R2
CU asociados	
Precondiciones	El usuario con el rol de líder de proyecto se debe haber autenticado con anterioridad.
Poscondiciones	<ul style="list-style-type: none"> •Factor de riesgo adicionado a la Base de Datos.
Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema
1-El líder de proyecto selecciona la opción Factor de Riesgo.	1.1-El sistema muestra el listado de los factores de riesgos.
2-El líder de proyecto selecciona la opción Insertar Factor de Riesgo.	2.1-El sistema muestra un formulario donde son mostrados los campos necesarios para que el líder de proyecto registre los datos. <ul style="list-style-type: none"> • Nombre • Descripción
3-El líder de proyecto introduce los datos.	3.1-El sistema verifica que los campos estén llenos. 3.2-El sistema verifica que el factor de riesgo no esté registrado en la Base de Datos. 3.3-El sistema almacena todos los datos entrados por el líder de proyecto y finaliza el caso de uso.
Flujos Alternos	
Cursos Alternos	3.1-Se emite un mensaje “Debe introducir los datos” con el objetivo de que llene los campos obligatorios y retorna a la acción 2. 3.2-Si el factor de riesgo existe el sistema muestra un mensaje de alerta y retorna a la acción 2.
Prioridad	Crítico

Tabla 7: Caso de uso del sistema <Gestionar factor de riesgo>

Caso de uso	Gestionar factor de riesgo	
Actor	Administrador	
Propósito	Permite al administrador modificar y eliminar los factores de riesgos.	
Resumen	El CUS se inicia cuando el administrador selecciona la opción Factor de Riesgo, luego se selecciona el tipo de gestión, se introducen los datos necesarios, el sistema realiza la acción seleccionada y termina el CU.	
Referencia	R2	
CU asociados		
Precondiciones	El usuario con el rol de administrador se debe haber autenticado con anterioridad.	
Poscondiciones	<ul style="list-style-type: none"> • Factor de riesgo modificado en la Base de Datos. • Factor de riesgo eliminado de la Base de Datos. 	
Flujo Normal de Eventos		
Acción del actor	Respuesta del sistema	
1-El administrador selecciona la opción Factor de Riesgo.	1.1-El sistema muestra el listado de los factores de riesgos y las siguientes opciones: <ul style="list-style-type: none"> ▪ Modificar factor de riesgo. ▪ Eliminar factor de riesgo. 	
Sección 1: “Modificar factor de riesgo”		
Acción del actor	Respuesta del sistema	
1-El administrador selecciona el factor de riesgo a modificar.	1.1-El sistema localiza los datos y los muestra.	
2-El administrador realiza las modificaciones deseadas.	2.1-Se verifica que todos los campos obligatorios estén llenos. 2.2-Se actualiza la información entrada por el administrador y finaliza el caso de uso.	
Flujos Alternos		

Cursos Alternos	2.1-En caso de que no se llenen los campos que son obligatorios, se emite el mensaje: “Debe introducir los datos” y retorna a la acción 2.
Sección 2: “Eliminar factor de riesgo”	
Acción del actor	Respuesta del sistema
1-El administrador selecciona un factor de riesgo de la lista.	1.1-El sistema localiza los datos del factor de riesgo seleccionado y muestra un mensaje de advertencia para la acción a realizar.
2-El administrador confirma la eliminación del factor de riesgo.	2.1-El sistema procede a la eliminación de los datos del factor de riesgo seleccionado y culmina el CU.
Flujos Alternos	
Cursos Alternos	2.1-Si el administrador cancela la acción se culmina el CU sin ejecutar ninguna acción.
Prioridad	Crítico

Tabla 8: Caso de uso del sistema < Adicionar entorno de trabajo >

Caso de uso	Adicionar entorno de trabajo
Actor	Líder de proyecto
Propósito	Permite al líder de proyecto adicionar un nuevo entorno de trabajo.
Resumen	El CUS se inicia cuando el líder de proyecto selecciona la opción Entorno de Trabajo, el sistema realiza la acción seleccionada y finaliza el CU.
Referencia	R3
CU asociados	
Precondiciones	El líder de proyecto se debe haber autenticado con anterioridad.
Poscondiciones	<ul style="list-style-type: none"> • Entorno de trabajo adicionado a la Base de Datos.
Flujo Normal de Eventos	

Acción del Actor	Respuesta del sistema
1-El líder de proyecto selecciona la opción Entorno de Trabajo.	1.1-El sistema muestra un campo para que introduzca un nuevo entorno de trabajo.
2-El líder de proyecto llena los campos mostrados.	2.1-El sistema verifica que los campos estén llenos. 2.2-El sistema guarda los datos y termina el caso de uso.
Flujos Alternos	
Cursos Alternos	2.1-En caso de que no se llenen los campos obligatorios, se emite el mensaje: “Debe introducir los datos” y retorna a la acción 2.
Prioridad	Secundario

Tabla 9: Caso de uso del sistema<Gestionar entorno de trabajo>

Caso de uso	Gestionar entorno de trabajo
Actor	Administrador
Propósito	Permite al administrador modificar y eliminar un entorno de trabajo.
Resumen	El CUS se inicia cuando el líder de proyecto selecciona la opción Entorno de Trabajo, luego selecciona lo que desea hacer, el sistema realiza la acción seleccionada y culmina el CU.
Referencia	R3
CU asociados	
Precondiciones	El usuario con el rol de administrador se debe haber autenticado con anterioridad.
Poscondiciones	<ul style="list-style-type: none"> • Entorno de trabajo eliminado de la Base de Datos. • Entorno de trabajo modificado en la Base de Datos.
Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema

1-Selecciona la opción de Entorno de Trabajo.	1.1-El sistema brinda la posibilidad de ver el listado de los entornos de trabajo y las opciones: <ul style="list-style-type: none"> • Eliminar entorno de trabajo. • Modificar entorno de trabajo.
Sección 1: “Eliminar entorno de trabajo”	
Acción del actor	Respuesta del sistema
1-Selecciona el entorno de trabajo que desea eliminar.	1.1-El sistema localiza los datos y muestra un mensaje de confirmación de eliminación.
2-El administrador confirma la eliminación.	2.1-Se procede a la eliminación de los datos y termina el caso de uso.
Flujos Alternos	
Cursos Alternos	2.1-Si cancela la acción el CUS culmina sin ejecutar ninguna acción.
Sección 2: “Modificar entorno de trabajo”	
Acción del actor	Respuesta del sistema
1-Selecciona el entorno de trabajo que desea modificar.	1.1-El sistema localiza los datos y los muestra en un formulario.
2-El administrador realiza las modificaciones deseadas.	2.1-Se verifica que los campos obligatorios estén llenos. 2.2-Se actualiza la información entrada por el administrador y finaliza el caso de uso.
Flujos Alternos	
Cursos Alternos	2.1-En caso de que no se llenen los campos obligatorios, se emite el mensaje: “Debe introducir los datos” y retorna a la acción 2.
Prioridad	Secundario

Tabla 10: Caso de uso del sistema < Gestionar valor entorno de trabajo >

Caso de uso	Gestionar valor entorno de trabajo	
Actor	Líder de proyecto	
Propósito	Permite al líder de proyecto gestionar (Insertar, Modificar, Eliminar) los valores del entorno de trabajo.	
Resumen	El CUS se inicia cuando el líder de proyecto selecciona la opción Entorno de Trabajo, luego selecciona el tipo de gestión, el sistema realiza la acción seleccionada y termina el CU.	
Referencia	R4	
CU asociados		
Precondiciones	El líder de proyecto se debe haber autenticado con anterioridad.	
Poscondiciones	<ul style="list-style-type: none"> • El valor del entorno de trabajo adicionado a la Base de Datos. • El valor del entorno de trabajo modificado en la Base de Datos. • El valor del entorno de trabajo eliminado de la Base de Datos. 	
Flujo Normal de Eventos		
Acción del actor	Respuesta del sistema	
1-El líder de proyecto selecciona la opción Entorno de Trabajo.	1.1-El sistema muestra un listado con los valores del entorno de trabajo y las siguientes opciones: <ul style="list-style-type: none"> ▪ Insertar valor entorno de trabajo. ▪ Modificar valor entorno de trabajo. ▪ Eliminar valor entorno de trabajo. 	
Sección 1: "Insertar valor entorno de trabajo"		
Acción del actor	Respuesta del sistema	
1-El líder de proyecto selecciona la opción Insertar valor entorno de trabajo.	1.1-El sistema muestra un formulario donde aparecen los campos necesarios para que el líder de proyecto registre los datos.	

2-El líder de proyecto llena los campos del formulario.	2.1-El sistema verifica que todos los campos estén llenos. 2.2-El sistema guarda todos los datos entrados por el líder de proyecto y finaliza el caso de uso.
Flujos Alternos	
Cursos Alternos	2.1- En caso de que no se llenen los campos, se emite un mensaje “Debe introducir los datos” con el objetivo de que llenen los campos obligatorios y retorna a la acción 2.
Sección 2: “Modificar valor entorno de trabajo”	
Acción del actor	Respuesta del sistema
1-El líder de proyecto selecciona el valor que desea modificar.	1.1-El sistema localiza los datos necesarios y los muestra.
2-El líder realiza la modificación deseada.	2.1-El sistema verifica que todos los campos estén llenos. 2.2-Se actualiza la información entrada por el líder de proyecto y finaliza el caso de uso.
Flujos Alternos	
Cursos Alternos	2.1- En caso de que no se llenen los campos, se emite un mensaje “Debe introducir los datos” con el objetivo de que llenen los campos obligatorios y retorna a la acción 2.
Sección 3: “Eliminar valor entorno de trabajo”	
Acción del actor	Respuesta del sistema
1-El líder de proyecto selecciona el valor que desea eliminar.	1.1-El sistema localiza los datos necesarios y muestra un mensaje de advertencia para la acción a realizar.

2-El líder de proyecto confirma la eliminación.	2.1-El sistema procede a la eliminación de los datos y finaliza el CU.
Flujos Alternos	
Cursos Alternos	2.1-Si el líder de proyecto cancela la acción se culmina el CU sin ejecutar ninguna acción.
Prioridad	Crítico

Tabla 11: Caso de uso del sistema<Gestionar relación riesgo_factor>

Caso de uso	Gestionar relación riesgo_factor	
Actor	Líder de proyecto	
Propósito	Permite al líder de proyecto gestionar (Insertar, Modificar, Eliminar) la relación entre riesgo y factor de riesgo.	
Resumen	El CUS se inicia cuando el líder de proyecto selecciona la opción Relación Riesgo_Factor de Riesgo, luego se selecciona el tipo de gestión, se introducen los datos necesarios, el sistema realiza la acción seleccionada y termina el CU.	
Referencia	R5	
CU asociados		
Precondiciones	El usuario con el rol de líder de proyecto se debe haber autenticado con anterioridad. Los riesgos y factores de riesgos deben ser adicionados con anterioridad.	
Poscondiciones	<ul style="list-style-type: none"> • Información de la relación adicionada a la Base de Datos. • Información de la relación eliminada de la Base de Datos. • Información de la relación modificada en la Base de Datos. 	
Flujo Normal de Eventos		
Acción del actor	Respuesta del sistema	
1-Selecciona la opción de Relación Riesgo_Factor de Riesgo.	1.1-El sistema muestra las opciones: <ul style="list-style-type: none"> • Insertar relación riesgo_factor. 	

	<ul style="list-style-type: none"> • Eliminar relación riesgo_factor. • Modificar relación riesgo_factor.
Sección 1: "Insertar relación riesgo_factor"	
Acción del actor	Respuesta del sistema
1-El líder de proyecto selecciona la opción Insertar relación.	1.1-El sistema muestra un formulario donde aparece un listado de riesgos, otro de factores de riesgos y un campo para introducir el peso de la relación.
2-El líder de proyecto selecciona el factor de riesgo y su riesgo asociado y le inserta su peso correspondiente.	2.1-El sistema verifica estén llenos los campos. 2.2-El sistema verifica que la relación no esté en la Base de Datos. 2.3-El sistema guarda todos los datos entrados por el líder de proyecto y finaliza el caso de uso.
Flujos Alternos	
Cursos Alternos	2.1-Se emite un mensaje "Debe introducir los datos" con el objetivo de que llenen los campos obligatorios y retorna a la acción 2. 2.2-Si ya está registrada, el sistema muestra un mensaje de alerta indicando el error y retorna a la acción 2.
Sección 2: "Eliminar relación riesgo_factor"	
Acción del actor	Respuesta del sistema
1-El líder de proyecto selecciona un riesgo.	1.1-El sistema muestra un listado con las relaciones de ese riesgo y los factores de riesgos asociados.
2-Selecciona la relación que desea eliminar.	2.1-El sistema localiza los datos y muestra un mensaje de confirmación de eliminación.

3-El líder de proyecto confirma la eliminación.	3.1-Se elimina la relación y termina el caso de uso.
Flujos Alternos	
Cursos Alternos	3.1-Si cancela la acción se culmina el CUS sin ejecutar ninguna acción.
Sección 3: “Modificar relación riesgo_factor”	
Acción del actor	Respuesta del sistema
1-El líder de proyecto selecciona un riesgo.	1.1-El sistema muestra un listado con las relaciones que tiene con los factores de riesgos.
2-Selecciona la relación que desea modificar.	2.1-El sistema localiza los datos y los muestra en un formulario.
3-El líder de proyecto realiza las modificaciones deseadas.	3.1-Se verifica que estén llenos los campos. 3.2- Se actualiza la información entrada por el líder de proyecto y finaliza el caso de uso.
Flujos Alternos	
Cursos Alternos	3.1-En caso de que no se llenen los campos, se emite el mensaje: “Debe introducir los datos” y retorna a la acción 3.
Prioridad	Secundario

Tabla 12: Caso de uso del sistema<Gestionar relación riesgo_entorno>

Caso de uso	Gestionar relación riesgo_entorno
Actor	Líder de proyecto
Propósito	Permite al líder de proyecto gestionar (Insertar, Modificar, Eliminar) las relaciones entre riesgos y el entorno de trabajo.
Resumen	El CUS se inicia cuando el líder de proyecto selecciona la opción Relación Riesgo_Entorno de Trabajo, luego se selecciona el tipo de gestión, se introducen los datos necesarios, el sistema realiza la acción seleccionada y termina el CU.

Referencia	R6	
CU asociados		
Precondiciones	<p>El usuario con el rol de líder de proyecto se debe haber autenticado con anterioridad.</p> <p>Los riesgos y el entorno de trabajo deben ser adicionados anteriormente.</p>	
Poscondiciones	<ul style="list-style-type: none"> • Relación adicionada a la Base de Datos. • Relación eliminada de la Base de Datos. • Relación modificada en la Base de Datos. 	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del sistema	
1-Selecciona la opción de Relación Riesgo_ Entorno de Trabajo.	<p>1.1-El sistema muestra las opciones:</p> <ul style="list-style-type: none"> • Insertar relación riesgo_entorno. • Eliminar relación riesgo_entorno. • Modificar relación riesgo_entorno. 	
Sección 1: "Insertar relación riesgo_entorno"		
Acción del actor	Respuesta del sistema	
1-El líder de proyecto selecciona la opción Insertar relación.	<p>1.1-El sistema muestra un listado de riesgos y el listado del entorno de trabajo y un campo para introducir el peso.</p>	
2-El líder de proyecto selecciona el riesgo y el entorno de trabajo asociado y llena el campo del peso de la relación.	<p>2.1-El sistema verifica que los campos estén llenos.</p> <p>2.2-El sistema verifica que la relación no esté registrada en la Base de Datos.</p> <p>2.3-El sistema guarda todos los datos entrados por el líder de proyecto y finaliza el caso de uso.</p>	
Flujos Alternos		
Cursos Alternos	2.1-En caso de que no se llenen los campos, se	

	<p>emite el mensaje: “Debe introducir los datos” y retorna a la acción 2.</p> <p>2.2-Si ya existe la relación, el sistema muestra un mensaje de alerta indicando el error y retorna a la acción 2.</p>
Sección 2: “Eliminar relación riesgo_entorno”	
Acción del actor	Respuesta del sistema
1-El líder de proyecto selecciona un riesgo.	1.1-El sistema muestra un listado con las relaciones que tiene el riesgo con los entornos.
2-El líder de proyecto selecciona la relación que desea eliminar.	2.1-El sistema localiza los datos y muestra un mensaje de confirmación de eliminación.
3-El líder de proyecto ratifica la eliminación.	3.1-Se procede a la eliminación de la relación y termina el caso de uso.
Flujos Alternos	
Cursos Alternos	3.1-Si cancela la acción finaliza el CUS sin ejecutar ninguna acción.
Sección 3: “Modificar relación riesgo_entorno”	
Acción del actor	Respuesta del sistema
1-El líder de proyecto selecciona un riesgo.	1.1-El sistema muestra un listado con las relaciones que tiene el riesgo con los entornos.
2-Selecciona la relación que va a modificar.	2.1-El sistema localiza los datos y los muestra en un formulario.
3-El líder de proyecto realiza las modificaciones deseadas.	<p>3.1-Se verifica que el campo obligatorio esté lleno.</p> <p>3.2-Se actualiza la información entrada por el líder de proyecto y finaliza el caso de uso.</p>
Flujos Alternos	

Cursos Alternos	3.1-En caso de que no se llenen los campos obligatorios, se emite el mensaje: “Debe introducir los datos” y retorna a la acción 3.
Prioridad	Secundario

Tabla 13: Caso de uso del sistema<Gestionar relación factor_entorno>

Caso de uso	Gestionar relación factor_entorno
Actor	Líder de proyecto
Propósito	Permite al líder de proyecto gestionar (Insertar, Modificar, Eliminar) las relaciones entre los factores de riesgos y el entorno de trabajo.
Resumen	El CUS se inicia cuando el líder de proyecto selecciona la opción Relación Factor_Entorno de Trabajo, luego se selecciona el tipo de gestión, se introducen los datos necesarios, el sistema realiza la acción seleccionada y termina el CUS.
Referencia	R7
CU asociados	
Precondiciones	El usuario con el rol de líder de proyecto se debe haber autenticado con anterioridad. Los factores de riesgos y el entorno de trabajo deben ser adicionados anteriormente.
Poscondiciones	<ul style="list-style-type: none"> • Relación adicionada a la Base de Datos. • Relación eliminada de la Base de Datos. • Relación modificada en la Base de Datos.
Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema
1-Selecciona la opción de Relación Factor_Entorno de Trabajo.	1.1-El sistema muestra las opciones: <ul style="list-style-type: none"> • Insertar relación factor_entorno. • Eliminar relación factor_entorno.

	<ul style="list-style-type: none"> • Modificar relación factor_entorno.
Sección 1: “Insertar relación factor_entorno”	
Acción del actor	Respuesta del sistema
1-El líder de proyecto selecciona la opción Insertar relación.	1.1-El sistema muestra un listado de los factores de riesgos, el listado del entorno de trabajo y un campo para introducir el peso.
2-El líder de proyecto selecciona el factor de riesgo y el entorno de trabajo asociado e introduce el peso.	2.1-El sistema verifica que los campos estén llenos. 2.2-El sistema verifica que la relación no esté registrada en la Base de Datos. 2.3-El sistema guarda todos los datos entrados por el líder de proyecto y termina el caso de uso.
Flujos Alternos	
Cursos Alternos	2.1-En caso de que no se llenen los campos obligatorios, se emite el mensaje: “Debe introducir los datos” y retorna a la acción 2. 2.2-Si ya existe la relación, el sistema muestra un mensaje de alerta indicando el error y retorna a la acción 2.
Sección 2: “Eliminar relación factor_entorno”	
Acción del actor	Respuesta del sistema
1-El líder de proyecto selecciona un factor de riesgo.	1.1-El sistema muestra un listado con las relaciones que tiene este factor con los entornos.
2-El líder de proyecto selecciona la relación que desea eliminar.	2.1-El sistema localiza los datos y muestra un mensaje de confirmación de eliminación.
3-El líder de proyecto confirma la eliminación.	3.1-Se elimina la relación y termina el CU.

Flujos Alternos	
Cursos Alternos	3.1-Si cancela la acción se culmina el CU sin ejecutar ninguna acción.
Sección 3: “Modificar relación factor_entorno”	
Acción del actor	Respuesta del sistema
1-El líder de proyecto selecciona un factor de riesgo.	1.1-El sistema muestra un listado con las relaciones que tiene éste con los entornos.
2-Selecciona la relación que va a modificar.	2.1-El sistema localiza los datos y los muestra en un formulario.
3-El líder de proyecto realiza las modificaciones deseadas.	3.1-Se verifica que los campos obligatorios estén llenos. 3.2-Se actualiza la información entrada por el líder de proyecto y finaliza el caso de uso.
Flujos Alternos	
Cursos Alternos	3.1-En caso de que no se llenen los campos obligatorios, se emite el mensaje: “Debe introducir los datos” y retorna a la acción 3.
Prioridad	Secundario

Tabla 14: Caso de uso del sistema<Gestionar usuarios>

Caso de uso	Gestionar usuarios
Actor	Líder de proyecto
Propósito	Permite al líder de proyecto gestionar (Insertar, Modificar, Eliminar, Asignar permisos) los usuarios de su proyecto.
Resumen	El CUS se inicia cuando el líder de proyecto selecciona la opción Usuarios, luego se selecciona el tipo de gestión, se introducen los datos necesarios, el sistema realiza la acción seleccionada y termina el CU.
Referencia	R8

CU asociados	
Precondiciones	El usuario con el rol de líder de proyecto debe autenticarse con anterioridad.
Poscondiciones	<ul style="list-style-type: none"> • Información del usuario adicionada a la Base de Datos. • Información del usuario eliminada de la Base de Datos. • Información del usuario modificada en la Base de Datos.
Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema
1-Selecciona la opción Usuarios.	1.1-El sistema muestra un listado con los usuarios en dependencia del proyecto y las opciones: <ul style="list-style-type: none"> • Insertar usuarios. • Eliminar usuarios. • Modificar usuarios. • Asignar permisos.
Sección 1: “Insertar usuarios”	
Acción del actor	Respuesta del sistema
1-El líder de proyecto selecciona la opción Insertar usuario.	1.1-El sistema muestra un formulario donde aparecen los campos necesarios para que el líder de proyecto registre los datos del usuario local. <ul style="list-style-type: none"> • Tipo de Usuario • Nombre • Usuario • Contraseña • Rol • Departamento

2-El líder de proyecto llena los campos del formulario.	2.-El sistema verifica que todos los campos estén llenos. 2.2-El sistema guarda todos los datos entrados por el líder de proyecto y termina el caso de uso.
Flujos Alternos	
Cursos Alternos	2.1- En caso de que no se llenen los campos obligatorios, se emite un mensaje “Debe introducir los datos” con el objetivo de que llene los campos obligatorios y retorna a la acción 2.
Sección 2: “Eliminar usuarios”	
Acción del actor	Respuesta del sistema
1-El líder de proyecto selecciona el usuario que desea eliminar.	1.1-El sistema localiza los datos del usuario del proyecto y muestra un mensaje de advertencia para la acción a realizar.
2-Confirma la eliminación del usuario del proyecto.	2.1-El sistema procede a la eliminación de los datos y termina el CU.
Flujos Alternos	
Cursos Alternos	2.1-Si cancela la acción se culmina el CU sin ejecutar ninguna acción.
Sección 3: “Modificar usuarios”	
Acción del actor	Respuesta del sistema
1-El líder de proyecto selecciona el usuario del proyecto que desea modificar.	1.1-El sistema localiza los datos del usuario del proyecto y los muestra en un formulario.
2-El líder de proyecto realiza las modificaciones deseadas.	2.1-Se verifica que todos los campos obligatorios estén llenos. 2.2-Se actualiza la información entrada por el líder de proyecto y finaliza el caso de uso.

Flujos Alternos	
Cursos Alternos	2.1-En caso de que no se llenen los campos obligatorios, se emite el mensaje: "Debe introducir los datos" y retorna a la acción 2.
Sección 4: "Asignar permisos"	
Acción del actor	Respuesta del sistema
1-El líder de proyecto selecciona el usuario del proyecto que desea asignarle permiso.	1.1-El sistema muestra un formulario que permite cambiarle el proyecto y el rol al usuario seleccionado.
2-El líder de proyecto realiza la asignación deseada.	2.1-El sistema guarda los cambios y termina el caso de uso.
Flujos Alternos	
Cursos Alternos	1.1-El sistema muestra un formulario donde aparecen los campos necesarios para que el líder de proyecto registre los datos del usuario del dominio UCI. <ul style="list-style-type: none"> • Tipo de Usuario • Nombre • Usuario • Rol • Departamento
Prioridad	Secundario

Tabla 15: Caso de uso del sistema<Generar reportes>

Caso de uso	Generar reportes
Actor	Usuario
Propósito	Permite a los usuarios ver reportes.

Resumen	El CUS se inicia cuando el usuario selecciona la opción Reportes, luego se selecciona el tipo de reporte que desea ver, el sistema lo muestra y termina el CU.	
Referencia	R9	
CU asociados		
Precondiciones	El usuario se debe haber autenticado con anterioridad.	
Poscondiciones	<ul style="list-style-type: none"> Mostrar los reportes solicitados. 	
Flujo Normal de Eventos		
Acción del actor	Respuesta del sistema	
1-El usuario selecciona la opción Reportes.	1.1-El sistema muestra un formulario dándole la posibilidad de seleccionar el tipo de reporte que desea.	
2-El usuario selecciona lo que desea ver.	2.1-El sistema localiza los datos, los muestra y termina caso de uso.	
Flujos Alternos		
Cursos Alternos		
Prioridad	Secundario	

Tabla 16: Caso de uso del sistema<Exportar a PDF>

Caso de uso	Exportar a PDF
Actor	Usuario
Propósito	Permite al usuario exportar los reportes a formato PDF.
Resumen	El CUS se inicia cuando el usuario selecciona la opción de Exportar a PDF, el sistema realiza la acción seleccionada y termina el CUS.
Referencia	R9
CU asociados	
Precondiciones	El usuario se debe haber autenticado con anterioridad.
Poscondiciones	

Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema
1-El usuario selecciona la opción Exportar a PDF.	1.1-El sistema muestra una ventana para que el usuario escoja la opción de abrir o guardar el PDF.
2-El usuario escoge una opción.	2.1-El sistema realiza la acción seleccionada y finaliza el caso de uso.
Flujos Alternos	
Cursos Alternos	
Prioridad	Secundario

Tabla 17: Caso de uso del sistema<Notificar disparo>

Caso de uso	Notificar disparo
Actor	Líder de proyecto
Propósito	Permite al líder de proyecto notificar cuando se dispare un riesgo o un factor de riesgo.
Resumen	El CUS se inicia cuando el líder de proyecto selecciona la opción de Notificar Disparo de un riesgo o un factor de riesgo, el sistema realiza la acción seleccionada y termina el CUS.
Referencia	R10,R11
CU asociados	
Precondiciones	El líder de proyecto se debe haber autenticado con anterioridad.
Poscondiciones	
Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema
1-El líder de proyecto necesita notificar el disparo de un riesgo o un factor de riesgo.	1.1-El sistema muestra las siguientes opciones: <ul style="list-style-type: none"> • Notificar disparo riesgo. • Notificar disparo factor de riesgo.

Sección 1: “Notificar disparo riesgo”	
Acción del actor	Respuesta del sistema
1-El líder de proyecto selecciona la opción Notificar Disparo.	1.1-El sistema muestra el listado de riesgo existente.
2-Selecciona el riesgo que se le disparó y llena el campo de la fecha.	2.1-El sistema guarda los datos y termina el caso de uso.
Sección 2: “Notificar disparo factor de riesgo”	
Acción del actor	Respuesta del sistema
1-El líder de proyecto selecciona la opción Notificar Disparo.	1.1-El sistema muestra el listado de los factores de riesgos existente.
2-Selecciona el factor de riesgo que se le disparó y llena el campo de la fecha.	2.1-El sistema guarda los datos y termina el caso de uso.
Flujos Alternos	
Cursos Alternos	
Prioridad	Auxiliar

Tabla 18: Caso de uso del sistema<Autenticar usuario>

Caso de uso	Autenticar usuario
Actor	Usuario
Propósito	Todo el usuario que de una forma u otra interactúe con el sistema debe autenticarse primeramente.
Resumen	El usuario a la hora de interactuar con la aplicación debe autenticarse, el sistema le muestra una interfaz con un formulario, el usuario introduce los datos y termina el caso de uso.
Referencia	R12
CU asociados	
Precondiciones	
Poscondiciones	El sistema permite que el usuario se autentifique y acceda a las páginas

	que tiene acceso.
Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema
1-El usuario inicia la aplicación.	1.1-El sistema le muestra una interfaz con un formulario con los campos: <ul style="list-style-type: none"> • Usuario • Contraseña • Dominio
2-El usuario ingresa los datos en formulario.	2.1-El sistema verifica que los datos estén correctamente escritos. 2.2-El sistema le da acceso al usuario de interactuar con la aplicación y termina el caso de uso.
Flujos Alternos	
Cursos Alternos	2.1-Si los datos son incorrectos el sistema no permite el acceso a la aplicación y vuelve a mostrarle la interfaz de autenticación.
Prioridad	Crítico

En el **Anexo 2** se encuentran los prototipos de interfaz de usuario de cada uno de los casos de uso detallados anteriormente.

2.5 Conclusiones parciales

- Planteamiento de los requisitos funcionales (RF) y los no funcionales (RNF) de la aplicación.
- Modelación de la aplicación en términos de casos de uso del sistema.
- Obtención enfocada del análisis y futuro diseño de la aplicación Web a desarrollar.

Capítulo 3

Análisis y Diseño del Sistema

3.1 Introducción

En este capítulo se muestra la arquitectura que guiará la construcción del software, se expone además la concepción general referente al Análisis y Diseño de la aplicación propuesta, donde el Análisis tiene como objetivo mantener un modelo eficiente de la solución propuesta que sirva de base para el Diseño. Este último tiene a su vez como objetivo, presentar cómo es que está construido el sistema, lo cual se realiza en este caso a partir de los diagramas de clases Web, que tienen la finalidad de describir la interacción entre las distintas páginas de la aplicación. Por otro lado se presenta el diseño de la Base de Datos del sistema y los principios del diseño.

3.2 Arquitectura

Una Arquitectura de Software, también denominada *Arquitectura lógica*, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información (RICOTE y FIGUEREDO, 2006). La misma tiene que ver con el diseño y la implementación de estructuras de software de alto nivel. Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, como la confiabilidad, escalabilidad, portabilidad y disponibilidad.

Generalmente, no es necesario inventar una nueva arquitectura software para cada sistema de información. Lo habitual es adoptar una arquitectura conocida en función de sus ventajas e inconvenientes para cada caso en concreto.

La herramienta se basa en una arquitectura tres capas la cual está centrada básicamente en dividir un problema en pequeñas partes que puedan ser manejadas e implementadas de forma independiente, dichas partes poseerán responsabilidades específicas que no dependan del funcionamiento de las otras o al menos que su dependencia sea mínima. Este aspecto constituye una ventaja considerable pues proporciona una amplia reutilización de las clases implementadas al hacer abstracciones de las distintas

funcionalidades o responsabilidades del sistema agrupándolas en capas. La arquitectura en capas brinda además beneficios como:

- Desarrollo paralelo (en cada capa).
- Aplicaciones más robustas debido al encapsulamiento.
- Mantenimiento y soporte más sencillo (es más sencillo cambiar un componente que modificar una aplicación monolítica).
- Mayor flexibilidad (se pueden añadir nuevos módulos para dotar al sistema de nueva funcionalidad).
- Alta escalabilidad. La principal ventaja de una aplicación distribuida bien diseñada es que puede manejar muchas peticiones con el mismo rendimiento simplemente añadiendo más hardware. El crecimiento es casi lineal y no es necesario añadir más código para conseguir esta escalabilidad.

Una aplicación de tres capas es una aplicación cuya funcionalidad puede ser segmentada en tres niveles lógicos (capas):

Capa de Presentación: es responsable de obtener información del usuario, enviar la información del usuario a los servicios de negocios para su procesamiento, recibir los resultados del procesamiento de los servicios de negocios y presentar estos resultados al usuario.

Capa de Negocio (Lógica de Negocios): es la responsable de recibir la entrada del nivel de presentación, interactuar con los servicios de datos para ejecutar las operaciones de negocios y enviar el resultado procesado al nivel de presentación.

Capa de Acceso a Datos: es la responsable de almacenar, recuperar y mantener los datos, además de la integridad de los mismos.

En la capa de Acceso a Datos se implementó el patrón de diseño Factory, el cual permite centralizar la creación de objetos en una clase a pedido y devuelve instancias de los mismos.

3.3 Modelo de análisis

Durante el Análisis, se estudian los requisitos que se describieron en la captura de requisitos, refinándolos y estructurándolos. El objetivo es conseguir una comprensión más precisa de los requisitos y una

descripción de los mismos que sea fácil de mantener y que ayude a estructurar el sistema entero, incluyendo su arquitectura.

A pesar de que en el modelo de análisis hay un refinamiento de los requisitos, no se toman en cuenta el lenguaje de programación a usar en la construcción, la plataforma en la que se ejecutará la aplicación, los componentes prefabricados o reusables de otras aplicaciones, entre otras características que afectan al sistema; el objetivo es comprender perfectamente los requisitos del software y no precisar cómo se implementará la solución. En este modelo los casos de uso del sistema se describen a través de las clases del análisis y sus objetos.

3.3.1 Diagrama de clases del análisis

El diagrama de clases del análisis es uno de los artefactos que se generan en el modelo de análisis. El mismo representa los objetos en un dominio del problema. Se utilizan para el modelado de las realizaciones de los casos de uso, mostrando las clases y sus relaciones. Las clases que están presentes en estos diagramas se clasifican en Interfaz, Control y Entidad.

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

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

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

A continuación se muestra el diagrama de clases del análisis para el CU crítico Gestionar riesgos En el **Anexo 3** se pueden encontrar los restantes diagramas de clases del análisis por cada caso de uso.

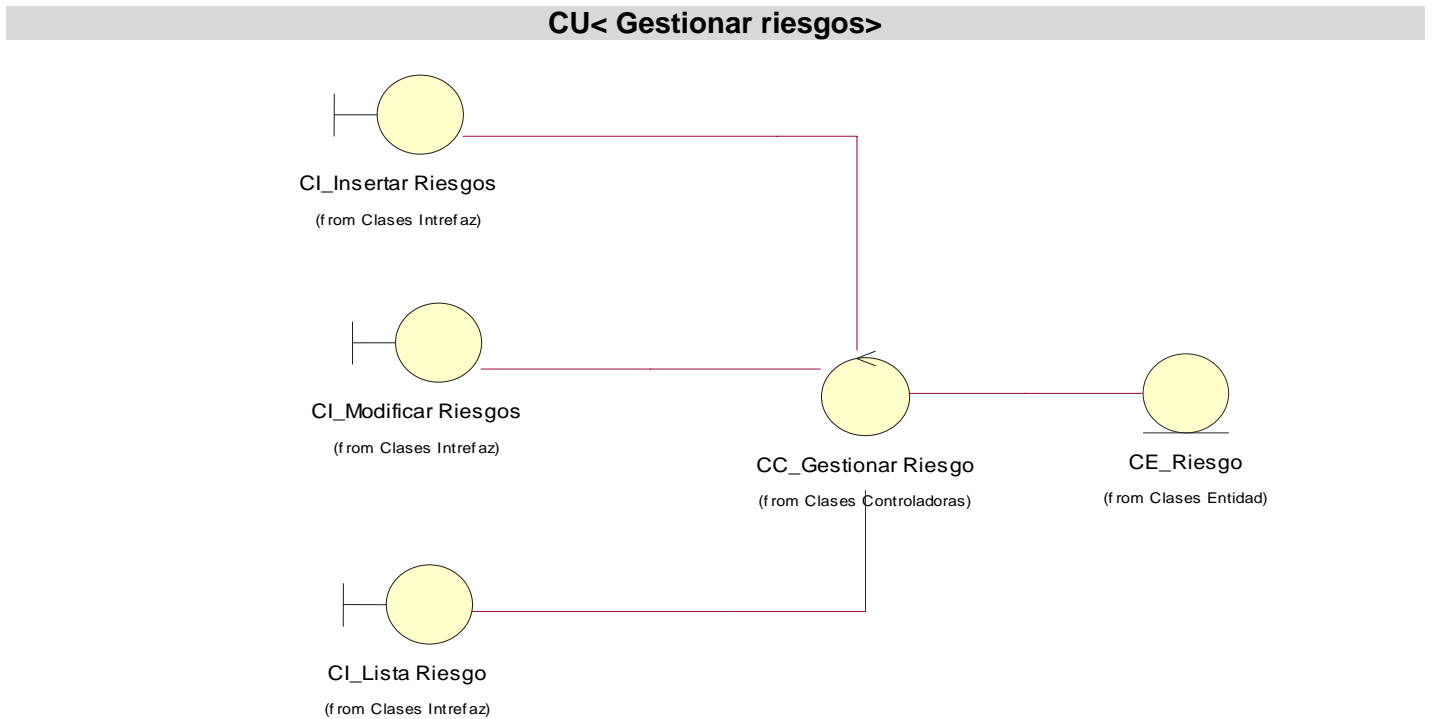


Figura 4: Diagrama de clases del análisis <Gestionar riesgos>

3.3.2 Diagrama de interacción

La secuencia de acciones en un caso de uso comienza cuando un actor invoca el caso de uso mediante el envío de algún tipo de mensaje al sistema. Si se considera el “interior” del sistema, un objeto de interfaz recibirá este mensaje del actor. El objeto de interfaz enviará a su vez un mensaje hacia algún otro objeto, y de esa forma los objetos implicados interactuarán para llevar a cabo el caso de uso. (RUMBAUHG *et al.*, 2000a)

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema, éstos muestran una interacción, que consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos. Un **diagrama de secuencia** es un diagrama de interacción que destaca la ordenación temporal de los mensajes; un **diagrama de colaboración** es un diagrama de interacción que expone la organización estructural de los objetos que envían y reciben mensajes.

En la **Figura 5** se muestra el diagrama de colaboración para el escenario Insertar riesgos del CU crítico Gestionar riesgos. En el **Anexo 4** se pueden encontrar los restantes diagramas de colaboración por cada caso de uso.

Escenario <Insertar riesgos>

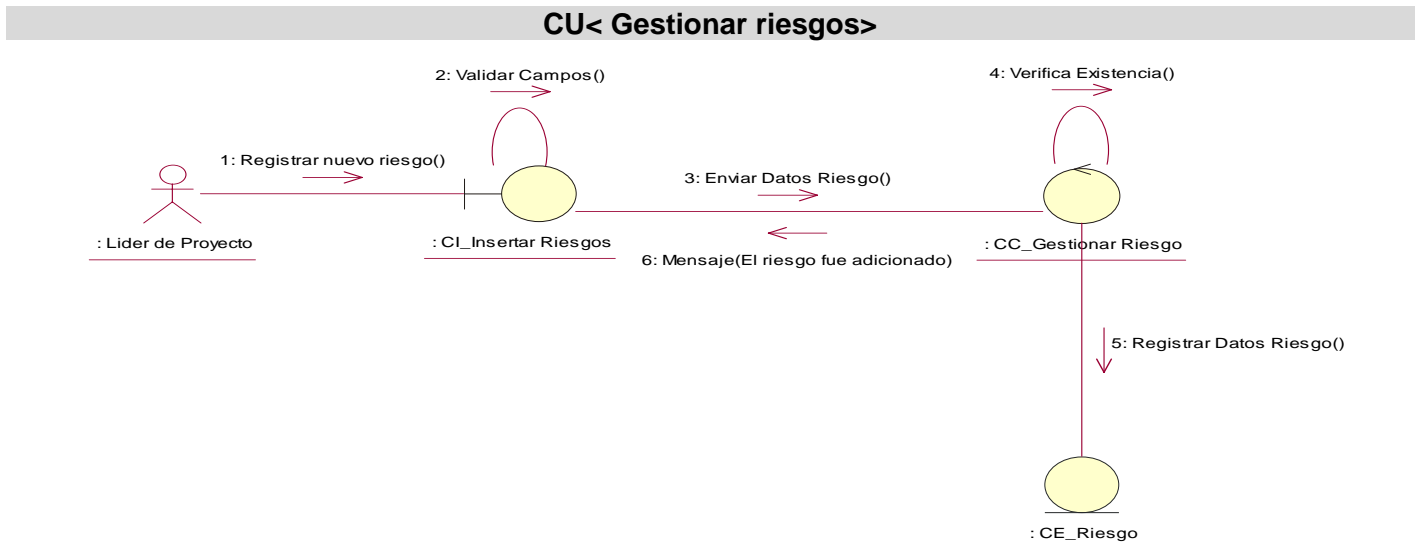


Figura 5: Diagrama de colaboración<Insertar riesgos>

3.4 Modelo de diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar, constituyendo una entrada principal en la actividad de Implementación. (RUMBAUHG *et al.*, 2000b)

En este modelo, los casos de uso son realizados por las clases del diseño y sus objetos, lo cual se denota por la realización de casos de uso del diseño que describe cómo se realizan éstos en particular.

3.4.1 Diseño de la Base de Datos

Uno de los pasos cruciales en la construcción de una aplicación que maneje un conjunto de datos, es sin dudas, el diseño de la Base de Datos (BD). Si las tablas no son definidas apropiadamente, se puede hacer complicado el trabajo a la hora de ejecutar consultas para obtener algún tipo de información.

Para lograr diseñar la Base de Datos se parte del diagrama de clases persistentes, que son las clases cuyo valor va a perdurar en el tiempo.

Modelo lógico de datos.

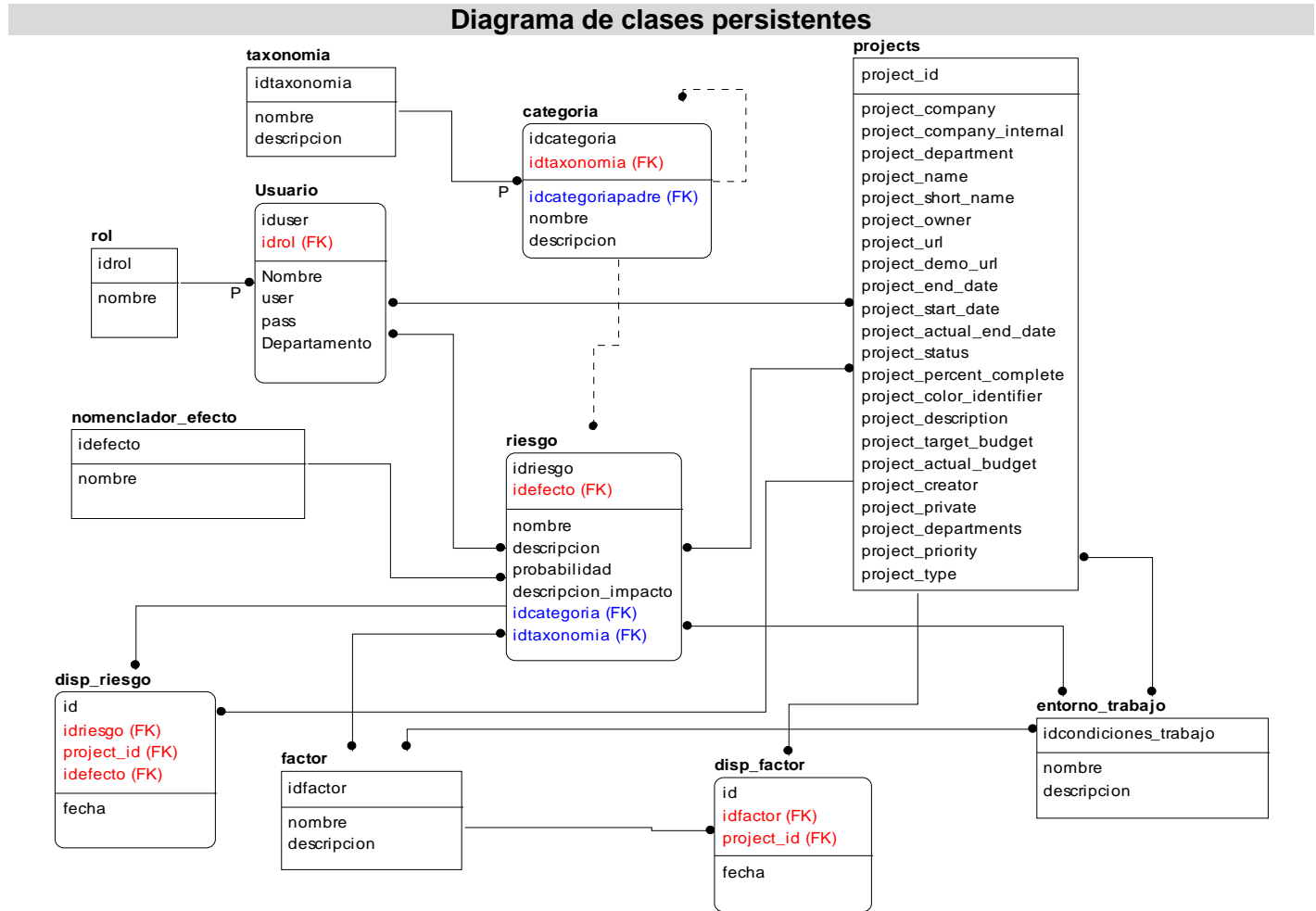


Figura 6: Diagrama de clases persistentes

A partir del diagrama de clases persistentes se obtuvo el modelo de datos que se muestra en la **Figura 7**. El modelo de datos se utilizó para definir los cambios de las clases persistentes en las estructuras de datos utilizadas en el sistema.

Modelo físico de datos.

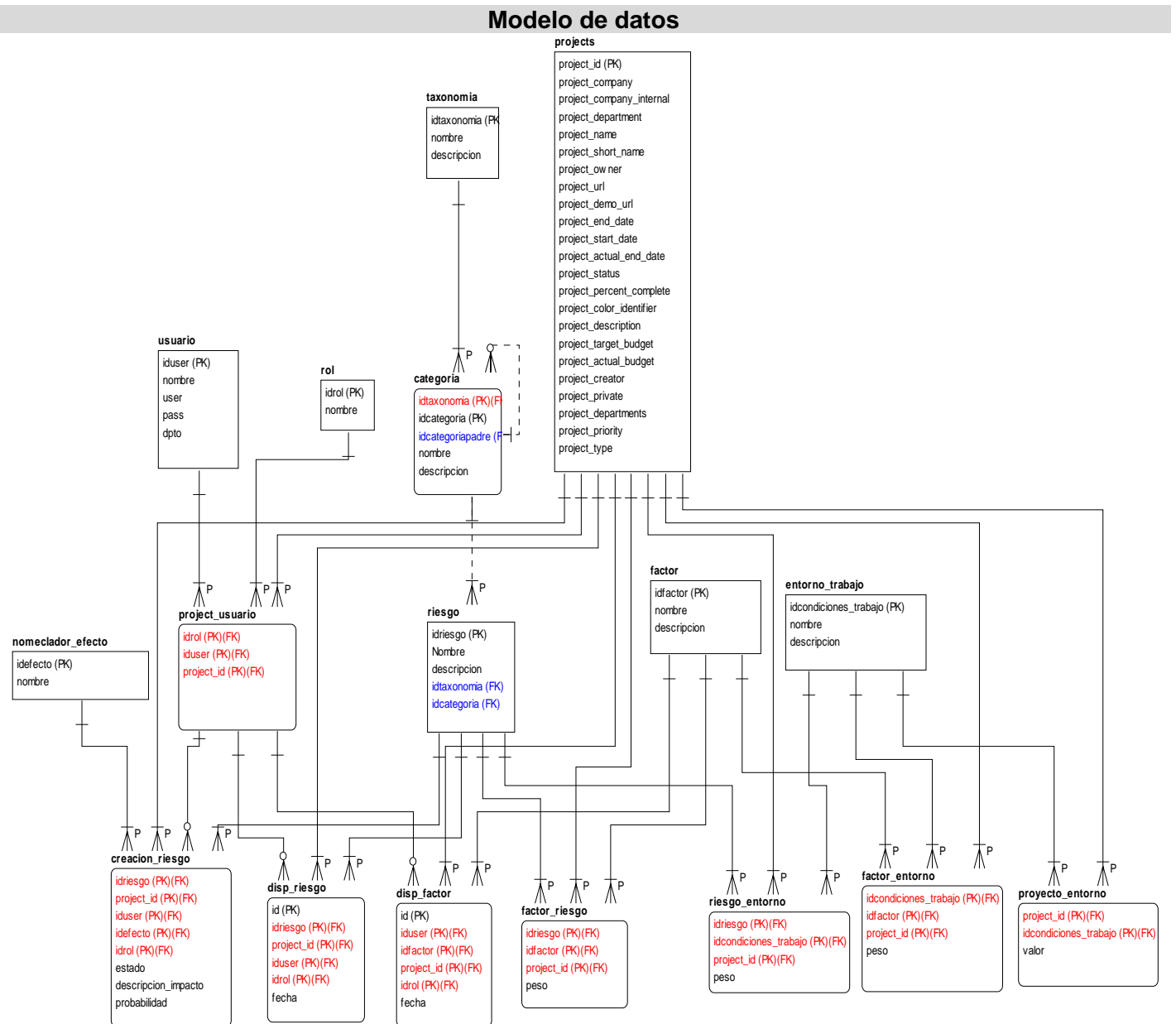


Figura 7: Modelo de datos.

En el **Anexo 8** se describen todas las tablas de la base de datos.

3.4.2 Diagrama de clases del diseño

La forma tradicional de modelar las clases no es aplicable a la hora de diseñar una aplicación Web. Por tal motivo se creó una extensión UML que permitiera representar el nivel de abstracción adecuado y la relación con los restantes artefactos. Para lograr una mejor comprensión de estos diagramas se utilizaron una serie de estereotipos de clases. **Ver Anexo 5.**

Se elaboró un diagrama de clases Web para cada caso de uso del sistema de forma tal que se comprenda cómo se relacionan los distintos componentes en la realización de cada uno de ellos. **Ver Anexos 6 y 7.**

3.4.3 Principios del diseño

El diseño de una aplicación constituye uno de los puntos más importantes a tratar en el proceso de desarrollo de la misma y tiene que basarse en el usuario al cual va dirigido. El diseño Web no se refiere sólo a la apariencia o la combinación de colores, de él depende que los servicios puedan ser usados con facilidad y que la información mostrada sea útil.

Una aplicación con una interfaz bien diseñada debe tener, además de un buen diseño gráfico, una buena navegabilidad, usabilidad y distribución de los contenidos. Para ello, este sistema utiliza ciertos principios generales que garantizan la usabilidad en los diseños para aplicaciones Web:

- Mostrar al usuario solamente aquellas opciones a las que, dado su rol, tiene derecho a acceder.
- Requerir de los usuarios un mínimo esfuerzo para alcanzar sus objetivos.

3.5 Estándares en la interfaz de la aplicación

La interfaz de una aplicación es el medio por el cual los usuarios interactúan con el sistema, ésta debe ser lo más amigable posible para lograr que se sientan identificados con la misma. Para el diseño de la interfaz del sistema se tuvo en cuenta lo siguiente:

- La utilización de un mismo formato y estilo en cada una de las páginas.
- Facilitar al usuario la plena navegabilidad dentro de la aplicación.
- Evitar sobrecarga de colores e imágenes.
- Proporcionar un ambiente amigable.

3.6 Tratamiento de excepciones

El tratamiento de errores posibilita el buen funcionamiento de una aplicación proporcionándole una sobresaliente apariencia ante los clientes. En el sistema se chequea la entrada de datos del usuario a nivel de interfaz. Cuando se produce un error por la entrada incorrecta de un valor suministrado por el usuario se le muestra un mensaje. Los errores en la aplicación se controlan principalmente en el lado del cliente, manejando el lenguaje Java Script y sus mensajes de alerta.

3.7 Conclusiones parciales

- Se mostraron los diferentes elementos que ilustran cómo está construido el sistema, en términos de clases del análisis y del diseño; este último dando la posibilidad de comprender la lógica del sistema en general.
- Se presentó el diseño de la Base de Datos del sistema.
- Por último fueron expuestos los principios del diseño, los estándares de la interfaz de usuario de la aplicación y cómo se le dará tratamiento a los errores.

Capítulo 4

Implementación y Pruebas

4.1 Introducción

En este capítulo se describe cómo los elementos del modelo de diseño son implementados en términos de componentes (ficheros de código fuente, scripts ejecutables) y cómo se organizan de acuerdo a los nodos específicos en el modelo de despliegue. Además se exponen las diferentes pruebas realizadas a cada caso de uso, siguiendo específicamente, el método de pruebas de Caja Negra.

4.2 Modelo de implementación

En el modelo de implementación se describe cómo los elementos del diseño, es decir las clases, se implementan en términos de componentes, como son: ficheros de código fuente, ejecutables, entre otros. Además, se describe cómo están organizados los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje de programación que se utilice, así como la dependencia existente entre estos componentes.(RUMBAUHG *et al.*, 2000b)

4.2.1 Diagramas de componentes

Un diagrama de componentes representa las dependencias entre componentes de software, incluyendo componentes de código fuente, de código binario y ejecutable. Algunos componentes existen en tiempo de compilación, algunos en tiempo de enlace y algunos en tiempo de ejecución, otros en varias de éstas.(RUMBAUHG *et al.*, 2000a)

Un componente es la parte modular de un sistema, desplegable y reemplazable que encapsula implementación y un conjunto de interfaces y proporciona la realización de los mismos. Un componente típicamente contiene clases y puede ser implementado por uno o más artefactos.

A continuación se muestran los diagramas de componentes del sistema.

Diagrama de componentes general

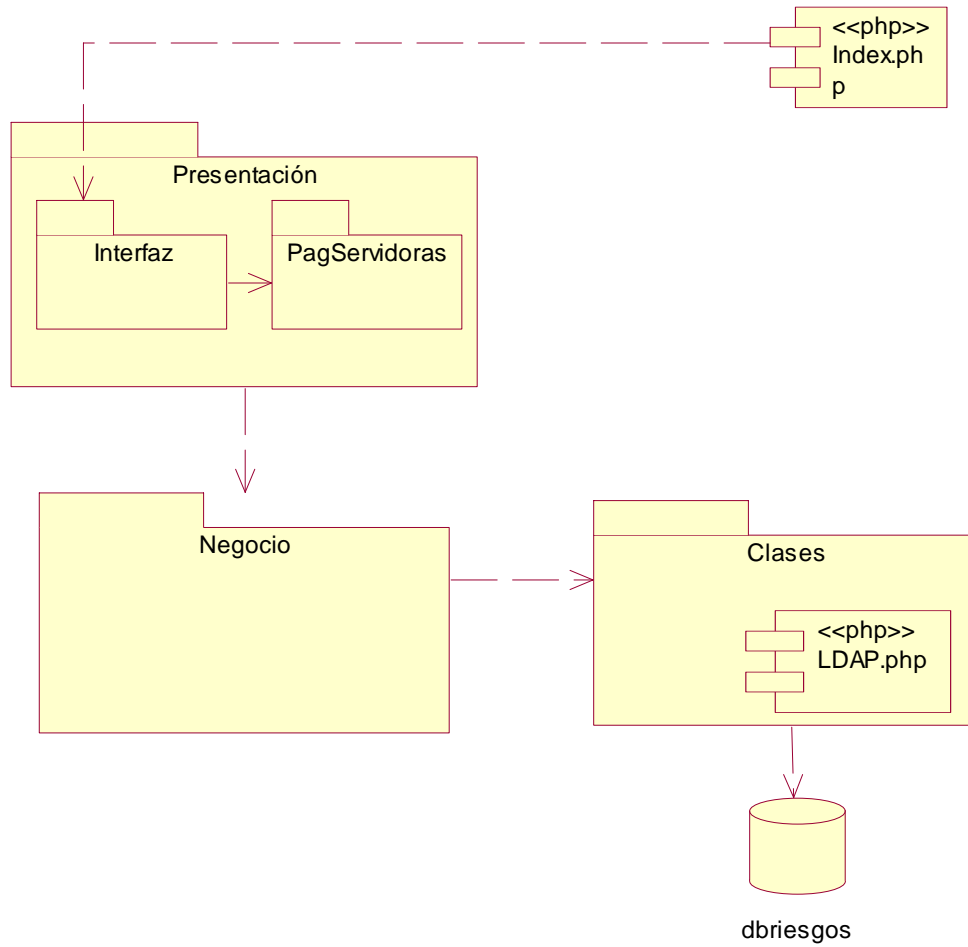


Figura 8: Diagrama de componentes general del sistema.

Paquete de componentes <PagServidoras>

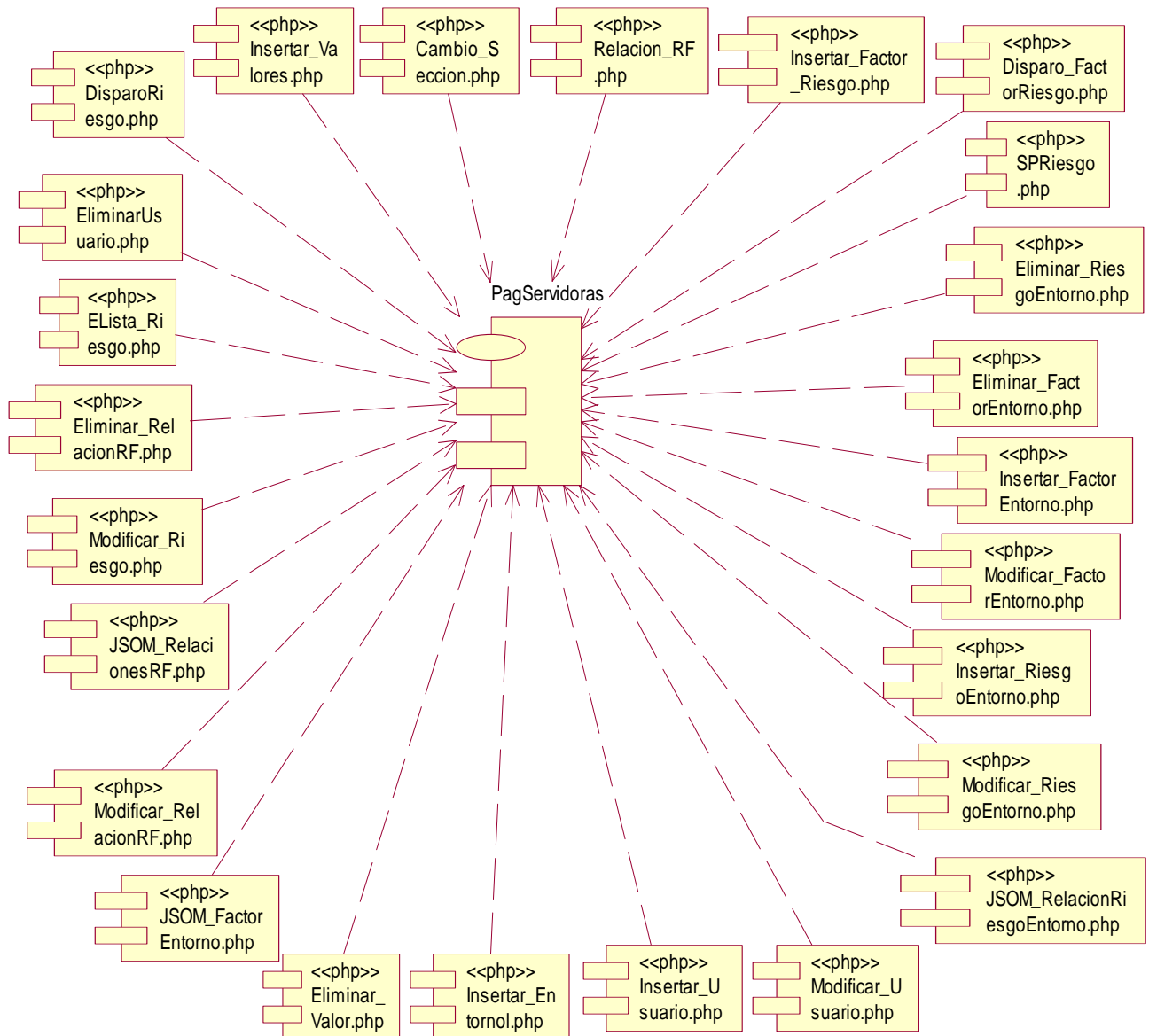


Figura 10: Diagrama de componente del paquete PagServidoras.

Paquete de componentes <Negocio>

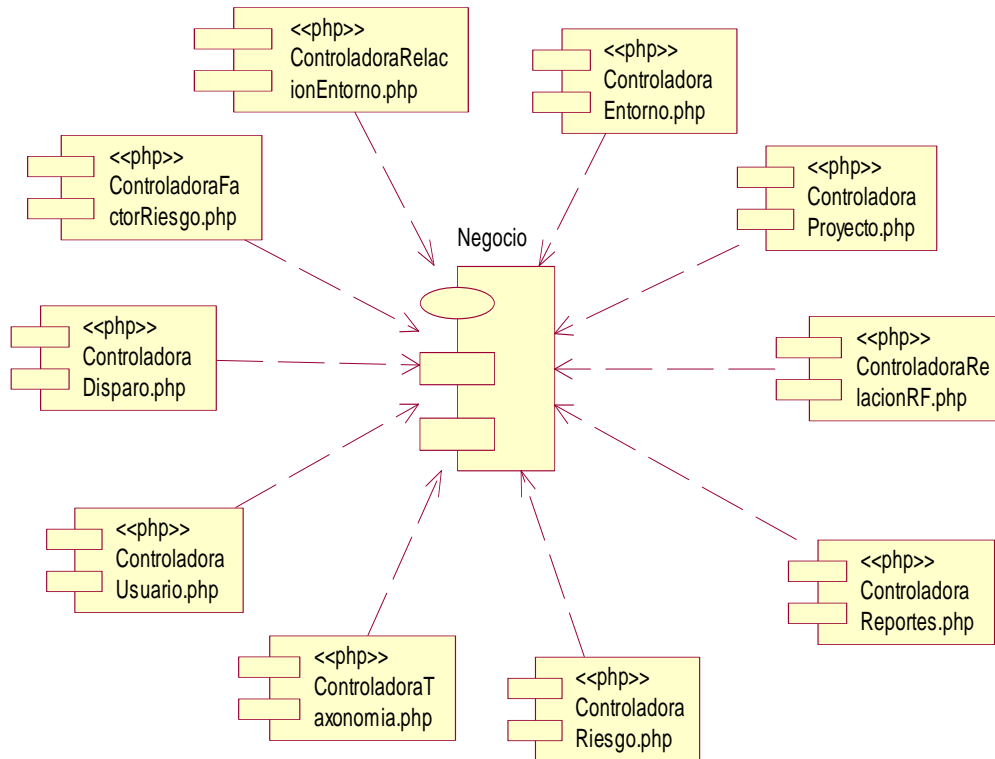


Figura 11: Diagrama de componentes del Paquete Negocio.

Paquete de componentes <Clases>

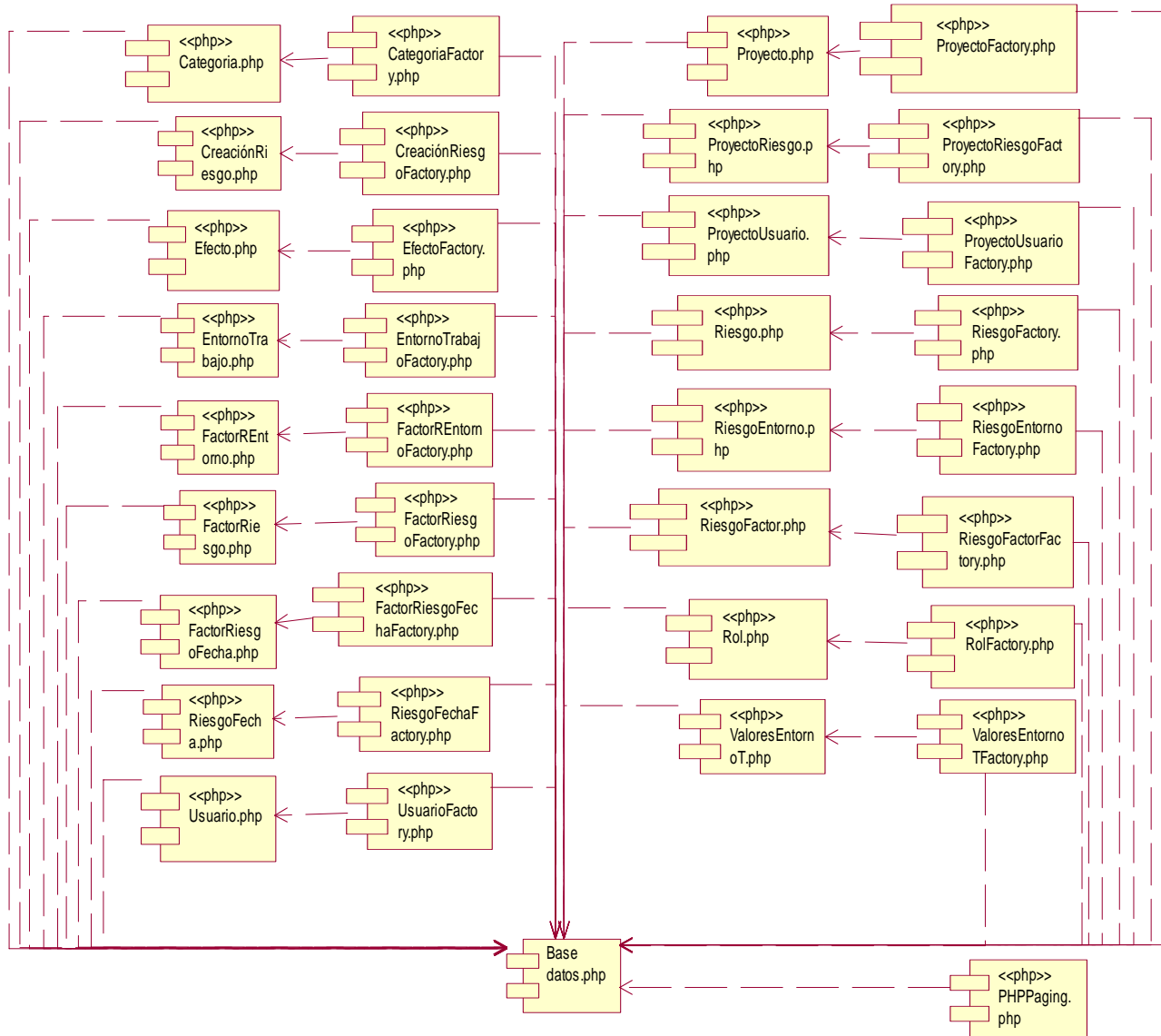


Figura 12: Diagrama de componentes del Paquete Clases.

4.2.2 Diagrama de despliegue

En un diagrama de despliegue se muestran las relaciones físicas entre los componentes de hardware y de software, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes de software, que no son más, que los procesos que se ejecutan en ellos. Se puede decir también que un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación, donde un nodo puede contener instancias de componentes. En general, un nodo se entiende como una unidad de computación de algún tipo como es el caso de impresoras o la misma computadora. A continuación se muestra el diagrama de despliegue del sistema:

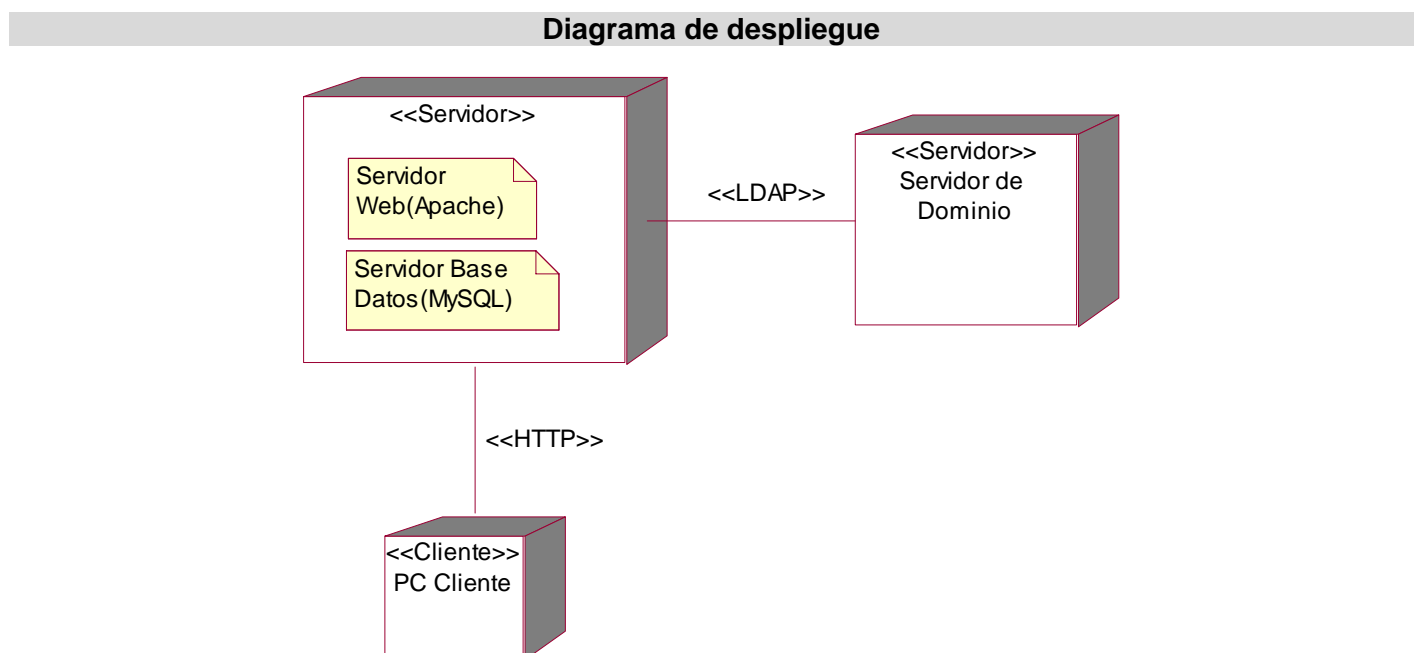


Figura 13: Diagrama de despliegue.

4.3 Pruebas

Las pruebas constituyen una actividad en la cual un sistema o componente de éste, es ejecutado bajo ciertas condiciones o requerimientos específicos, en el que los resultados obtenidos son observados y registrados, para la realización posterior de alguna evaluación de dicho componente o sistema.

Dentro de los métodos de pruebas que existen, se encuentran las conocidas como prueba de Caja Negra o prueba de comportamiento, que se centran en los requisitos funcionales y permiten al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa (PRESSMAN, 2002). Estas pruebas tienen las siguientes características:

- Verifican las especificaciones funcionales y no consideran la estructura interna del programa.
- No validan funciones ocultas, como es el caso de las funciones que son implementadas pero no descritas durante la fase de diseño, por lo que los errores asociados a ellas no serán encontrados.
- Son realizadas sin el conocimiento interno del producto.

A continuación se presentan algunas de las pruebas realizadas al sistema por cada caso de uso, siguiendo el método de Caja Negra descrito anteriormente. El resto de las pruebas realizadas se muestran en el **Anexo 9**.

4.3.1 Pruebas de Caja Negra para el caso de uso <Autenticar usuario>

Condiciones de Entrada	Casos Válidos	Casos no Válidos
Usuario	Cadena de caracteres	Dejar vacío el campo usuario.
Contraseña	Cadena de caracteres	Dejar vacío el campo contraseña.

Caso de uso	Autenticar usuario
Caso de prueba	Permitir autenticar un usuario entrando correctamente los datos.
Entrada	
El usuario introduce correctamente los datos necesarios para acceder al sistema. Usuario: "admin". Contraseña: "admin".	
Resultados	El sistema permite el acceso al usuario autenticado y lo redirecciona para la página de administrador.
Condiciones	Usuario y contraseña introducidos correctamente.

Caso de uso	Autenticar usuario
Caso de prueba	Permitir autenticar un usuario dejando vacío un campo.
Entrada	
El usuario no introduce la contraseña para acceder al sistema. Usuario: "admin". Contraseña: "".	
Resultados	El sistema muestra un mensaje de alerta notificando el error ("Debe introducir la contraseña").
Condiciones	No introduce el dato para el campo contraseña.

4.3.2 Pruebas de Caja Negra para el caso de uso <Gestionar riesgos>

Condiciones de Entrada	Casos Válidos	Casos no Válidos
Nombre	Cadena de caracteres.	Dejar el campo vacío. Introducir números.
Descripción	Cadena de caracteres.	Dejar el campo vacío.
Clase	Seleccionar clase.	No seleccionar una clase.
Elemento	Seleccionar elemento.	No seleccionar un elemento.
Categoría	Seleccionar categoría.	No seleccionar una categoría.
Probabilidad	Número de 0 a 1.	Dejar el campo vacío. Números fuera del rango de 0 a 1.
Efecto	Seleccionar el efecto.	No seleccionar ningún efecto.
Descripción del impacto	Cadena de caracteres.	Dejar el campo vacío.

Escenario <Insertar riesgos>

Caso de uso	Gestionar riesgos(Insertar riesgos)
Caso de prueba	Adicionar un riesgo entrando los datos del mismo correctamente.
Entrada	
<p>El líder introduce correctamente los datos necesarios para adicionar un riesgo. Nombre: “Insuficientes puestos de trabajo”. Descripción: “No hay suficientes computadoras para trabajar en el proyecto”. Probabilidad:”0.3”. Efecto: “Serios”. Descripción del impacto: “Retraso en la realización del proyecto”.</p>	
Resultados	El sistema adiciona el riesgo a la Base de Datos y redirecciona al líder a la página del listado general de riesgos.
Condiciones	Datos entrados correctamente.

Caso de uso	Gestionar riesgos(Insertar riesgos)
Caso de prueba	Adicionar un riesgo dejando un campo obligatorio vacío.
Entrada	
<p>El líder introduce incorrectamente los datos necesarios para adicionar un riesgo. Nombre: “”. Descripción: “No hay suficientes computadoras para trabajar en el proyecto”. Probabilidad:”0.3”. Efecto: “Serios”. Descripción del impacto: “Retraso en la realización del proyecto”.</p>	
Resultados	El sistema muestra un mensaje de alerta (“Debe introducir el nombre”).
Condiciones	No se introduce el dato en el campo nombre.

Caso de uso	Gestionar riesgos(Insertar riesgos)
Caso de prueba	Adicionar un riesgo insertando números en el campo nombre.
Entrada	
<p>El líder introduce incorrectamente un dato necesario para adicionar un riesgo. Nombre: "1235". Descripción: "No hay suficientes computadoras para trabajar en el proyecto". Probabilidad:"0.3". Efecto: "Serios". Descripción del impacto: "Retraso en la realización del proyecto".</p>	
Resultados	El sistema muestra un mensaje de alerta ("Ha introducido caracteres no admisibles en el nombre").
Condiciones	No se introduce el dato en el campo nombre.

Escenario <Modificar riesgos>

Caso de uso	Gestionar riesgos(Modificar riesgos)
Caso de prueba	Modificar datos de un riesgo entrando correctamente los datos necesarios.
Entrada	
<p>El líder introduce correctamente los datos necesarios para modificar un riesgo. Nombre: "Insuficientes puestos de trabajo". Descripción: "No hay suficientes computadoras para trabajar en el proyecto". Probabilidad:"0.6". Efecto: "Serio". Descripción del impacto: "Retraso en la realización del proyecto".</p>	
Resultados	El sistema modifica los datos del riesgo seleccionado en la Base de Datos.
Condiciones	Se insertaron los datos correctos para modificar a su vez los datos correspondientes del riesgo seleccionado.

Caso de uso	Gestionar riesgos(Modificar riesgos)
Caso de prueba	Modificar un riesgo entrando incorrectamente algún dato.
Entrada	
<p>El líder introduce incorrectamente un dato para modificar un riesgo.</p> <p>Nombre: “Insuficientes puestos de trabajo”.</p> <p>Descripción: “No hay computadoras suficientes para trabajar en el proyecto”.</p> <p>Probabilidad:”20”.</p> <p>Efecto: “Catastróficos”.</p> <p>Descripción del impacto: “Retraso en la realización del proyecto”.</p>	
Resultados	El sistema muestra un mensaje de alerta notificando el error (“La probabilidad está en el rango de 0 a 1”).
Condiciones	Dato fuera del rango de valores permitidos.

4.3.3 Pruebas de Caja Negra para el caso de uso <Adicionar factor de riesgo>

Condiciones de Entrada	Casos Válidos	Casos no Válidos
Nombre	Nombre	Dejar el campo vacío. Insertar números.
Descripción	Descripción	Dejar el campo vacío.

Caso de uso	Adicionar factor de riesgo
Caso de prueba	Adicionar un factor de riesgo entrando los datos correctamente.
Entrada	
<p>El líder introduce correctamente los datos para adicionar un factor de riesgo.</p> <p>Nombre: ”Atraso en la búsqueda de permisos”</p> <p>Descripción:”Imposibilidad de gestionar el aspecto legal de la propiedad intelectual de las medias”.</p>	
Resultados	El sistema adiciona el factor de riesgo a la Base de Datos.
Condiciones	Datos del factor de riesgo entrados correctamente.

Caso de uso	Adicionar factor de riesgo
Caso de prueba	Adicionar un factor de riesgo dejando un campo vacío.
Entrada	
El líder introduce incorrectamente los datos para adicionar un factor de riesgo. Nombre: "".	
Descripción: "Imposibilidad de gestionar el aspecto legal de la propiedad intelectual de las medias".	
Resultados	El sistema muestra un mensaje de alerta notificando el error ("Debe introducir el nombre del factor de riesgo").
Condiciones	No se introduce el dato en el campo nombre.

4.3.4 Pruebas de Caja Negra para el caso de uso <Gestionar factor de riesgo>

Condiciones de Entrada	Casos Válidos	Casos no Válidos
Nombre	Nombre	Dejar el campo vacío. Insertar números.
Descripción	Descripción	Dejar el campo vacío.

Escenario <Modificar factor de riesgo >

Caso de uso	Gestionar factor de riesgo (Modificar factor de riesgo)
Caso de prueba	Modificar un factor de riesgo entrando los datos correctamente.
Entrada	
El administrador introduce correctamente los datos para modificar un factor de riesgo. Nombre: "Atraso en la búsqueda de permisos "	
Descripción: "Dificultad de gestionar el aspecto legal de la propiedad intelectual de las medias a utilizar"	
Resultados	El sistema modifica el factor de riesgo en la Base de Datos.
Condiciones	Datos del factor de riesgo insertados correctamente.

Caso de uso	Gestionar factor de riesgo (Modificar factor de riesgo)
Caso de prueba	Modificar un factor de riesgo insertando números en el campo nombre.
Entrada	
El administrador introduce incorrectamente los datos para adicionar un factor de riesgo. Nombre: "1234 "	
Descripción: " Dificultad de gestionar el aspecto legal de la propiedad intelectual de las medias".	
Resultados	El sistema muestra un mensaje de alerta ("Ha introducido caracteres no admisibles en el nombre").
Condiciones	Se introduce el dato en el campo nombre incorrectamente.

4.3.5 Pruebas de Caja Negra para el caso de uso <Gestionar relación riesgo_factor>

Condiciones de Entrada	Casos Válidos	Casos no Válidos
Riesgo	Seleccionar el riesgo.	No seleccionar el riesgo.
Factor de riesgo	Seleccionar el factor de riesgo.	No seleccionar el factor de riesgo.
Peso	Números del 0 al 1.	Dejar el campo vacío. Números fuera del rango de 0 a 1. Insertar letras.

Escenario <Insertar relación riesgo_factor >

Caso de uso	Gestionar relación riesgo_factor (Insertar relación riesgo_factor)
Caso de prueba	Adicionar el peso de la relación entrando los datos correctamente.
Entrada	
El líder introduce correctamente los datos para adicionar el peso de la relación entre el riesgo y el factor de riesgo. Riesgo: "Atraso en la fecha de entrega del producto". Factor de riesgo: "Atraso en la búsqueda de permisos".	

Peso: "0.5".	
Resultados	El sistema adiciona el peso de la relación entre el riesgo y el factor de riesgo a la Base de Datos.
Condiciones	Datos de la relación entrados correctamente.

Caso de uso	Gestionar relación riesgo_factor (Insertar relación riesgo_factor)
Caso de prueba	Adicionar el peso de la relación sin seleccionar un riesgo.
Entrada	
El líder no selecciona un riesgo. Riesgo: "-----". Factor de riesgo: " Atraso en la búsqueda de permisos". Peso: "0.5".	
Resultados	El sistema muestra un mensaje de alerta notificando el error ("Debe seleccionar un riesgo").
Condiciones	No se seleccionó un riesgo.

Caso de uso	Gestionar relación riesgo_factor (Insertar relación riesgo_factor)
Caso de prueba	Adicionar una relación insertando letras en el campo peso.
Entrada	
El líder introduce incorrectamente un dato necesario para adicionar una relación. Riesgo: "Atraso en la fecha de entrega del producto". Factor de riesgo: "Atraso en la búsqueda de permisos". Peso: "prueba".	
Resultados	El sistema muestra un mensaje de alerta ("Ha introducido caracteres no admisibles en el Peso de la Relación").
Condiciones	Se introduce incorrectamente el dato en el campo peso.

Escenario <Modificar relación riesgo_factor >

Caso de uso	Gestionar relación riesgo_factor (Modificar relación riesgo_factor)
Caso de prueba	Modificar el peso de la relación entrando los datos correctamente.
Entrada	
El líder introduce correctamente los datos para modificar el peso de la relación entre el riesgo y el factor de riesgo. Riesgo: " Atraso en la fecha de entrega del producto". Factor de riesgo: " Atraso en la búsqueda de permisos". Peso: "0.1".	
Resultados	El sistema modifica el peso de la relación entre el riesgo y el factor de riesgo en la Base de Datos.
Condiciones	Datos de la relación entrados correctamente.

Caso de uso	Gestionar relación riesgo_factor (Modificar relación riesgo_factor)
Caso de prueba	Modificar la relación dejando un campo obligatorio vacío.
Entrada	
El líder introduce incorrectamente el dato para modificar el peso de la relación entre el riesgo y el factor de riesgo. Riesgo: "Atraso en la fecha de entrega del producto". Factor de riesgo: "Atraso en la búsqueda de permisos". Peso: " ".	
Resultados	El sistema muestra un mensaje de alerta notificando el error ("Debe introducir el peso de la relación").
Condiciones	No se introduce el dato en el campo peso.

Caso de uso	Gestionar relación riesgo_factor (Modificar relación riesgo_factor)
Caso de prueba	Modificar la relación entrando incorrectamente los datos.
Entrada	
<p>El líder introduce incorrectamente el dato para modificar el peso de la relación entre el riesgo y el factor de riesgo.</p> <p>Riesgo:"Atraso en la fecha de entrega del producto".</p> <p>Factor de riesgo:"Atraso en la búsqueda de permisos".</p> <p>Peso:"20".</p>	
Resultados	El sistema muestra un mensaje de alerta notificando el error ("El peso está en el rango de 0 a 1").
Condiciones	Dato fuera del rango de valores permitidos.

4.4 Conclusiones parciales

- Se presentó cómo está construido el sistema a partir de los diagramas de componentes.
- También fue mostrado el diagrama de despliegue, el cual ilustra cuáles serán los nodos que serán usados para la implantación de la aplicación.
- Fueron detalladas un conjunto de pruebas que se le realizaron al sistema, donde se especifica qué parámetros fueron utilizados durante la realización de las mismas, además del resultado obtenido durante la ejecución de cada una de ellas.

CONCLUSIONES GENERALES

Después de concluir el presente trabajo y haber logrado el cumplimiento de las metas trazadas al inicio del desarrollo del mismo, se arribaron a las siguientes conclusiones:

- Las herramientas que utilizan en procesos de Gestión de Riesgos en el mundo no brindan un soporte adecuado para el modelo de gestión que se plantea para los proyectos de Software Educativo y Multimedia que se desarrollan en la UCI.
- Se recopilaron algunos de los problemas que presentan los proyectos de Software Educativo y Multimedia, lo que permitió almacenar datos reales en la Base de Datos de la aplicación, además se pudo constatar las deficiencias que presenta la fase de Identificación de Riesgos en estos proyectos.
- La selección de las herramientas y del lenguaje fue la adecuada para la implementación de la aplicación, de acuerdo a parámetros condicionados fundamentalmente por los requerimientos del software. Permittiéndose así, el desarrollo de una aplicación Web flexible que puede ser incorporada en otras herramientas de Gestión de Proyectos.
- Se realizó el análisis y diseño de la aplicación, generándose todos los artefactos requeridos según la metodología de desarrollo seleccionada.
- Se implementó el software conforme al alcance planteado en las fases anteriores, obteniendo una aplicación Web que se puede utilizar en la Gestión de Riesgos para proyectos de Software Educativo y Multimedia.
- Las pruebas realizadas permitieron validar la aplicación, corrigiéndose todos los errores encontrados y posibilitando la obtención de un producto de una mayor calidad.

RECOMENDACIONES

Luego de haber cumplido con los objetivos planteados y buscando la mejora continua del sistema, se recomienda:

- Desarrollar la implementación del módulo que permita Gestionar la Taxonomía.
- Ampliar el módulo de generación de reportes.
- Mejorar la interfaz visual para clasificar los riesgos de acuerdo a la Taxonomía.

REFERENCIAS BIBLIOGRÁFICAS

- ADRFORMACIÓN.COM. *Curso de PHP*, Disponible en:
http://www.adrformacion.com/curso/php/leccion1/Introduccion_php.htm.
- ALVAREZ. *Sistema gestor de base de datos*. 2006, nº Disponible en: <http://www.desarrolloweb.com/faq/504.php>.
- CANALES MORA. *Adictos al Trabajo*. 2004, nº Disponible en:
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=vparadigm>.
- FUENTE y LOVELLE. *Gestión de Riesgo*. 2006, nº Disponible en:
<http://www.di.uniovi.es/~aquilino/Asignaturas/ProyectosInformatica/Documentos/Proyectos.v2006.C7.V2.pdf>.
- FUTRELL, S. S. *Quality Software Project Management*. Editorial Prentice Hall, 2002. ISBN 0130912972.
- GILFILLAN. *La Biblia de MySQL Madrid*,. *Guía para el desarrollo de aplicaciones web seguras*, 2003, nº 84-415-558-1, Disponible en: <http://www.desarrolloweb.com/articulos/996.php>.
- LARMAN. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*, Disponible en:
<http://bibliodoc.uci.cu/pdf/reg00061.pdf>.
- MARCELO y RODENES. *Estudio exploratorio sobre los métodos de gestión de proyectos de alto riesgo*. 2003, vol. Primer Congreso Soporte del Conocimiento con la Tecnología, SOCOTE, nº
- MARVIN; CARR, *et al. Taxonomy-Based Risk Identification*. 1993, vol. E.-T.-.-CMU/SEI-93-TR-6, nº p. 90
- MOLPECERES. *Procesos de desarrollo: RUP, XP y FDD* Disponible en:
<http://www.javahispano.org/articles.article.action?id=76>.
- MOORE y FOX. *Documentación PHP-GTK*. 2006, nº Disponible en:
<http://gtk.php.net/manual1/es/html/intro.what.is.php.history.html>.
- MURCIA. *Gestión de riesgos en ingeniería del software*. Última actualización: 30 de diciembre del 2006, (Portal de la Universidad de Murcia). Disponible en: <http://www.um.es/docencia/barzana/IAGP/Iagp5.html>.
- PARADIGM, V. *Visual Paradigm for UML*, Disponible en: <http://www.visualparadigm.com/product/vpuml/>.
- PMI, P. M. I. *Guía de los Fundamentos de la Dirección de Proyectos*. Tercera edición ed. Editorial PMI, 2004.
- PRESSMAN, R. S. *Ingeniería del Software. Un enfoque Práctico* Quinta Edición. ed. Mc Graw-Hill/Interamericana de España, S.A, 2002.
- RICOTE y FIGUEREDO. *SGIF: Sistema de Gestión de la Información para la facultad. Instituto Superior Politécnico "José Antonio Echeverría"*. 2006, nº

- RODAS HINOSTROZA. *Características de PHP*. 2007, n° Disponible en:
<http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>.
- ROSENBERG; HAMMER, *et al.* *Continuous Risk Management at NASA*. 1999.
- RUMBAUHG; JACOBSON, *et al.* *El Lenguaje Unificado de Modelado. Manual de referencia.*, 2000a, n° 84-7829-037-0, Disponible en: <http://bibliodoc.uci.cu/pdf/reg03050.pdf>.
- . *El Proceso Unificado de Desarrollo de Software*. 2000b, n° Disponible en:
<http://bibliodoc.uci.cu/pdf/reg00060.pdf>.
- SEI, S. E. I. *Continuous Risk Management Guidebook*. Carnegie University, 2004, n°
- WEBSTER. Merriam-Webster's Collegiate Dictionary. 2004, n° ISSN 0877797099.
- ZULUETA VÉLIZ, Y. y DESPAIGNE HERRERA, E. *Modelo de Gestión de Riesgos en Proyectos de Desarrollo de Software. III Conferencia Científica de la Universidad de la Ciencias Informáticas*, 2007, n° ISSN 978-959-286-005-6.

GLOSARIO DE TÉRMINOS

ANSI: Principal organización encargada de promover el desarrollo de estándares tecnológicos en Estados Unidos (EL Instituto Nacional Estadounidense de Estándares), es miembro de la Organización Internacional para la Estandarización (ISO) y de la Comisión Electrotécnica Internacional (internacional Electrotechnical Commission, IEC).

BD: Base de Datos.

CASE: Ingeniería del software asistida por computadora.

CORBA: Es un estándar que establece una plataforma de desarrollo de sistemas distribuidos facilitando la invocación de métodos remotos bajo un paradigma orientado a objetos. (Common Object Request Broker Architecture).

C#: Lenguaje de programación orientado a objetos, evolución del lenguaje C++, desarrollado por Microsoft.

C++: Es un lenguaje híbrido, que se puede compilar y resulta más sencillo de aprender para los programadores que ya conocen C. Actualmente existe un estándar, denominado ISO C++, al que se han adherido la mayoría de los fabricantes de compiladores más modernos. Las principales características son abstracción (encapsulación), el soporte para programación orientada a objetos (polimorfismo) y el soporte de plantillas o programación genérica (templates). Es un lenguaje que abarca tres paradigmas de la programación: La programación estructurada, la programación genérica y la programación orientada a objetos.

GPL: Es una licencia creada por la Free Software Foundation y orientada principalmente a los términos de distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software Libre (General Public License).

HTML: Es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el estándar de las páginas Web. (Hyper Text Markup).

IDL: Lenguaje de especificación de interfaces que se usa como parte de la tecnología CORBA. Ofrece la sintaxis necesaria para definir los métodos que se quieren invocar remotamente. (Interface Definition Language).

Java: Lenguaje de programación orientado a objetos con el que se puede realizar cualquier tipo de programa, es un lenguaje muy extendido, es un lenguaje independiente de la plataforma, es compilado en un bytecode que es interpretado desarrollado por la compañía Sun Microsystems a principios de los 90.

MySQL: Sistema de administración de Base de Datos, es una de las bases de datos más populares desarrolladas bajo la filosofía de código abierto.

OOP: Programación Orientada a Objetos.

Oracle: Es un sistema de administración de Base de Datos, es una potente herramienta cliente/servidor para la gestión de bases de datos.

Perl: Lenguaje de programación desarrollado por Larry Wall inspirado en otras herramientas de UNIX.

PDF: Es el acrónimo del inglés (**P**ortable **D**ocument **F**ormat, formato de documento portátil) es un formato de almacenamiento de documentos, desarrollado por la empresa Adobe Systems. Este formato es de tipo compuesto (imagen vectorial, mapa de bits y texto).

PHP: Es acrónimo de Hipertext Pre-processor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación (Personal Home Page).

PMBOK: Guía de los Fundamentos de la Dirección de Proyectos.

PMI: Instituto de Gestión de Proyectos.

PostgreSQL: Servidor de Base de Datos relacional libre, liberado bajo la licencia BSD, es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL, Firebird y MaxDB), así como sistemas propietarios como Oracle o DB2.

Rational Rose: Herramienta CASE desarrollada por los creadores de UML (Booch, Rumbaugh y Jacobson), que cubre todo el ciclo de vida de un proyecto.

RBS: La Estructura de Desglose del Riesgo enumera las categorías y subcategorías de donde pueden surgir riesgos para un proyecto típico.

RUP: Proceso Unificado de Desarrollo.

SEI: Instituto de Ingeniería de Software es un organismo financiado por el gobierno federal de los Estados Unidos y operado por la universidad Carnegie Mellon de ese país. El SEI existe para ayudar a las organizaciones a mejorar el estado de las prácticas de ingeniería, con el fin de incrementar la calidad de los sistemas que dependen del software y la ingeniería de sistemas.

SGBD: Sistema Gestor de Bases de Datos.

Taxonomía: Las Taxonomías son las clasificaciones ordenadas de elementos de acuerdo a sus relaciones presumidas; y pueden emplearse como herramientas de suma utilidad en diferentes ramas de la ciencia y la industria donde se pretende organizar y facilitar el acceso a un número importante de elementos que se encuentran mutuamente relacionados de alguna manera relevante.

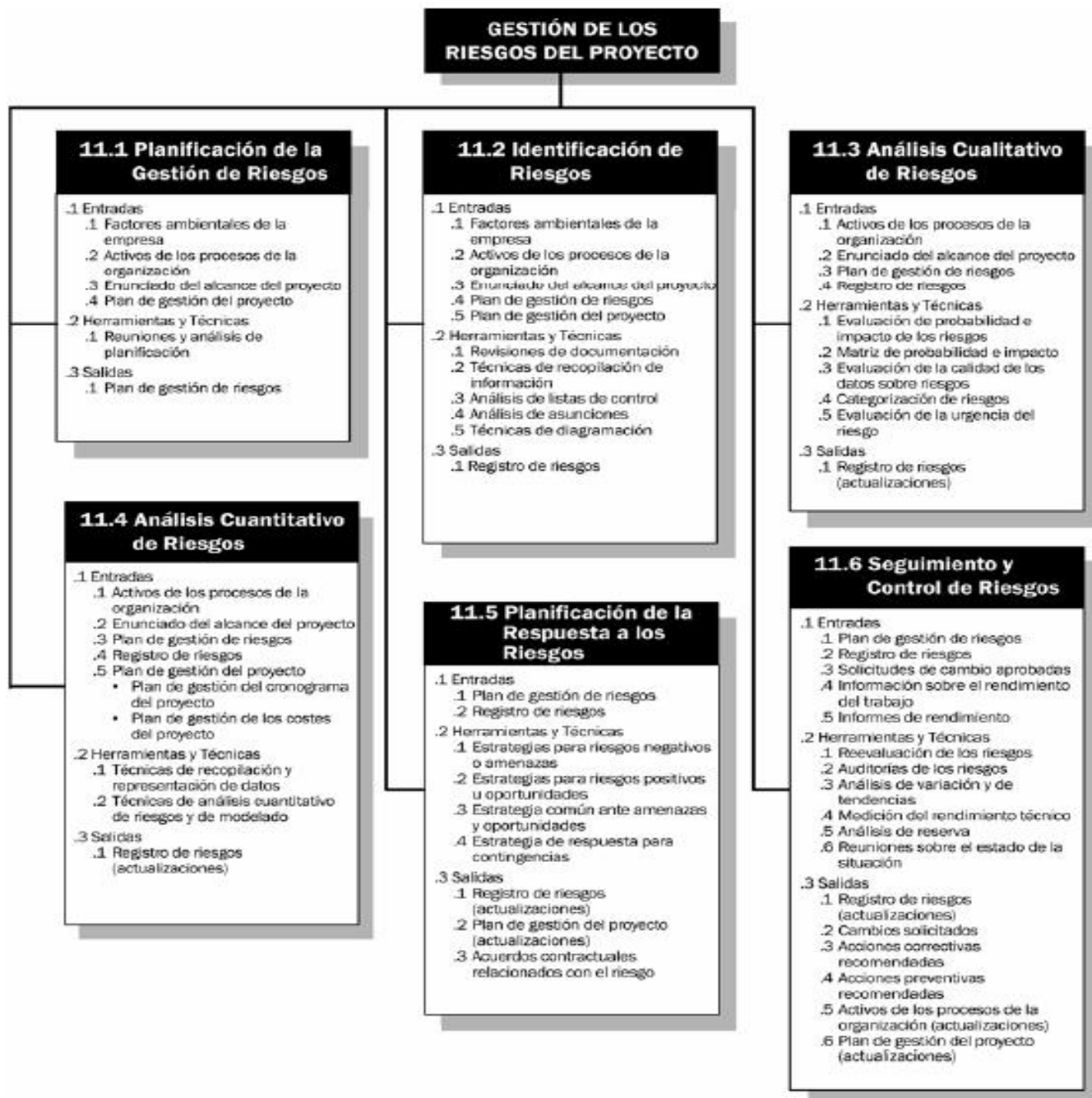
UML: Lenguaje Unificado de Modelado.

XML: Extensible Markup Language (Lenguaje extensible de etiquetas) Es un meta-lenguaje que permite definir lenguajes de marcado adecuado a usos determinados. Se propone como lenguaje de bajo nivel (a nivel de aplicación, no de programación) para intercambio de información estructurada entre diferentes plataformas.

XP: Programación Externa.

ANEXOS

Anexo 1: Descripción General de la Gestión de los Riesgos del Proyecto



Anexo 2: Prototipos de Interfaces de Usuarios de los casos de uso.

Prototipo de interfaz de usuario del caso de uso “Autenticar usuario”

Herramienta de Identificación de Riesgos

IR

INICIO DE SESIÓN

Usuario:

Contraseña:

Dominio: ▼

Entrar

Prototipos de interfaz de usuario del caso de uso “Gestionar riesgos”

Herramienta de Identificación de Riesgos

IR

Riesgos | Factor de Riesgo | Entorno de Trabajo | Relaciones | Reportes | Usuarios

Bienvenido "Juan Miguel" al Proyecto: "Historia Universal" Inicio | Salir


Listado de los Riesgos del Proyecto:

Nombre	Clasificación	Descripción	
Fallas con el fluido eléctrico	Instalaciones	Pérdidas de información por corte de electricidad o bajo voltaje	
Atraso en la fecha de entrega del producto	Estabilidad	Imposibilidad de cumplir con el plazo de entrega	
Cambios en el personal del proyecto	Personal	Cambios en los integrantes del proyectos por necesidades de la universidad	
Incumplimiento de los requerimientos	Personal	No se cumplen los requerimientos impuestos por el cliente	
Entrega por los clientes de los medios necesarios	Cooperación	Los clientes no entregan los medios necesarios para la realización del producto	

1

Insertar Riesgo

Sección<Insertar riesgos>

Herramienta de Identificación de Riesgos 

Riesgos | Factor de Riesgo | Entorno de Trabajo | Relaciones | Reportes | Usuarios

Bienvenido "Juan Miguel" al Proyecto: "Historia Universal" Inicio | Salir

Registro de Riesgo

Nombre:

Descripción:

Clases:

Clasificación: Elementos:

Atributos:

Probabilidad:

Efecto:

Descripción del Impacto:

Sección<Modificar riesgos>

Herramienta de Identificación de Riesgos 

Riesgos | Factor de Riesgo | Entorno de Trabajo | Relaciones | Reportes | Usuarios

Bienvenido "Juan Miguel" al Proyecto: "Multisaber" Inicio | Salir

Modificar el Riesgo

Nombre:

Descripción:

Clases:

Clasificación: Elementos:

Atributos:

Probabilidad:

Efecto:

Descripción del Impacto:

Prototipos de interfaz de usuario del caso de uso “Gestionar factor de riesgo”



Herramienta de Identificación de Riesgos

Riesgos | Factor de Riesgo | Entorno de Trabajo | Relaciones | Reportes | Usuarios

Bienvenido "Juan Miguel" al Proyecto: "Historia Universal" Inicio | Salir

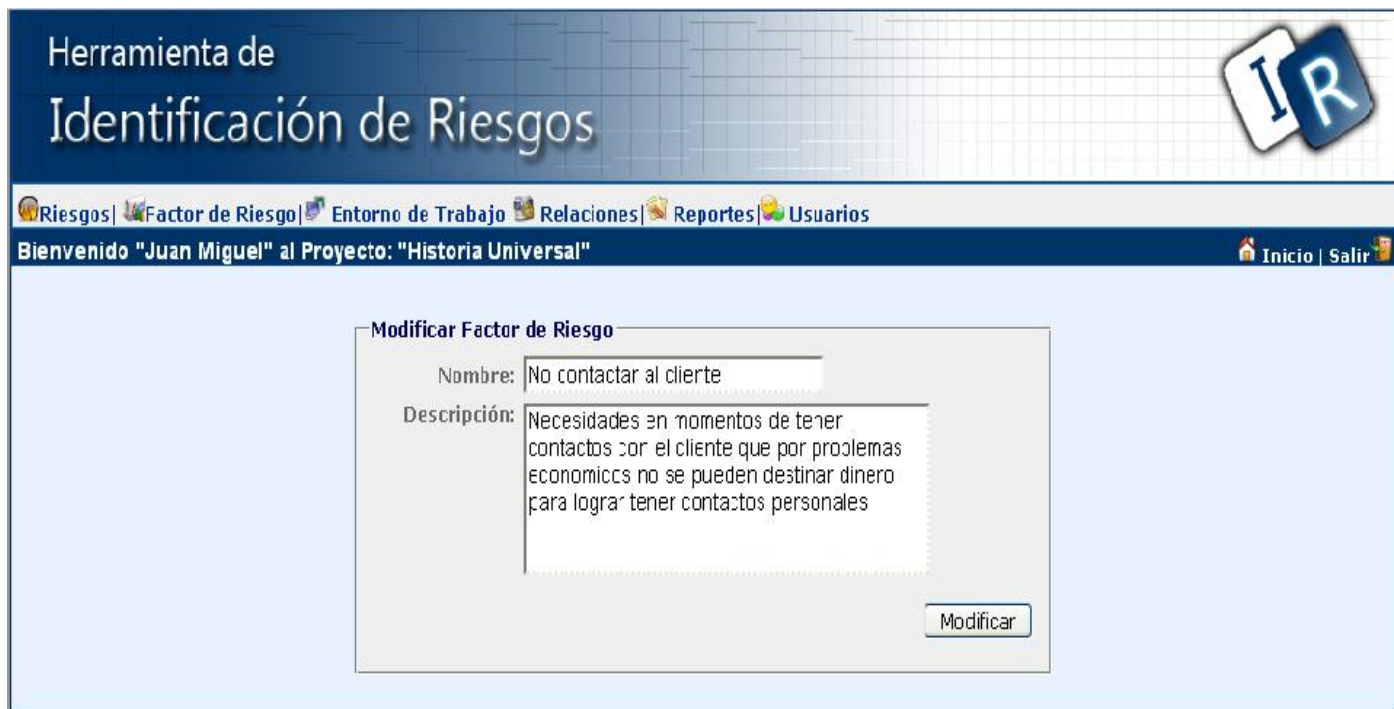
Listado General de los Factores de Riesgos:

Nombre	Descripción	
No contactar al cliente	Necesidades en momentos de tener contactos con el cliente que por problemas economicos no se pueden destinar dinero para lograr tener contactos personales	 
Clases de tercer año	En tercer año hay etapas en que la carga docente imposibilita a los estudiantes dedicarse a otras tareas	 

1

[Insertar Factor de Riesgo](#)

Sección <Modificar factor de riesgo>



Herramienta de Identificación de Riesgos

Riesgos | Factor de Riesgo | Entorno de Trabajo | Relaciones | Reportes | Usuarios

Bienvenido "Juan Miguel" al Proyecto: "Historia Universal" Inicio | Salir

Modificar Factor de Riesgo

Nombre:

Descripción:

Prototipo de interfaz de usuario del caso de uso “Adicionar factor de riesgo”

Herramienta de Identificación de Riesgos

Riesgos | Factor de Riesgo | Entorno de Trabajo | Relaciones | Reportes | Usuarios

Bienvenido "Juan Miguel" al Proyecto: "Historia Universal" Inicio | Salir

Registrar Factor de Riesgo

Nombre:

Descripción:

Crear

Prototipos de interfaz de usuario del caso de uso “Gestionar entorno de trabajo”

Herramienta de Identificación de Riesgos

Riesgos | Factor de Riesgo | Entorno de Trabajo | Relaciones | Reportes | Usuarios

Bienvenido "Juan Miguel" al Proyecto: "Historia Universal" Inicio | Salir

Listado General de los Entornos de Trabajo:

Nombre del Entorno	Descripción		
Edificio	Es el edificio o docente donde se encuentra el proyecto		
Laboratoric	La toratorio donde trabaja el proyecto		
Cantidad estudiantes	Son los estudiantes que van a trabajar en el proyecto		

1

Insertar Entorno de Trabajo y sus Valores

Sección <Modificar entorno de trabajo>

Herramienta de Identificación de Riesgos

Riesgos | Factor de Riesgo | Entorno de Trabajo | Relaciones | Reportes | Usuarios

Bienvenido "Juan Miguel" al Proyecto: "Historia Universal" Inicio | Salir

Modificar el Entorno de Trabajo

Nombre del Entorno: Edificio Descripción: Es el edificio o docente donde se encuentra el proyecto

Modificar

Asignar Valores al Entorno de Trabajo

Entornos Existentes: [dropdown] Valor: [input] Adicionar

Entorno de Trabajo y sus Valores

Entorno de Trabajo	Valor	
Edificio	Docente 5	[icon] [icon]
Cantidad estudiantes	15	[icon] [icon]

1

Prototipo de interfaz de usuario del caso de uso "Adicionar entorno de trabajo"

Herramienta de Identificación de Riesgos

Riesgos | Factor de Riesgo | Entorno de Trabajo | Relaciones | Reportes | Usuarios

Bienvenido "Juan Miguel" al Proyecto: "Historia Universal" Inicio | Salir

Crear un Nuevo Entorno de Trabajo

Nombre del Entorno: [input] Descripción: [input] Crear

Asignar Valores al Entorno de Trabajo

Entornos Existentes: [dropdown] Valor: [input] Adicionar

Entorno de Trabajo y sus Valores

Entorno de Trabajo	Valor	
Edificio	Docente 5	[icon] [icon]
Cantidad estudiantes	15	[icon] [icon]

1

Prototipos de interfaz de usuario del caso de uso “Gestionar valor entorno de trabajo”
 Sección<Insertar valor entorno de trabajo>

Herramienta de Identificación de Riesgos

Riesgos | Factor de Riesgo | Entorno de Trabajo | Relaciones | Reportes | Usuarios

Bienvenido "Juan Miguel" al Proyecto: "Historia Universal" Inicio | Salir

Crear un Nuevo Entorno de Trabajo

Nombre del Entorno: Descripción:

Crear

Asignar Valores al Entorno de Trabajo

Entornos Existentes: Valor:

Adicionar

Ecificio
 Laboratorio
 Cantidad estudiantes

Entorno de Trabajo y sus Valores

Entorno de Trabajo	Valor		
Edificio	Docente 5		
Cantidad estudiantes	15		

1

Sección<Modificar valor entorno de trabajo>

Herramienta de Identificación de Riesgos

Riesgos | Factor de Riesgo | Entorno de Trabajo | Relaciones | Reportes | Usuarios

Bienvenido "Juan Miguel" al Proyecto: "Historia Universal" Inicio | Salir

Crear un Nuevo Entorno de Trabajo

Nombre del Entorno: Descripción:

Crear

Modificar el Valor

Entornos Existentes: Valor:

Modificar

Ecificio

Entorno de Trabajo y sus Valores

Entorno de Trabajo	Valor		
Edificio	Docente 5		
Cantidad estudiantes	15		

1

Prototipos de interfaz de usuario del caso de uso “Gestionar relación riesgo_factor”
 Sección< Insertar riesgo_factor >

Sección<Modificar relación riesgo_factor>

Riesgo	Factor Riesgo	Peso		
No le guste el producto a los usuarios finales	No contactar al cliente	0.5		

Sección<Modificar relación riesgo_factor>

Herramienta de
Identificación de Riesgos

Riesgos| Factor de Riesgo| Entorno de Trabajo| Relaciones| Reportes| Usuarios

Bienvenido "Adriana" al Proyecto: "Curso de Español" Inicio| Salir

Relación Riesgo y Factor Riesgo

Riesgo: No le guste el producto a los usuarios finales

Factor de Riesgo: No cctactar al cliente

Peso: 0.5

Modificar

Prototipos de interfaz de usuario del caso de uso "Gestionar relación riesgo_entorno"
Sección<Insertar relación riesgo_entorno>

Herramienta de
Identificación de Riesgos

Riesgos| Factor de Riesgo| Entorno de Trabajo| Relaciones| Reportes| Usuarios

Bienvenido "Adriana" al Proyecto: "Curso de Español" Inicio| Salir

Relación Riesgo y Entorno de Trabajo

Riesgo:

Entorno de Trabajo:

Peso:

Crear

Sección<Modificar relación riesgo_entorno>

Herramienta de
Identificación de Riesgos

Riesgos | Factor de Riesgo | Entorno de Trabajo | Relaciones | Reportes | Usuarios

Bienvenido "" al Proyecto: "Curso de Español" Inicio | Salir

Relación Riesgo y Entorno de Trabajo

Riesgo:

Entorno de Trabajo:

Peso:

Modificar

Prototipos de interfaz de usuario del caso de uso "Gestionar relación factor_entorno"
Sección<Insertar relación factor_entorno>

Herramienta de
Identificación de Riesgos

Riesgos | Factor de Riesgo | Entorno de Trabajo | Relaciones | Reportes | Usuarios

Bienvenido "Adriana" al Proyecto: "Curso de Español" Inicio | Salir

Relación Factor Riesgo y Entorno de Trabajo

Factor de Riesgo:

Entorno de Trabajo:

Peso:

Crear

Sección <Modificar relación factor_entorno>

Herramienta de
Identificación de Riesgos

Riesgos | Factor de Riesgo | Entorno de Trabajo | Relaciones | Reportes | Usuarios

Bienvenido "Juan Miguel" al Proyecto: "Historia Universal" Inicio | Salir

Relación Factor Riesgo y Entorno de Trabajo

Factor Riesgo: Clases de tercer año

Entorno de Trabajo: Cantidad de estudiantes de tercero

Peso: C.4

Modificar

Prototipos de interfaz de usuario del caso de uso "Gestionar usuarios"

Herramienta de
Identificación de Riesgos

Riesgos | Factor de Riesgo | Entorno de Trabajo | Relaciones | Reportes | Usuarios

Bienvenido "Juan Miguel" al Proyecto: "Historia Universal" Inicio | Salir

Listado de los Usuarios del Proyecto:

Nombre	Usuario	Departamento	Rol	
Yusel	yariasguerra	Matemática	Usuario	  
Humberto	hmartin	Desarrolladores	Usuario	  
Adriana	aludeiro	Gestión	Lider	  
Yulaine	yaguerra	Diseño	Lider	  
Yanet	yarman	Calidad	Usuario	  

1

Insertar Usuario

Sección <Insertar usuarios>

Herramienta de
Identificación de Riesgos

Riesgos | Factor de Riesgo | Entorno de Trabajo | Relaciones | Reportes | Usuarios

Bienvenido "Juan Miguel" al Proyecto: "Historia Universal" Inicio | Salir

Registro de Usuario

Tipo de Usuario:

Nombre:

Usuario:

Contraseña:

Repetir Contraseña:

Proyecto:

Rol:

Departamento al que Pertenece:

Crear

Sección <Insertar usuarios>

Herramienta de
Identificación de Riesgos

Riesgos | Factor de Riesgo | Entorno de Trabajo | Relaciones | Reportes | Usuarios

Bienvenido "Juan Miguel" al Proyecto: "Historia Universal" Inicio | Salir

Registro de Usuario

Tipo de Usuario:

Nombre:

Usuario:

Contraseña:

Repetir Contraseña:

Proyecto:


Rol:

Departamento al que Pertenece:

Crear

Sección <Modificar usuarios>

Herramienta de
Identificación de Riesgos



Riesgos | Factor de Riesgo | Entorno de Trabajo | Relaciones | Reportes | Usuarios

Bienvenido "Juan Miguel" al Proyecto: "Historia Universal" [Inicio](#) | [Salir](#)


Modificar Usuario

Nombre: Yanet
Usuario: yamar
Proyecto: Historia Uriversa
Rol: Usuario
Departamento: Calidad

Modificar

Sección <Modificar usuarios>

Herramienta de
Identificación de Riesgos



Riesgos | Factor de Riesgo | Entorno de Trabajo | Relaciones | Reportes | Usuarios

Bienvenido "Juan Miguel" al Proyecto: "Historia Universal" [Inicio](#) | [Salir](#)

Modificar Usuario

Nombre: Elizabeth
Usuario: eludeiro
Contraseña:
Nueva Contraseña:
Repetir Contraseña:
Proyecto: Historia Universal
Rol: Usuario
Departamento: Soporte

Modificar

Sección <Asignar permisos>

Herramienta de Identificación de Riesgos

Riesgos | Factor de Riesgo | Entorno de Trabajo | Relaciones | Reportes | Usuarios

Bienvenido "Juan Miguel" al Proyecto: "Historia Universal" Inicio | Salir

Asignar Permisos

Nombre: Adriana

Usuario: aludeiro

Proyecto: Historia Universal

Rol: Lider

Departamento al que Pertenece: Gestión

Crear

Prototipos de interfaz de usuario del caso de uso "Notificar disparo"
Sección <Notificar disparo riesgo>

Herramienta de Identificación de Riesgos

Riesgos | Factor de Riesgo | Entorno de Trabajo | Relaciones | Reportes | Usuarios

Bienvenido "Juan Miguel" al Proyecto: "Historia Universal" Inicio | Salir

Fecha de Disparo de un Riesgo

Riesgo: Fallas con el flu do eléctrico

Fecha de disparo:

Guardar

mayo, 2009

L	M	M	J	V	S	D
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Tcday: sábado 30 de mayo de 2009

Sección <Notificar disparo factor de riesgo>

Herramienta de
Identificación de Riesgos

Riesgos | Factor de Riesgo | Entorno de Trabajo | Relaciones | Reportes | Usuarios

Bienvenido "Juan Miguel" al Proyecto: "Historia Universal" Inicio | Salir

Fecha de Disparo de un Factor de Riesgo:

Factor de Riesgo: No contactar al cliente Fecha de disparo: 13/05/2009 Guardar

Prototipos de interfaz de usuario del caso de uso "Generar reportes"

Herramienta de
Identificación de Riesgos

Riesgos | Factor de Riesgo | Entorno de Trabajo | Relaciones | Reportes | Usuarios

Bienvenido "Juan Miguel" al Proyecto: "Historia Universal" Inicio | Salir

Reportes

Tipo: Riesgo Proyectos: Multisabe

Parámetros: Clasificación Valores: Personal

Crear Reporte

Herramienta de
Identificación de Riesgos



Riesgos | Factor de Riesgo | Entorno de Trabajo | Relaciones | Reportes | Usuarios

Bienvenido "Juan Miguel" al Proyecto: "Multisaber" Inicio | Salir

Listado del Reporte:

Nombre	Clasificación	Proyecto	Efecto	Probabilidad	Fecha de Disparo
Pérdida de información	Personal	Multisaber	Catastróficos	0.3	2009-05-06
Personal con problemas de salud	Personal	Multisaber	Tolerable	0.2	No se ha Disparado
Sobrecarga de actividades ajenas al proyecto	Personal	Multisaber	Tolerable	0.5	No se ha Disparado
No existen expertos disponibles	Personal	Multisaber	Tolerable	0.3	No se ha Disparado

Exportar a PDF 

Prototipo de interfaz de usuario del caso de uso "Exportar a PDF"

Plantilla de Reportes

Listado del Reporte:

Nombre	Clasificación	Proyecto	Efecto	Probabilidad	Fecha de Disparo
Pérdida de información	Personal	Multisaber	Catastróficos	0.3	2009-05-06
Personal con problemas de salud	Personal	Multisaber	Tolerable	0.2	No se ha Disparado
Sobrecarga de actividades ajenas al proyecto	Personal	Multisaber	Tolerable	0.5	No se ha Disparado
No existen expertos disponibles	Personal	Multisaber	Tolerable	0.3	No se ha Disparado

Anexo 3: Diagramas de clases del análisis.

Diagrama de clases del análisis del CU<Autenticar usuario>

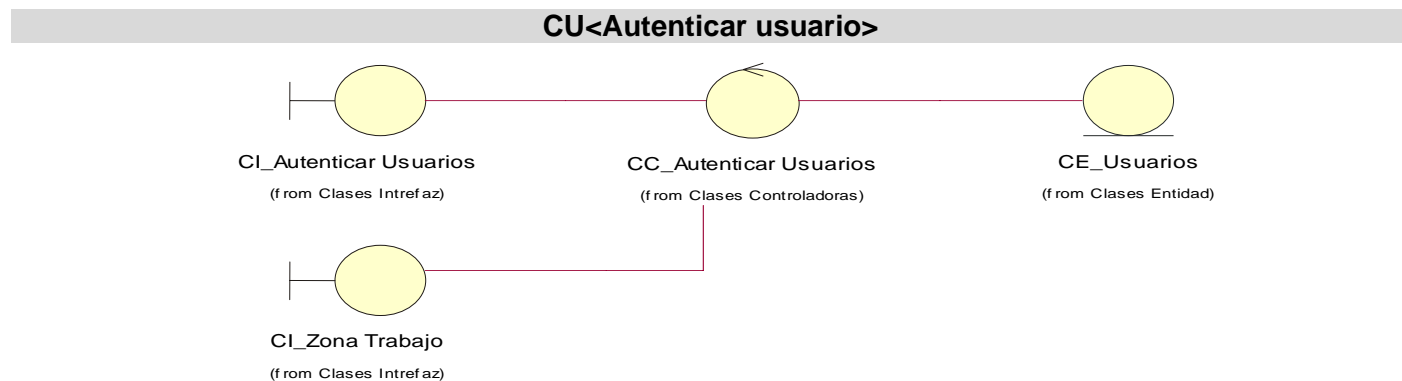


Diagrama de clases del análisis del CU< Adicionar factor de riesgo >

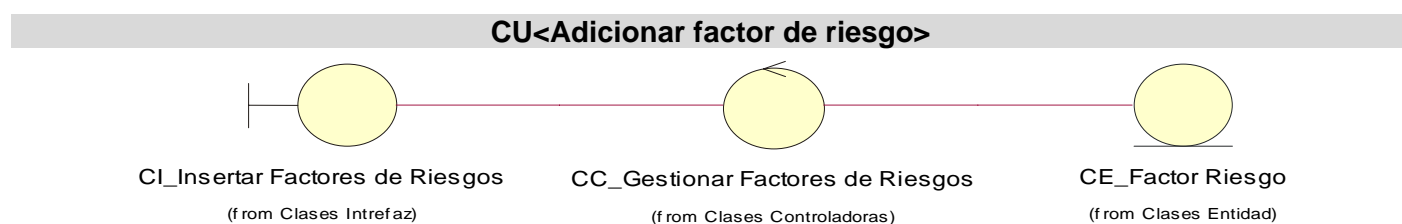


Diagrama de clases del análisis del CU< Gestionar factor de riesgo >

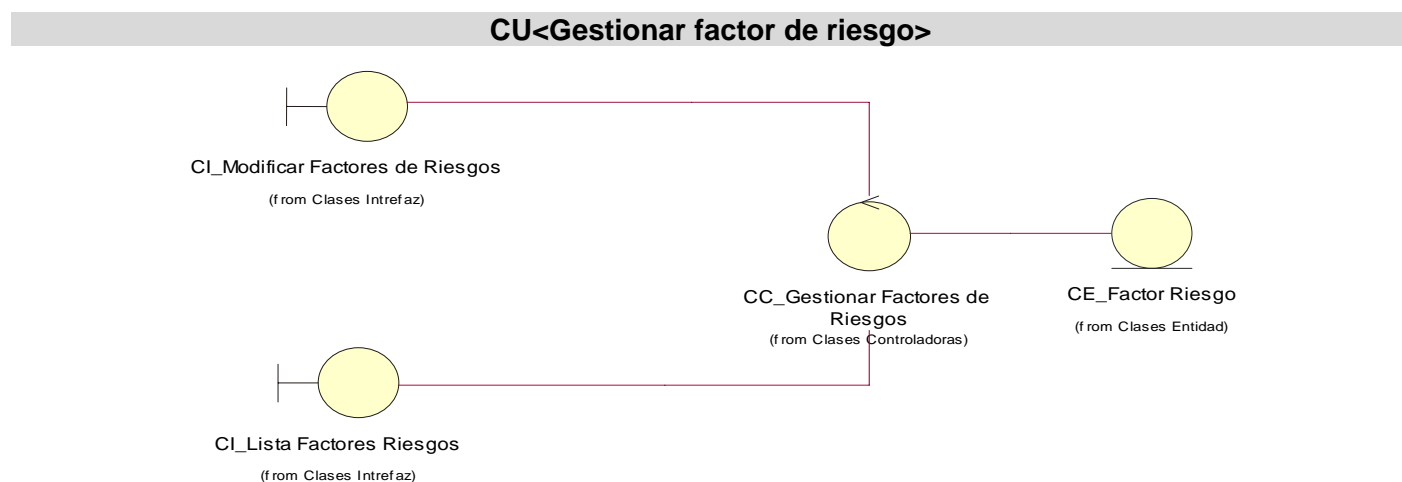


Diagrama de clases del análisis del CU< Gestionar relación riesgo_factor>

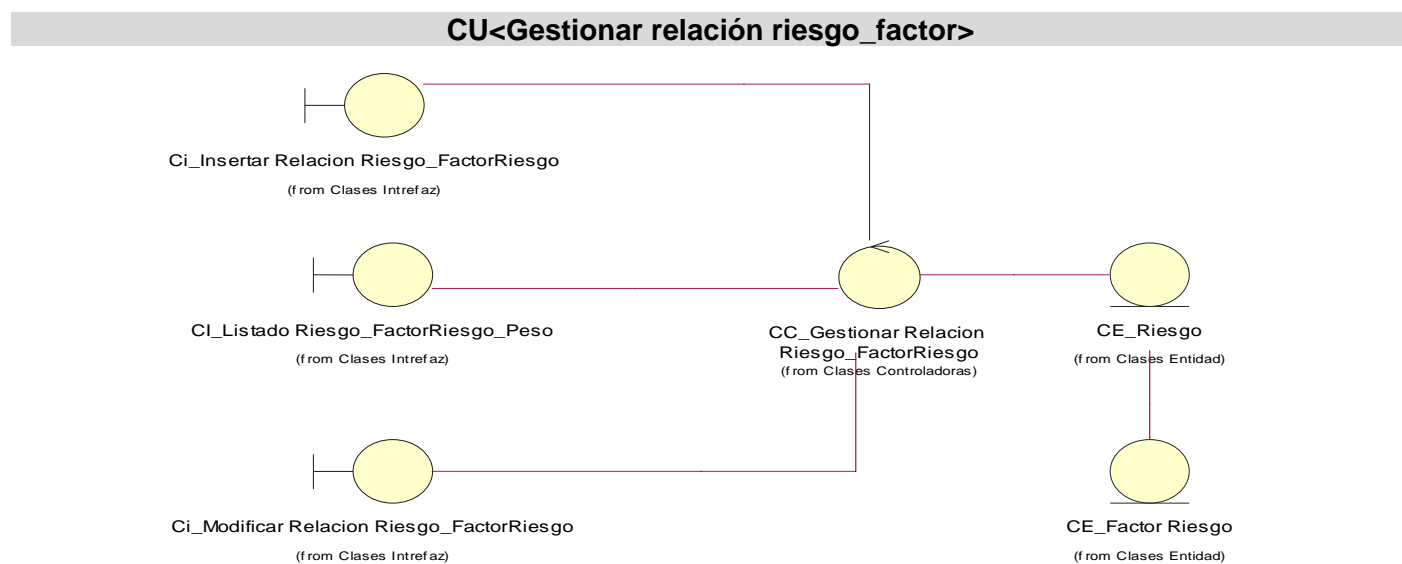


Diagrama de clases del análisis del CU< Adicionar entorno de trabajo >

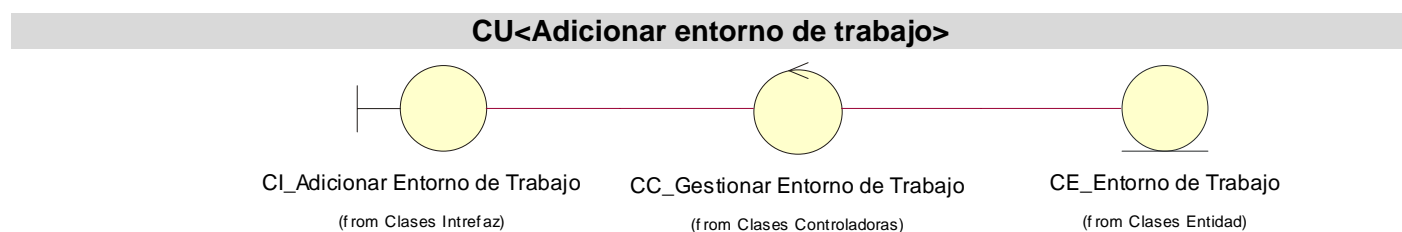


Diagrama de clases del análisis del CU< Gestionar entorno de trabajo >

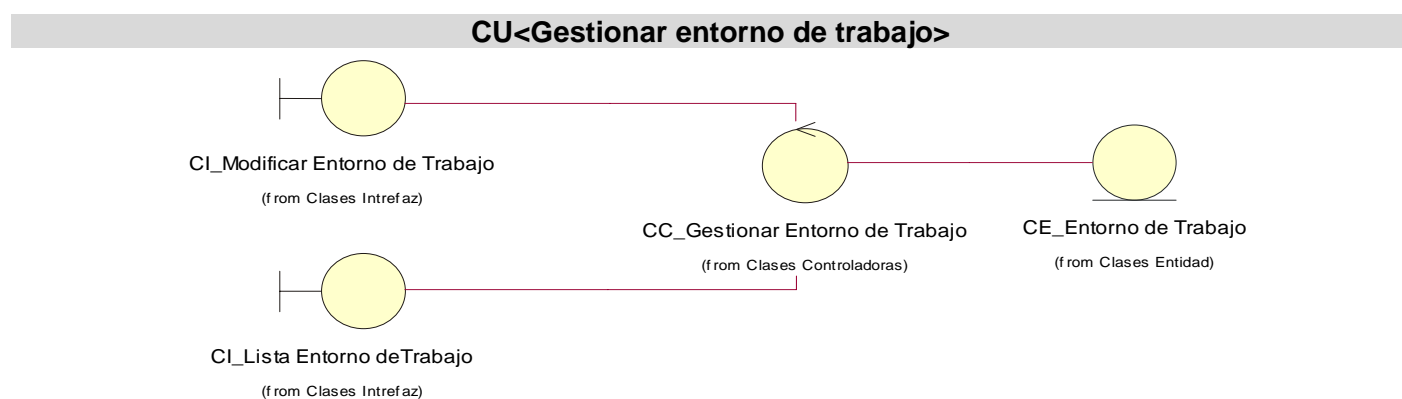


Diagrama de clases del análisis del CU< Gestionar relación riesgo_entorno >

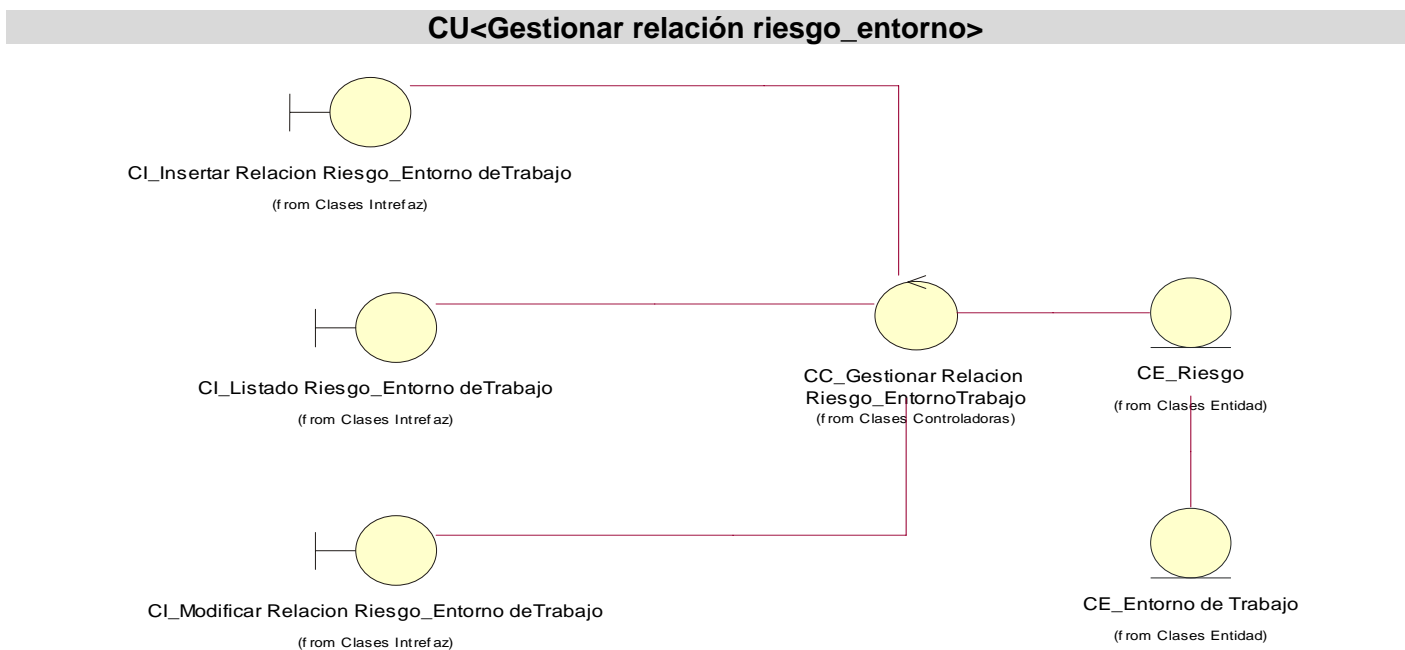


Diagrama de clases del análisis del CU<Gestionar relación factor_entorno >

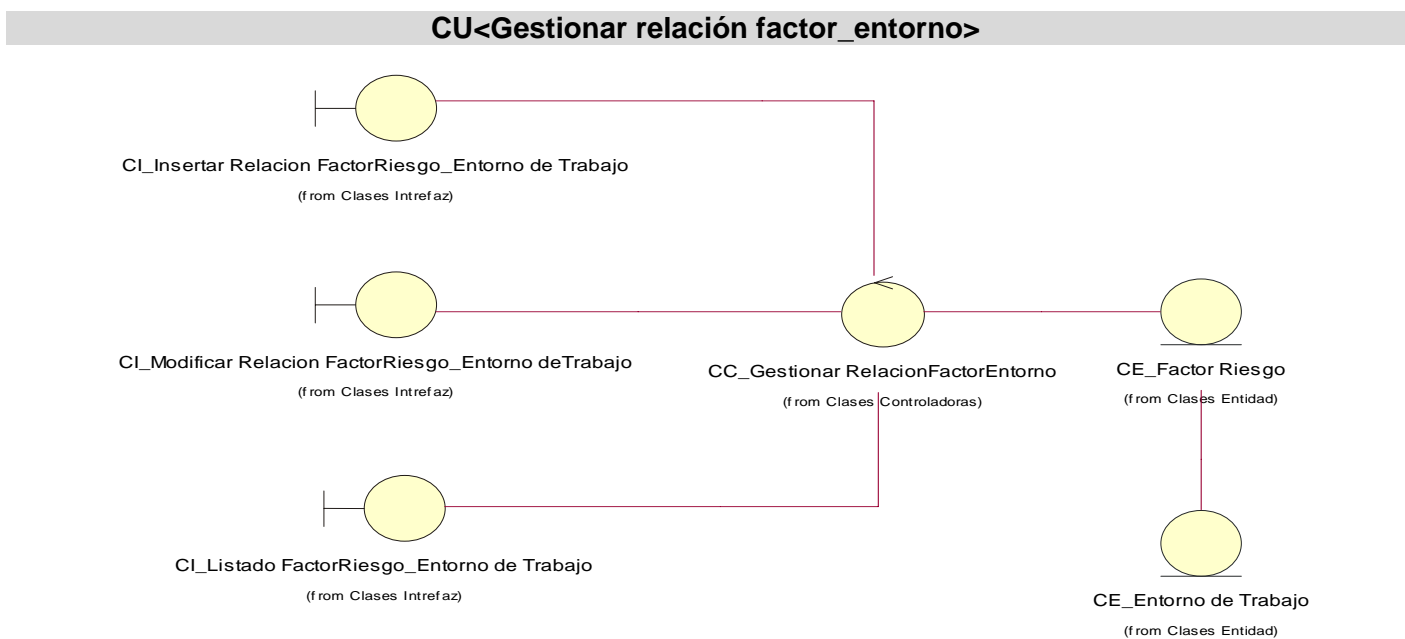


Diagrama de clases del análisis del CU< Gestionar valor entorno de trabajo >

CU<Gestionar valor entorno de trabajo>

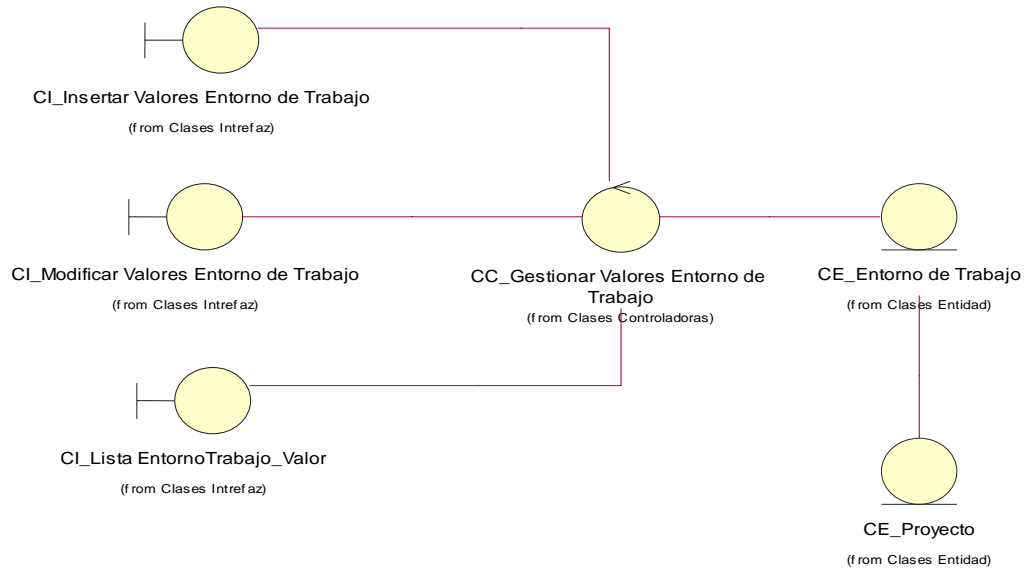


Diagrama de clases del análisis del CU< Notificar disparo>

CU<Notificar disparo>

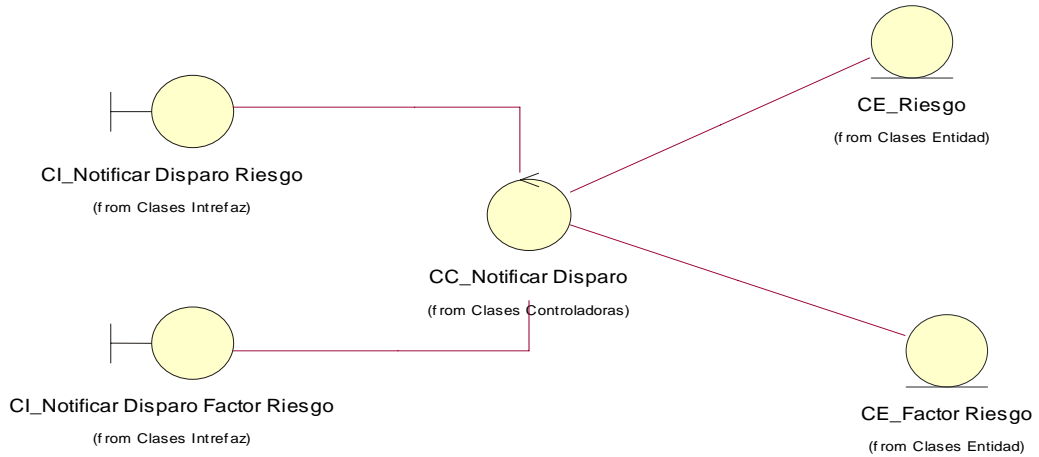


Diagrama de clases del análisis del CU<Gestionar usuarios>

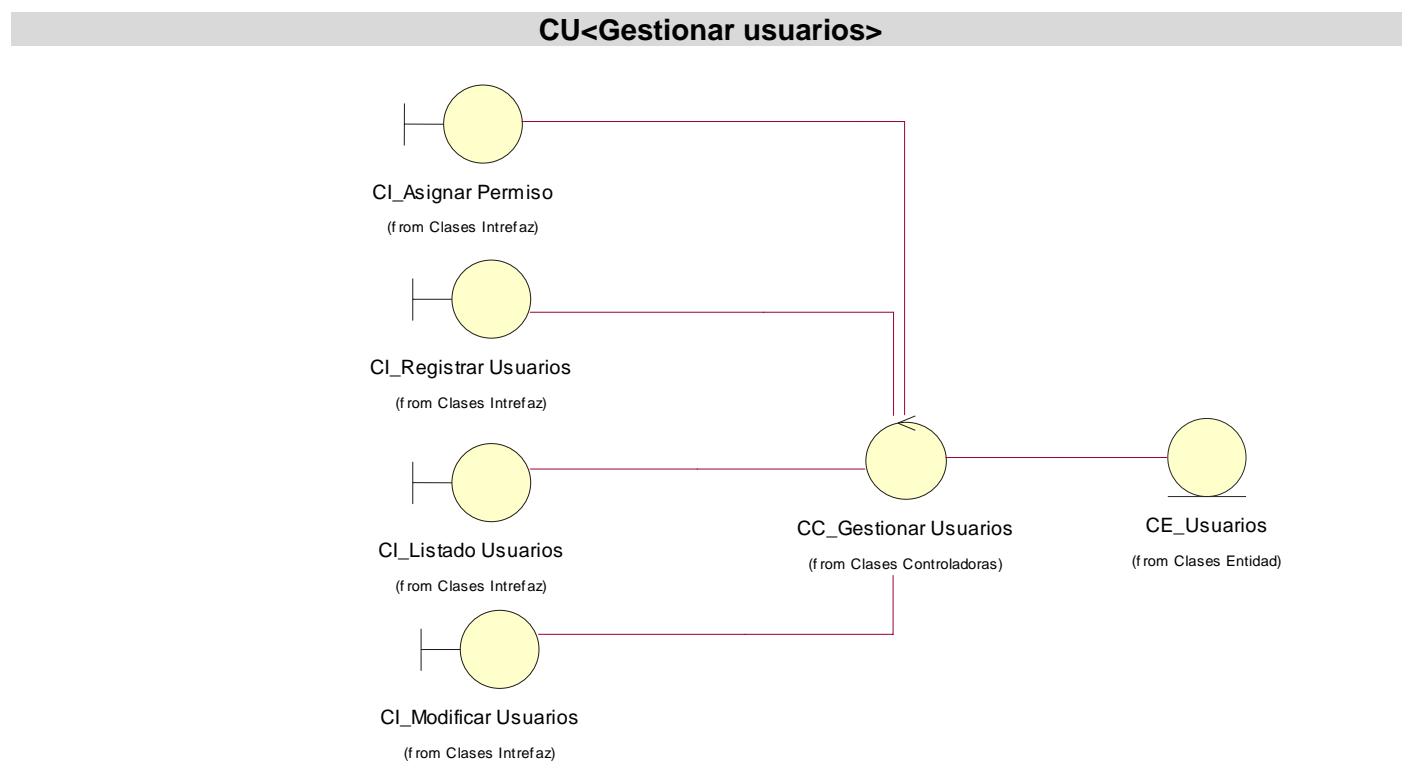


Diagrama de clases del análisis del CU< Generar reportes >

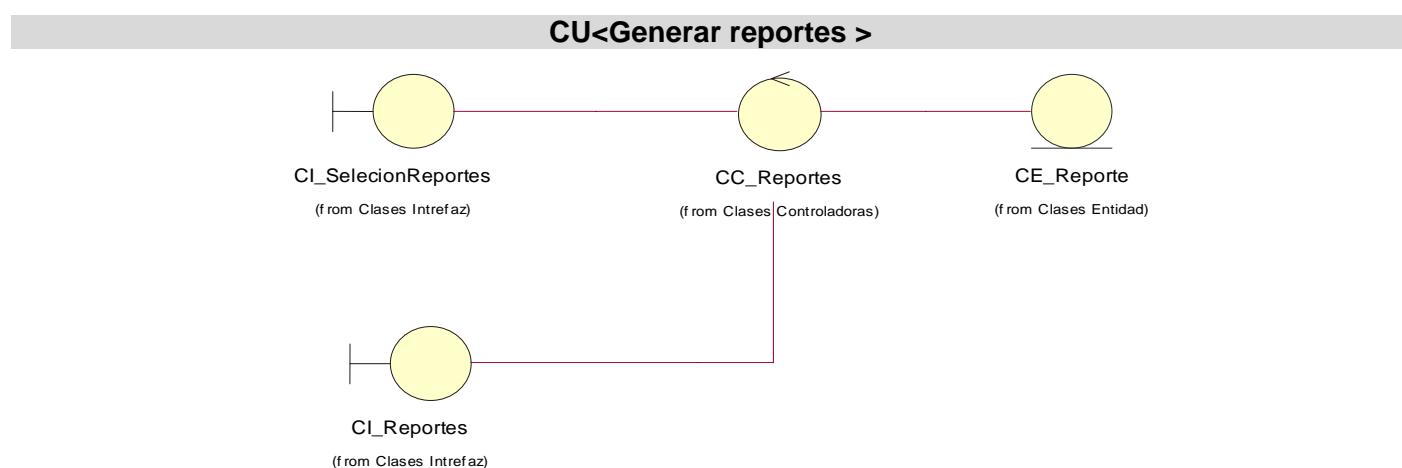
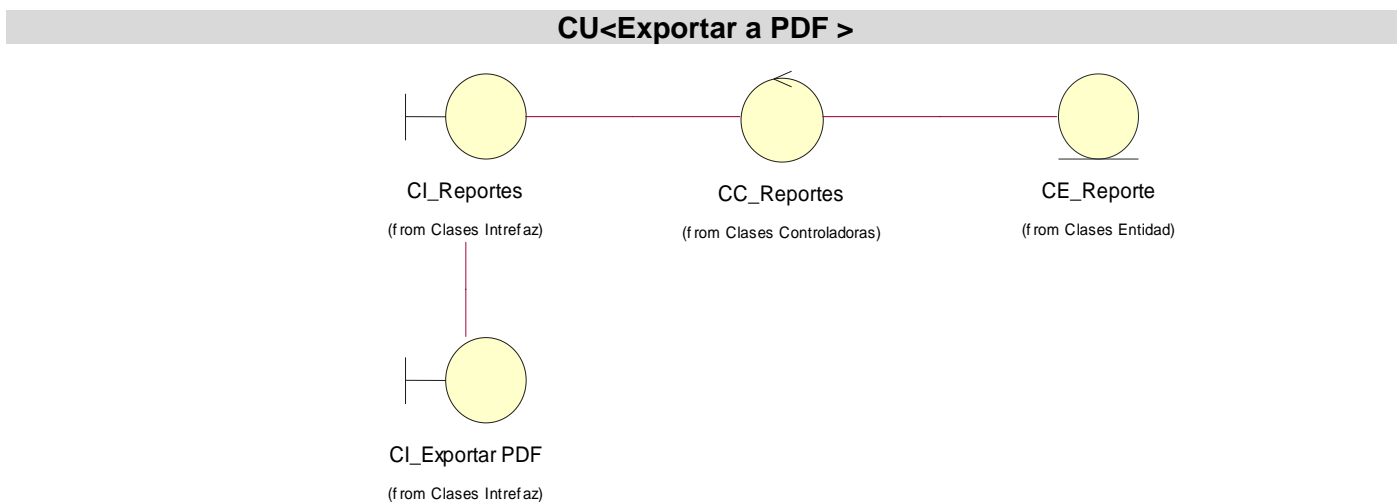


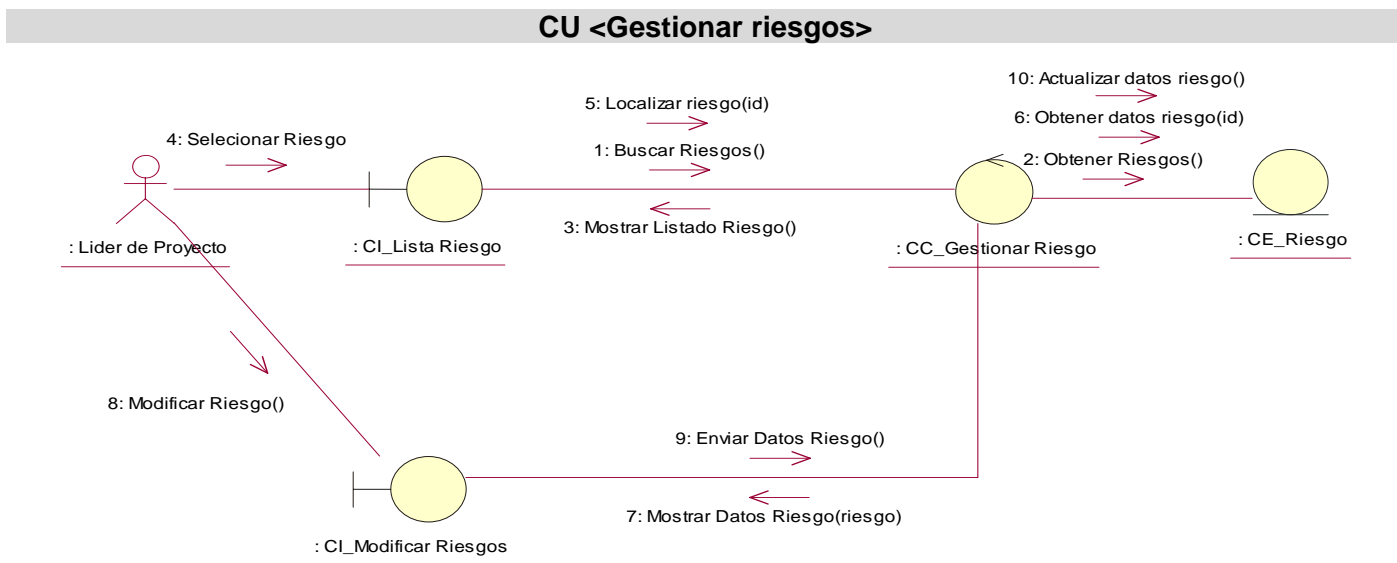
Diagrama de clases del análisis del CU< Exportar a PDF >



Anexo 4: Diagramas de Colaboración.

Diagramas de Colaboración <Gestionar riesgos>

Escenario <Modificar riesgos>



Escenario < Eliminar riesgos >

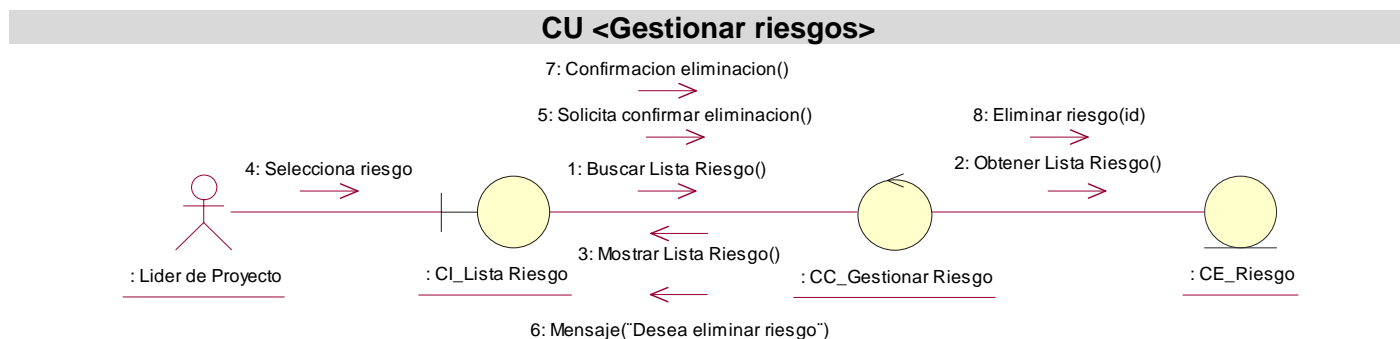
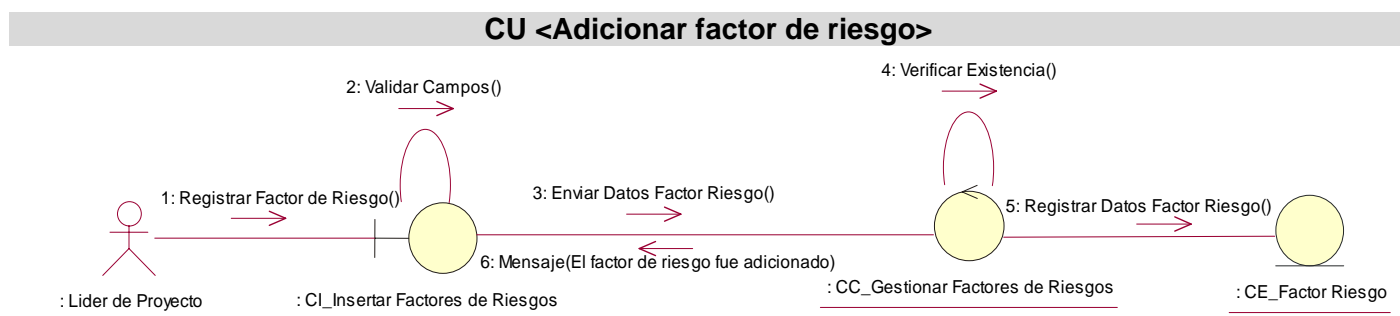
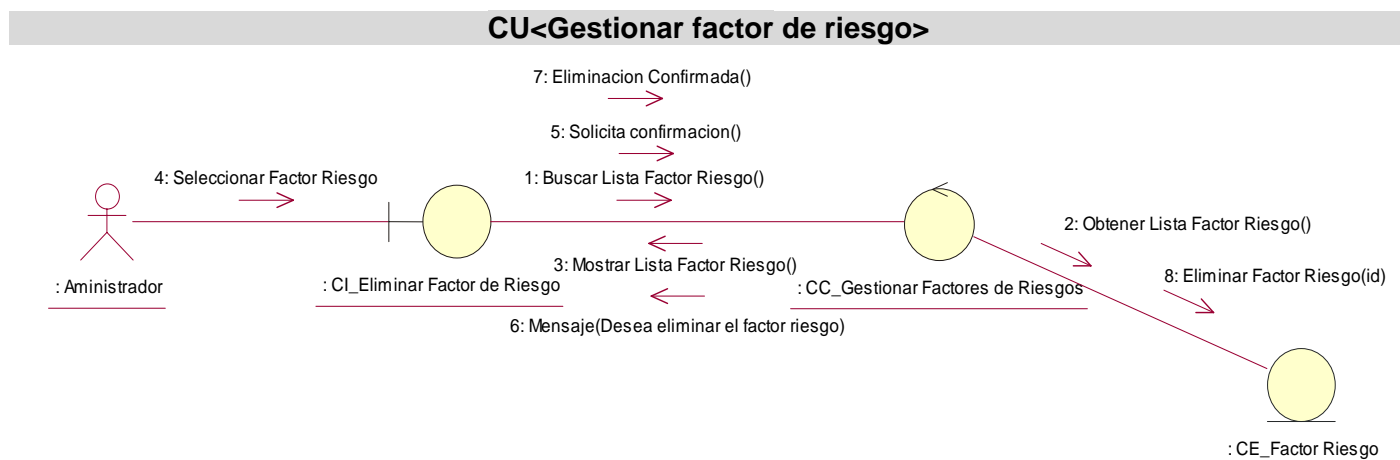


Diagrama de Colaboración <Adicionar factor de riesgo>



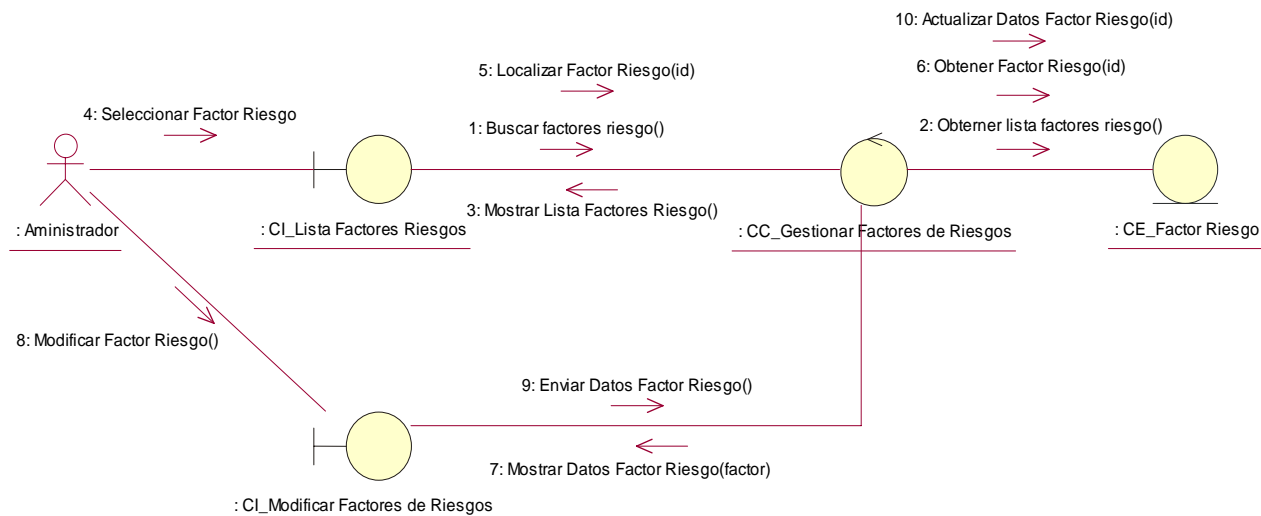
Diagramas de Colaboración <Gestionar factor de riesgo>

Escenario <Eliminar factor de riesgo>



Escenario <Modificar factor de riesgo>

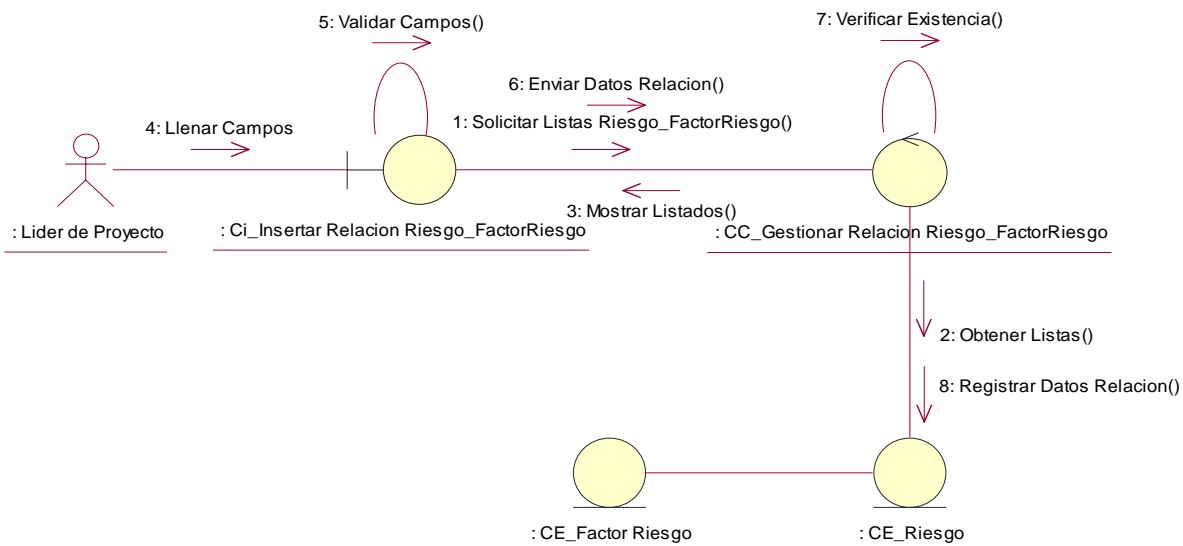
CU<Gestionar factor de riesgo>



Diagramas de Colaboración <Gestionar relación riesgo_factor>

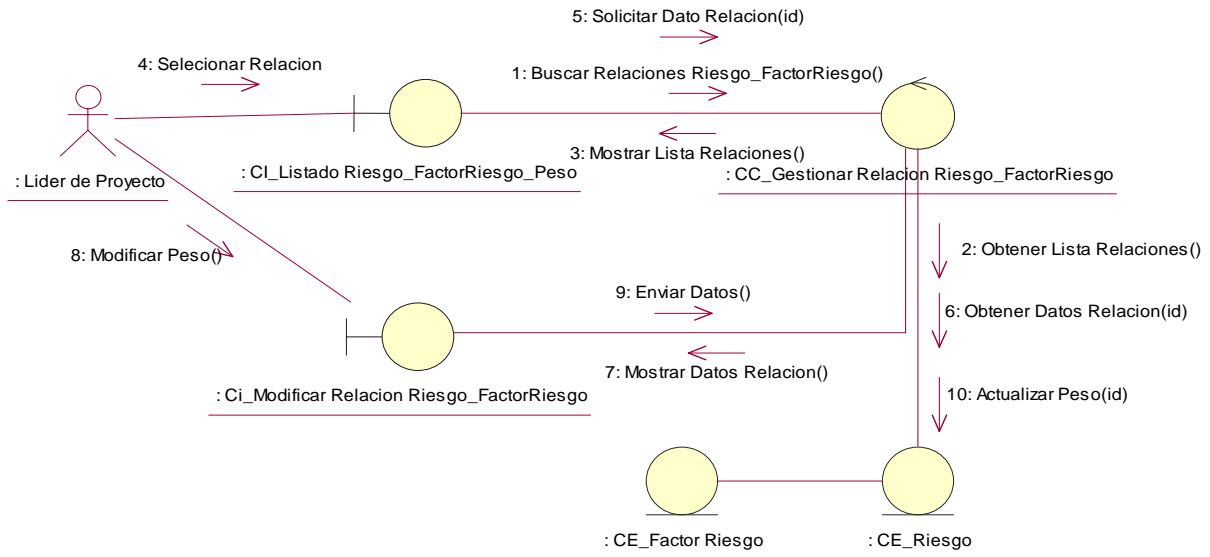
Escenario <Insertar relación riesgo_factor>

CU<Gestionar relación riesgo_factor>



Escenario <Modificar relación riesgo_factor>

CU<Gestionar relación riesgo_factor>



Escenario <Eliminar relación riesgo_factor>

CU<Gestionar relación riesgo_factor>

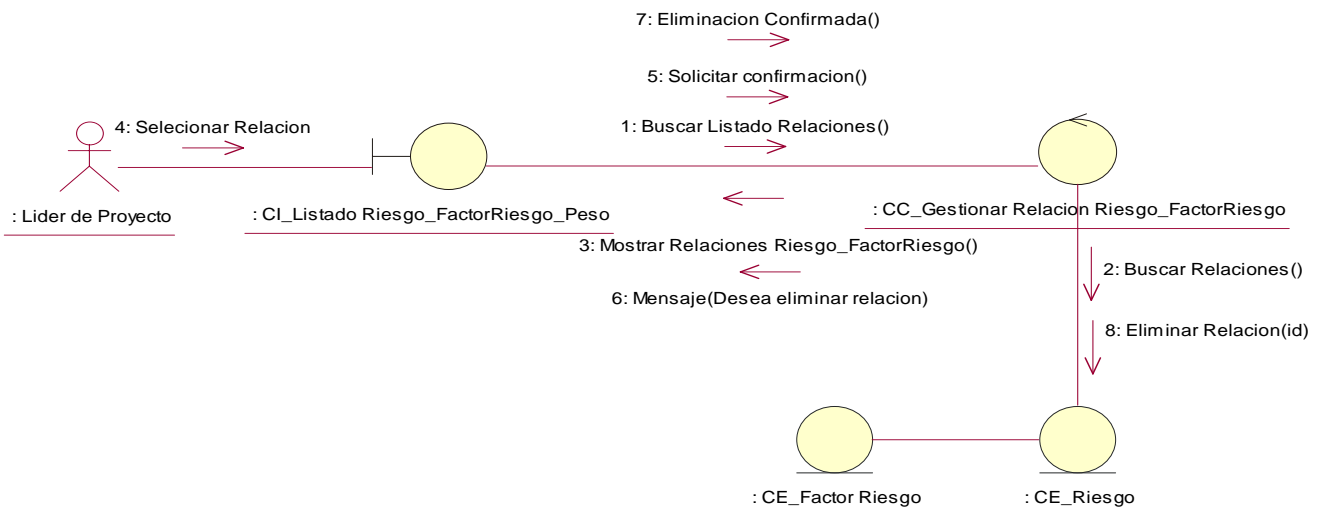
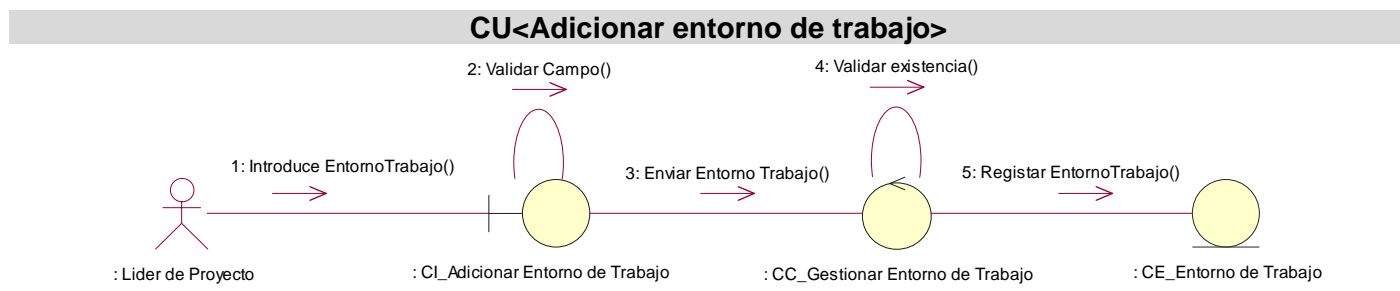
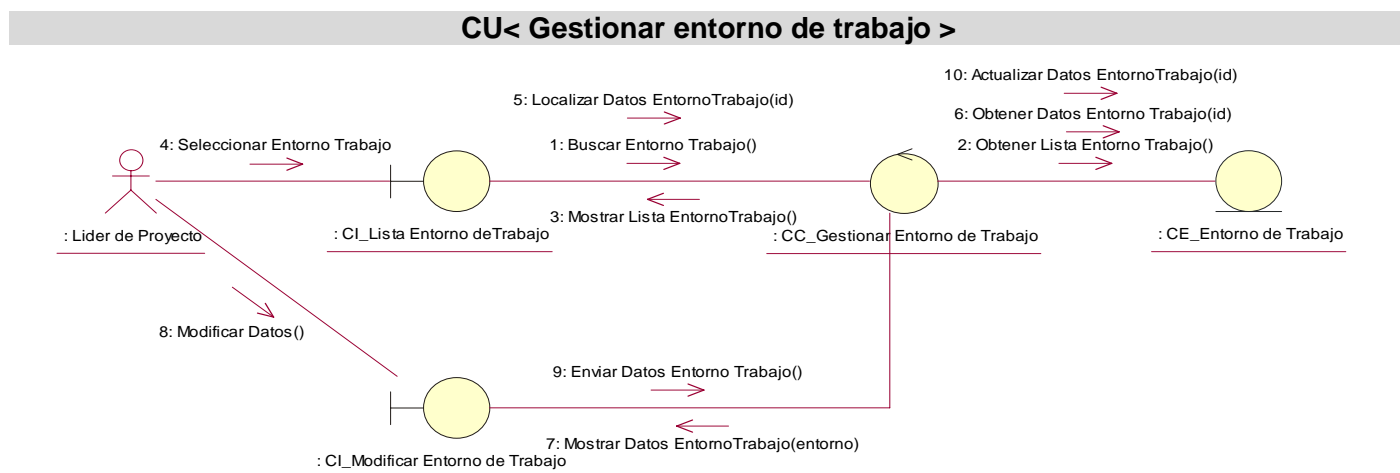


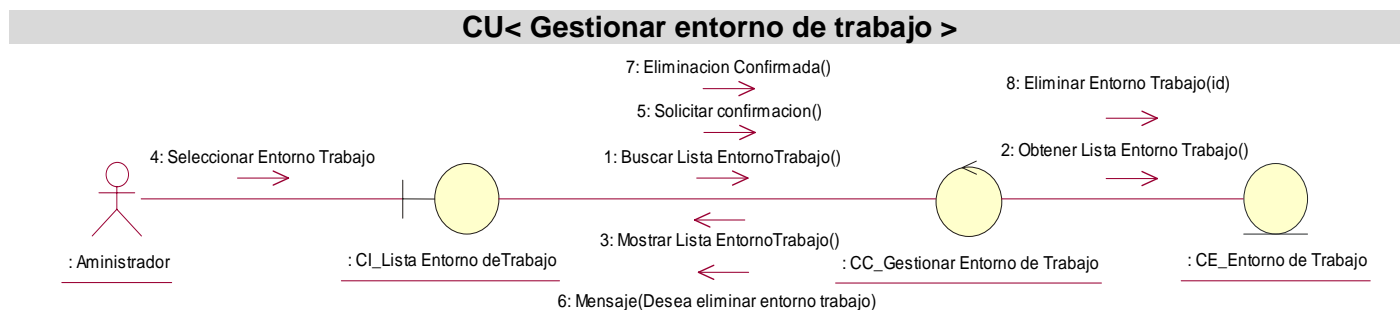
Diagrama de Colaboración <Adicionar entorno de trabajo>



**Diagramas de Colaboración <Gestionar entorno de trabajo>
Escenario <Modificar entorno de trabajo>**

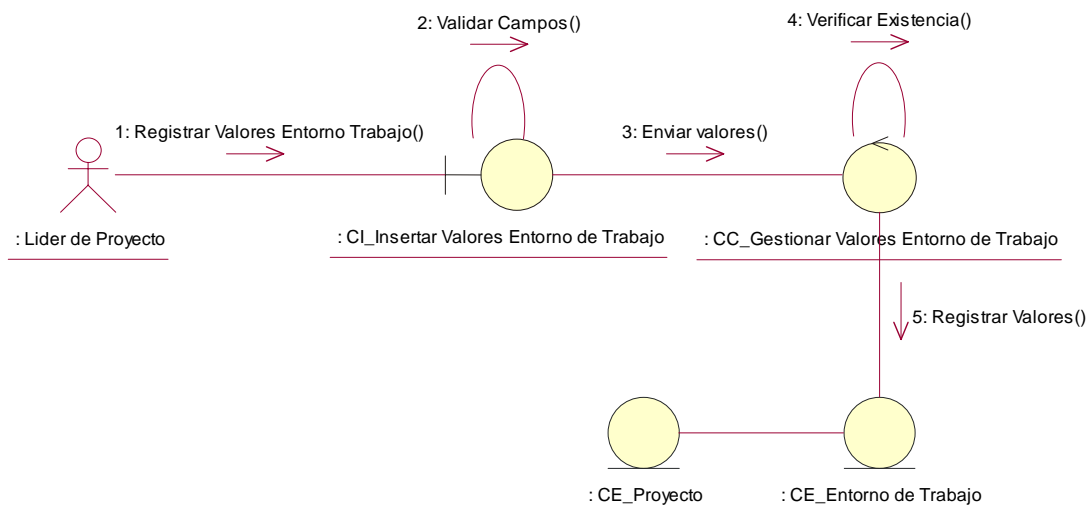


Escenario <Eliminar entorno de trabajo>



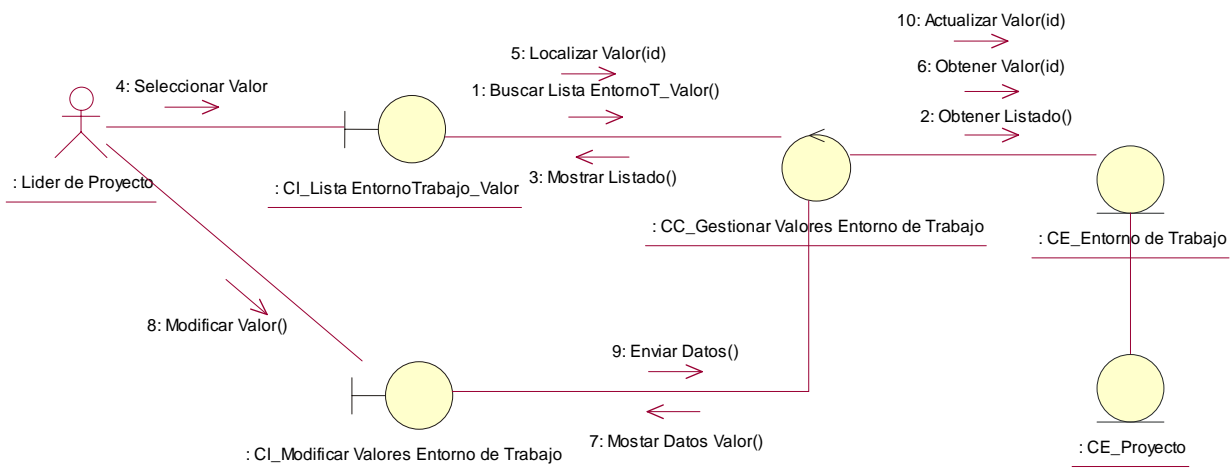
Diagramas de Colaboración <Gestionar valor entorno de trabajo>
 Escenario <Insertar valor entorno de trabajo>

CU< Gestionar valor entorno de trabajo >



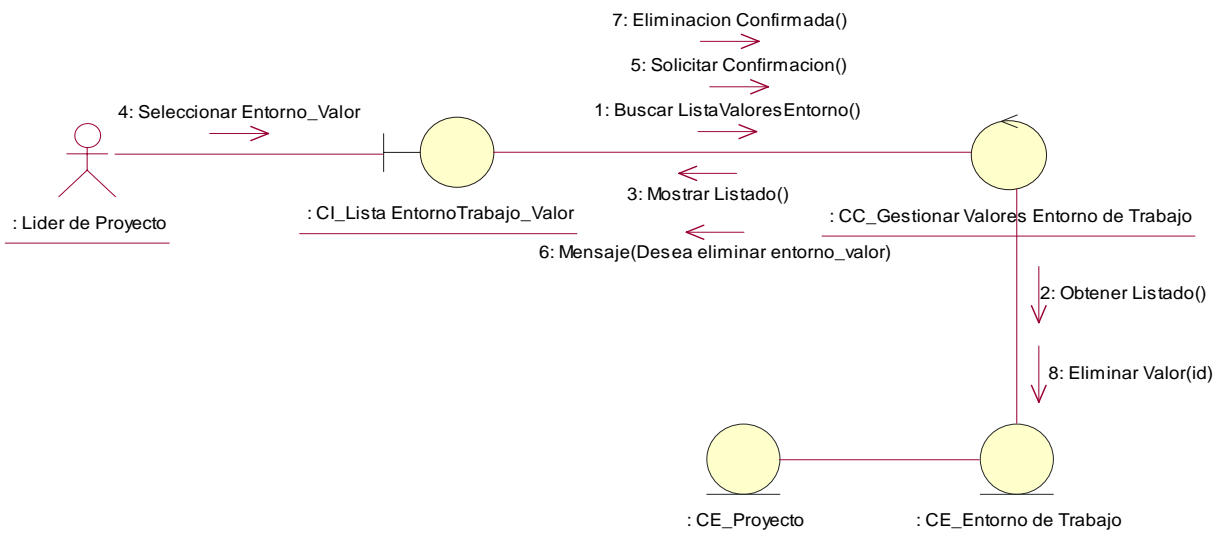
Escenario <Modificar valor entorno de trabajo>

CU< Gestionar valor entorno de trabajo >



Escenario <Eliminar valor entorno de trabajo>

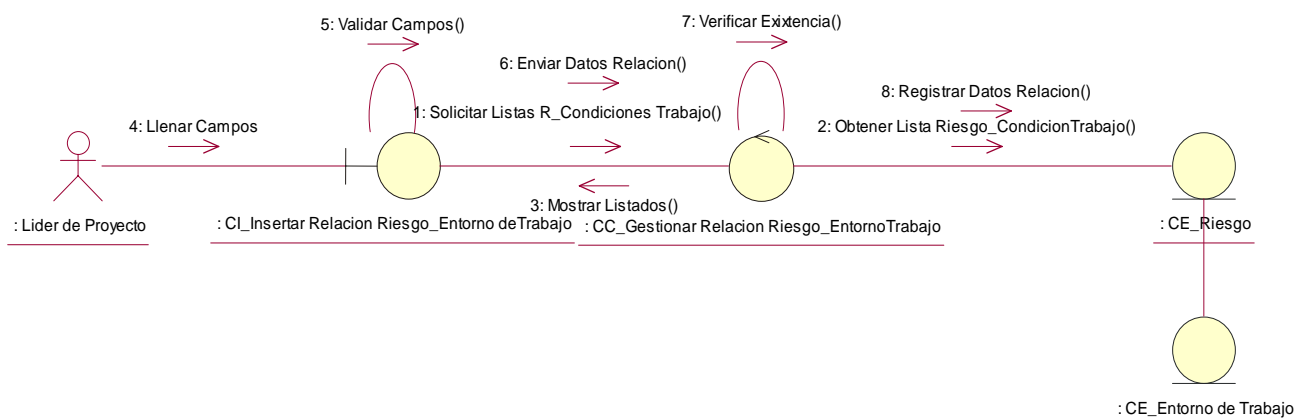
CU< Gestionar valor entorno de trabajo >



Diagramas de Colaboración <Gestionar relación riesgo_entorno>

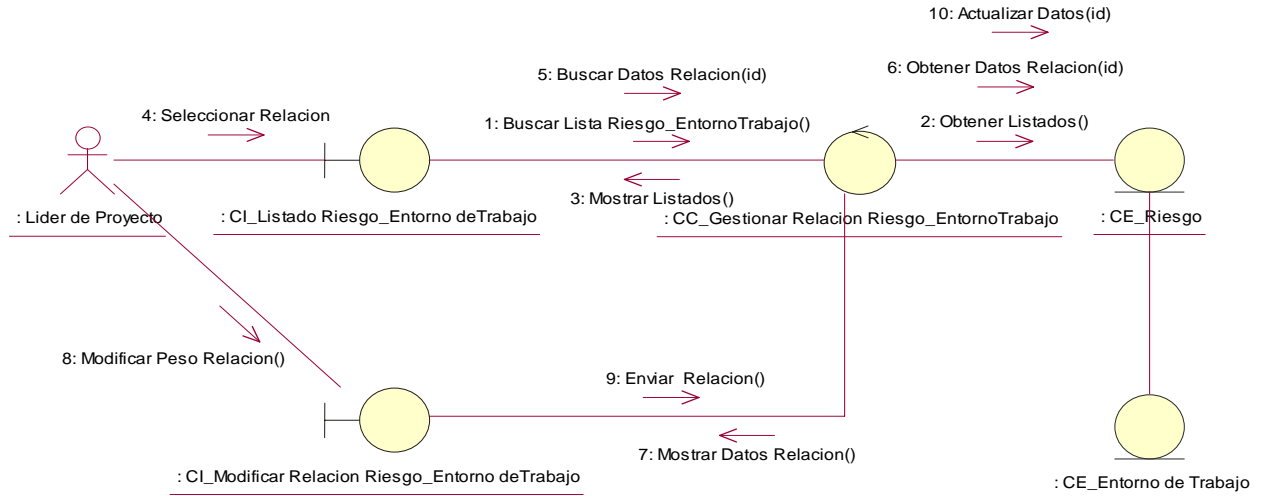
Escenario <Insertar relación riesgo_entorno>

CU< Gestionar relación riesgo_entorno >



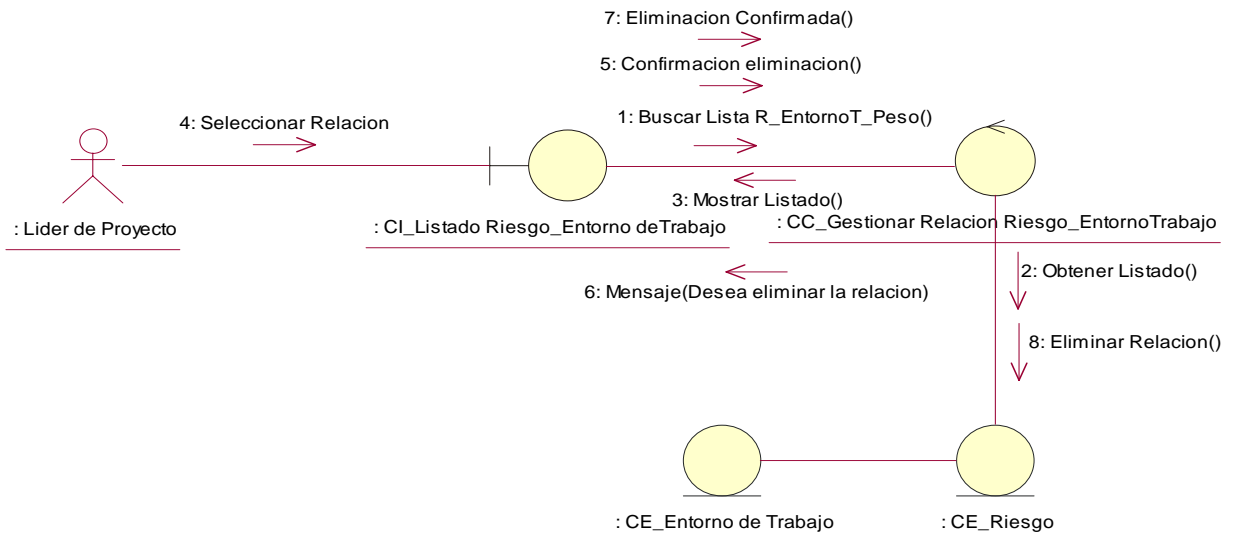
Escenario <Modificar relación riesgo_entorno>

CU< Gestionar relación riesgo_entorno>



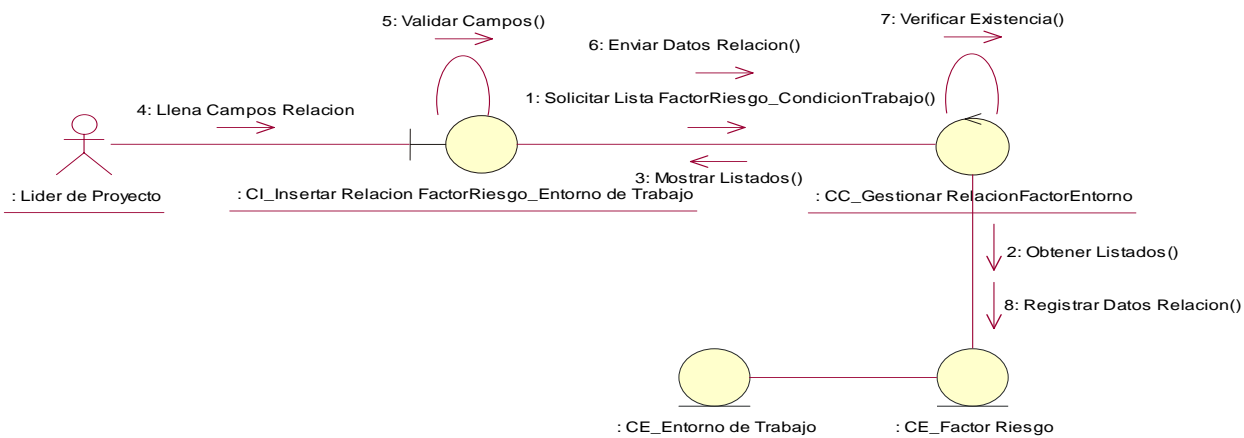
Escenario <Eliminar relación riesgo_entorno>

CU< Gestionar relación riesgo_entorno>



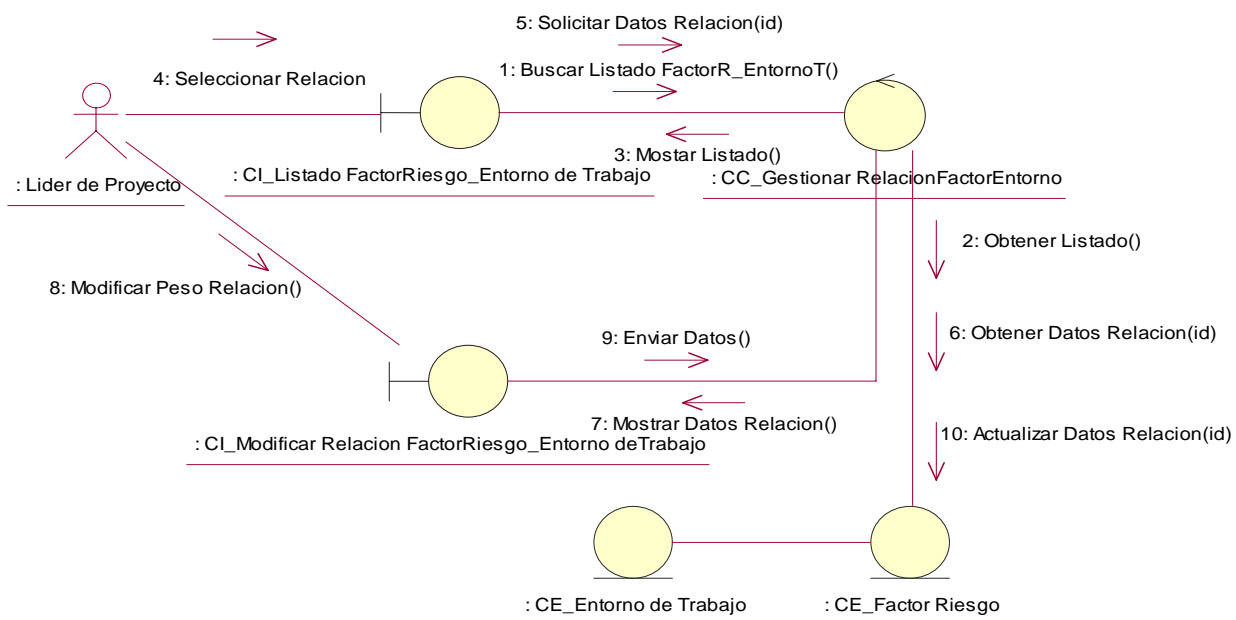
Diagramas de Colaboración <Gestionar relación factor_entorno>
 Escenario <Insertar relación factor_entorno>

CU< Gestionar relación factor_entorno>

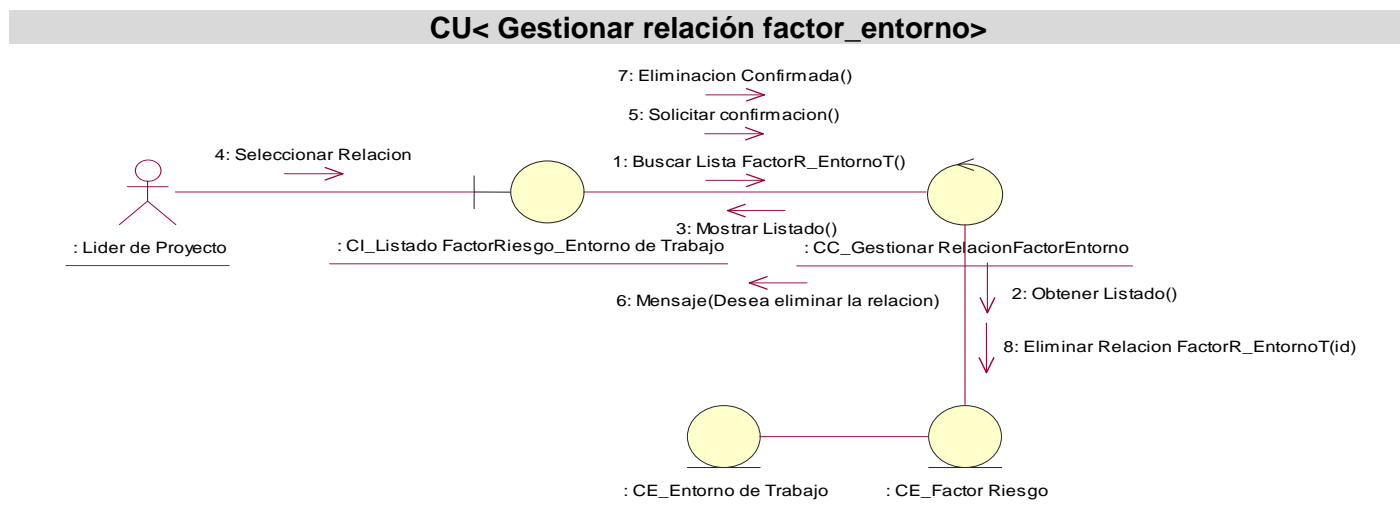


Escenario <Modificar relación factor_entorno>

CU< Gestionar relación factor_entorno>

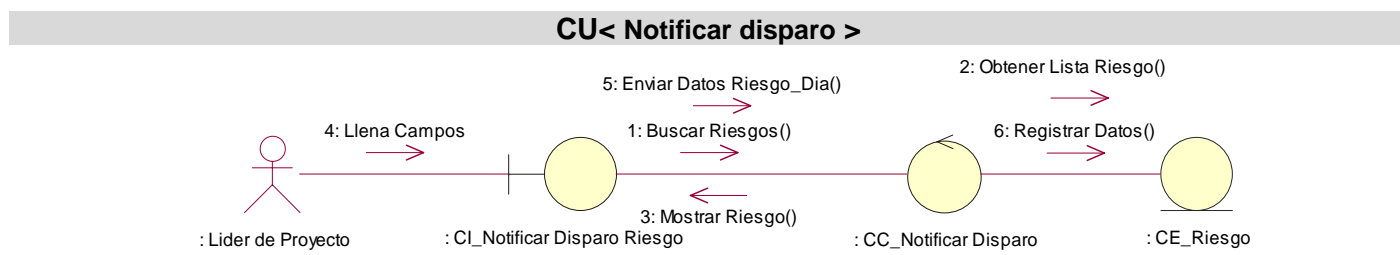


Escenario <Eliminar relación factor_entorno>

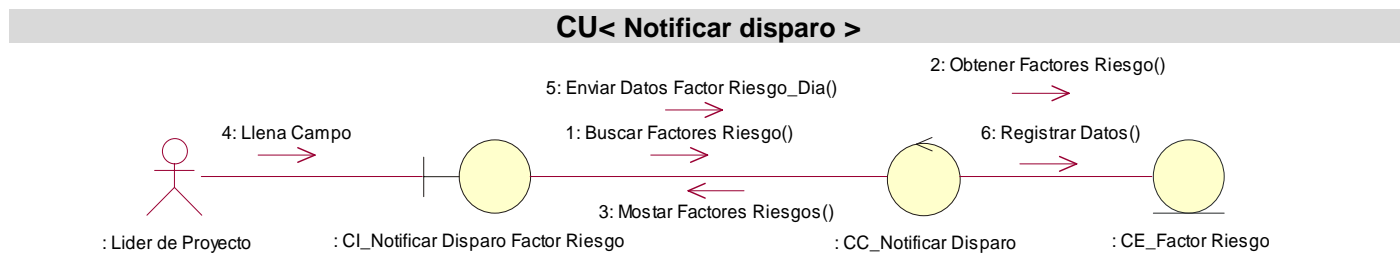


Diagramas de Colaboración <Notificar disparo>

Escenario <Notificar disparo riesgo>

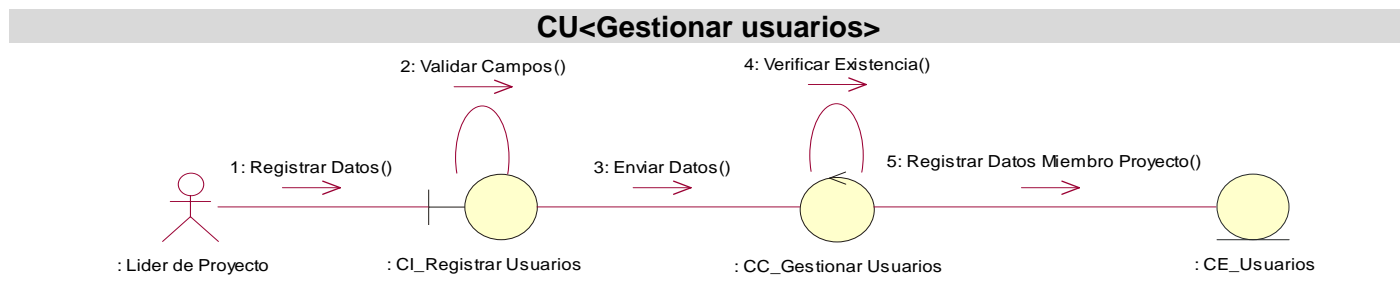


Escenario <Notificar disparo factor riesgo>

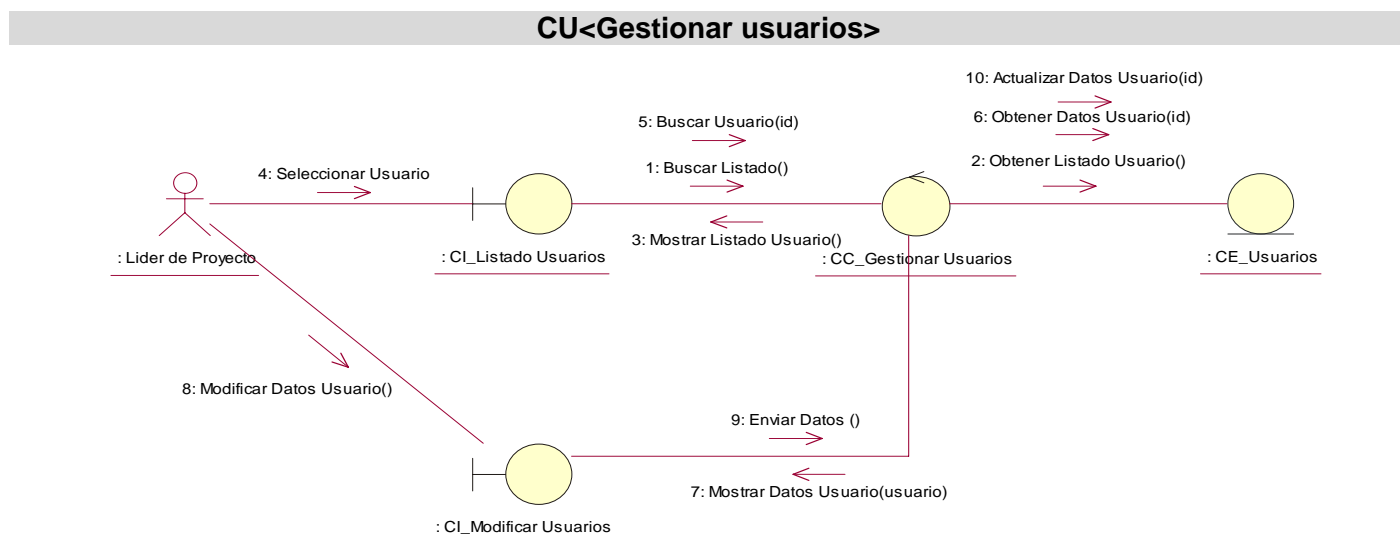


Diagramas de Colaboración <Gestionar usuarios>

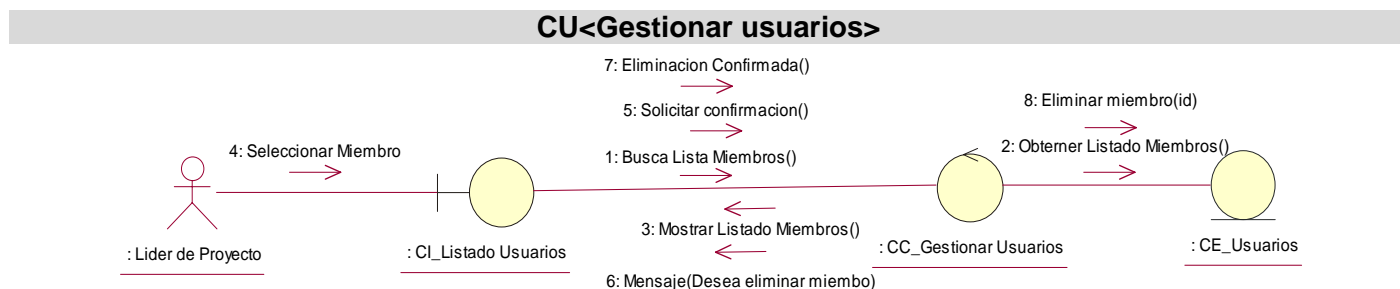
Escenario <Insertar usuarios>



Escenario <Modificar usuarios>



Escenario <Eliminar usuarios>



Escenario <Asignar permisos>

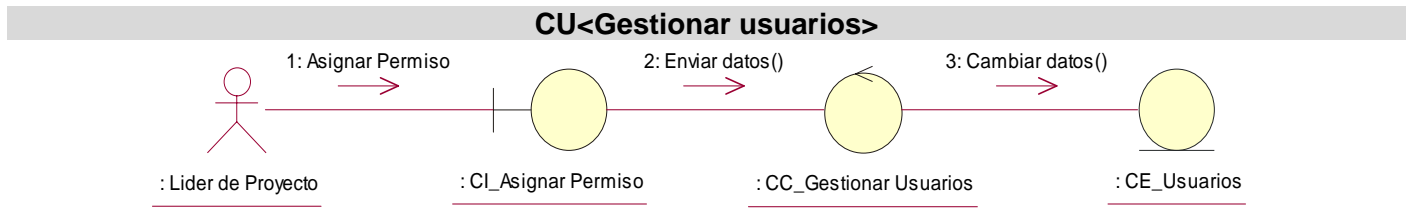


Diagrama de Colaboración <Autenticar usuario>

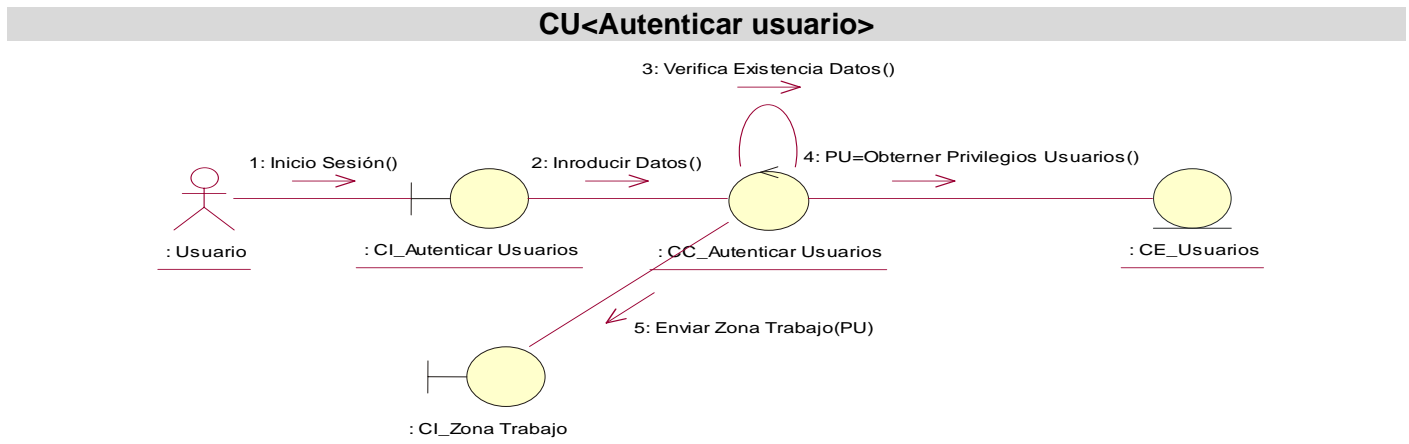


Diagrama de Colaboración <Generar reportes>

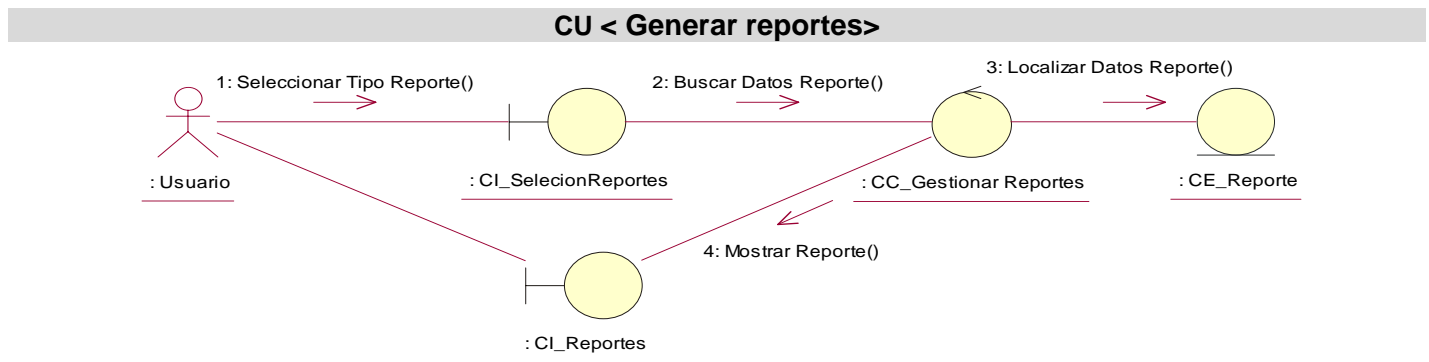
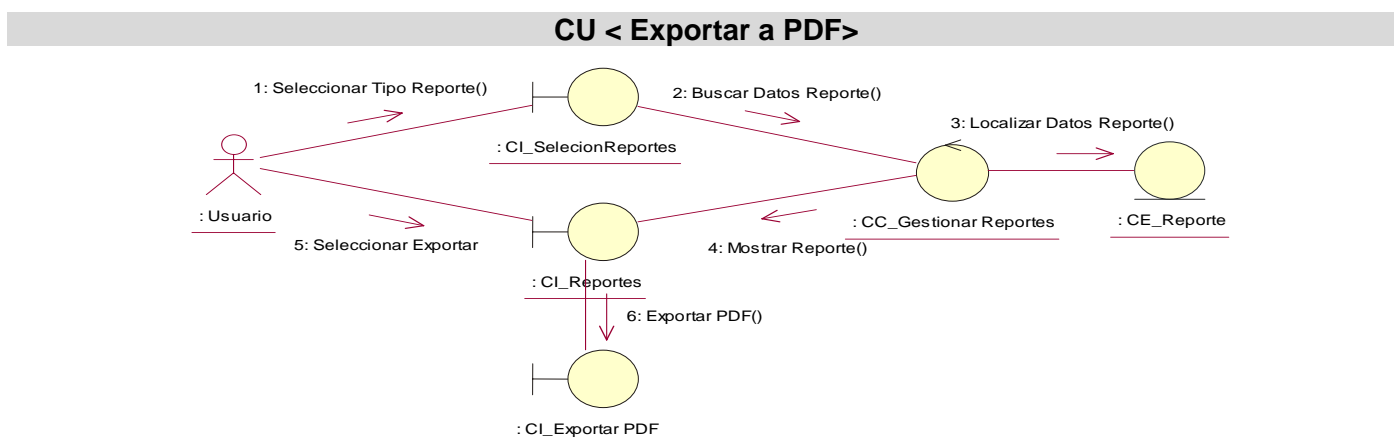


Diagrama de Colaboración <Exportar a PDF>



Anexo 5: Descripción de Estereotipos WEB




Estereotipos	Descripción
 <p>sp_<Nombre Clase Servidora></p>	<p><<Server Page>>: representa la clase que tiene código que se ejecuta en el servidor, la cual se encarga de construir (build) o generar el resultado HTML y/o realizar peticiones a la capa inferior.</p>
 <p>cl_<Nombre Clase Cliente ></p>	<p><<Client Page>>: es una página Web con formato HTML/XHTML. Son interpretadas por el navegador. Sus atributos son las variables declaradas dentro del script que son accesibles para cualquier función dentro de la página. Cada página cliente es construida por una sola página de servidor.</p>
 <p>fr_<Nombre Formulario ></p>	<p><<Form>>: es una colección de elementos de entrada que están contenidos en la página cliente. Sus atributos son los elementos de entrada del formulario (input boxes, text areas, radio buttons, check boxes, hidden fields, entre otros). No tienen operaciones, el método para el paso de los parámetros es \$_POST y se comunican con las páginas servidores mediante submit.</p>

Diagrama de clases del diseño del CU Adicionar factor de riesgo.

CU<Adicionar factor de riesgo>

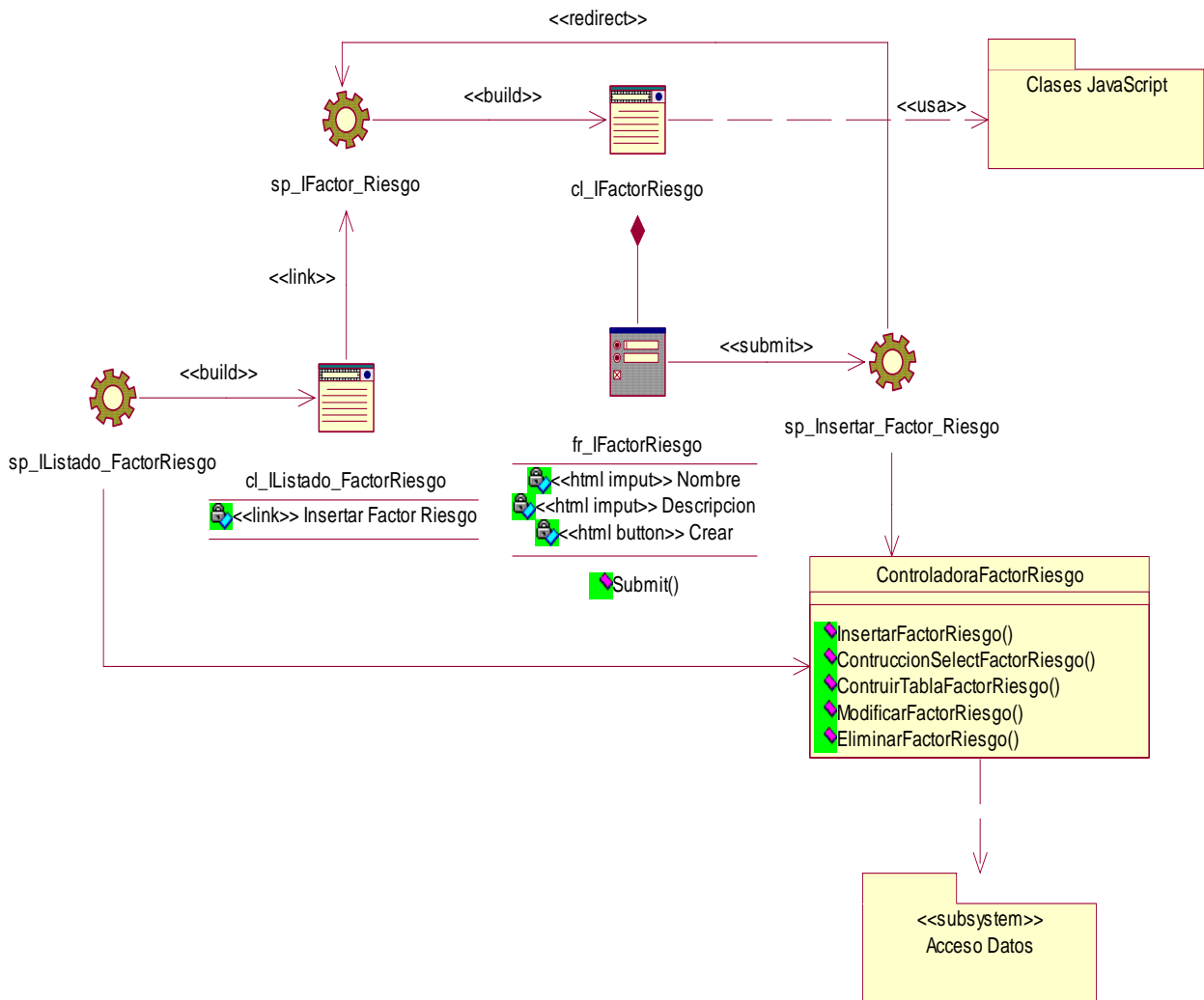


Diagrama de clases del diseño del CU Gestionar factor de riesgo.

CU<Gestionar factor de riesgo>

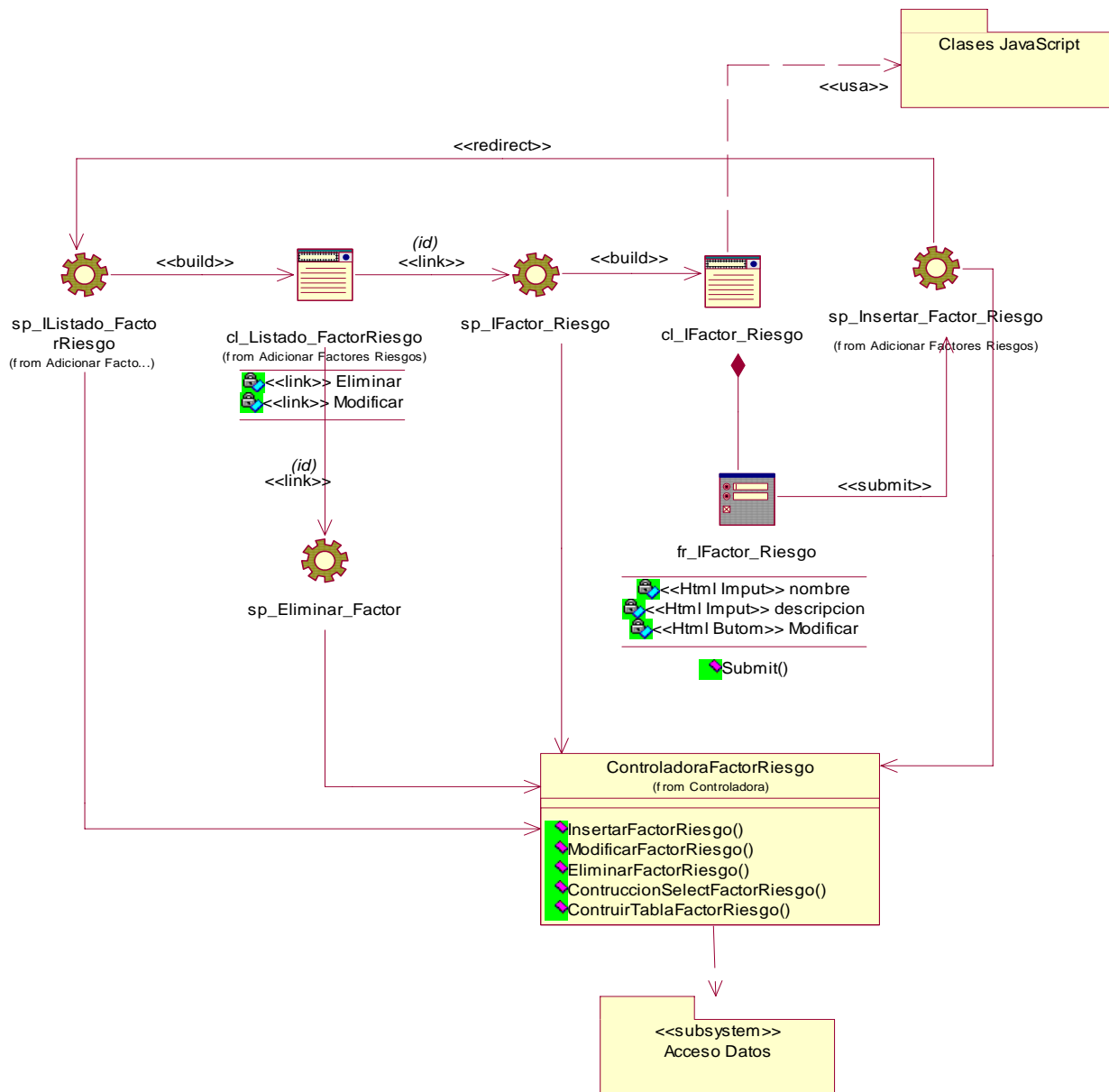


Diagrama de clases del diseño del CU Gestionar valor entorno de trabajo.

CU<Gestionar valor entorno de trabajo>

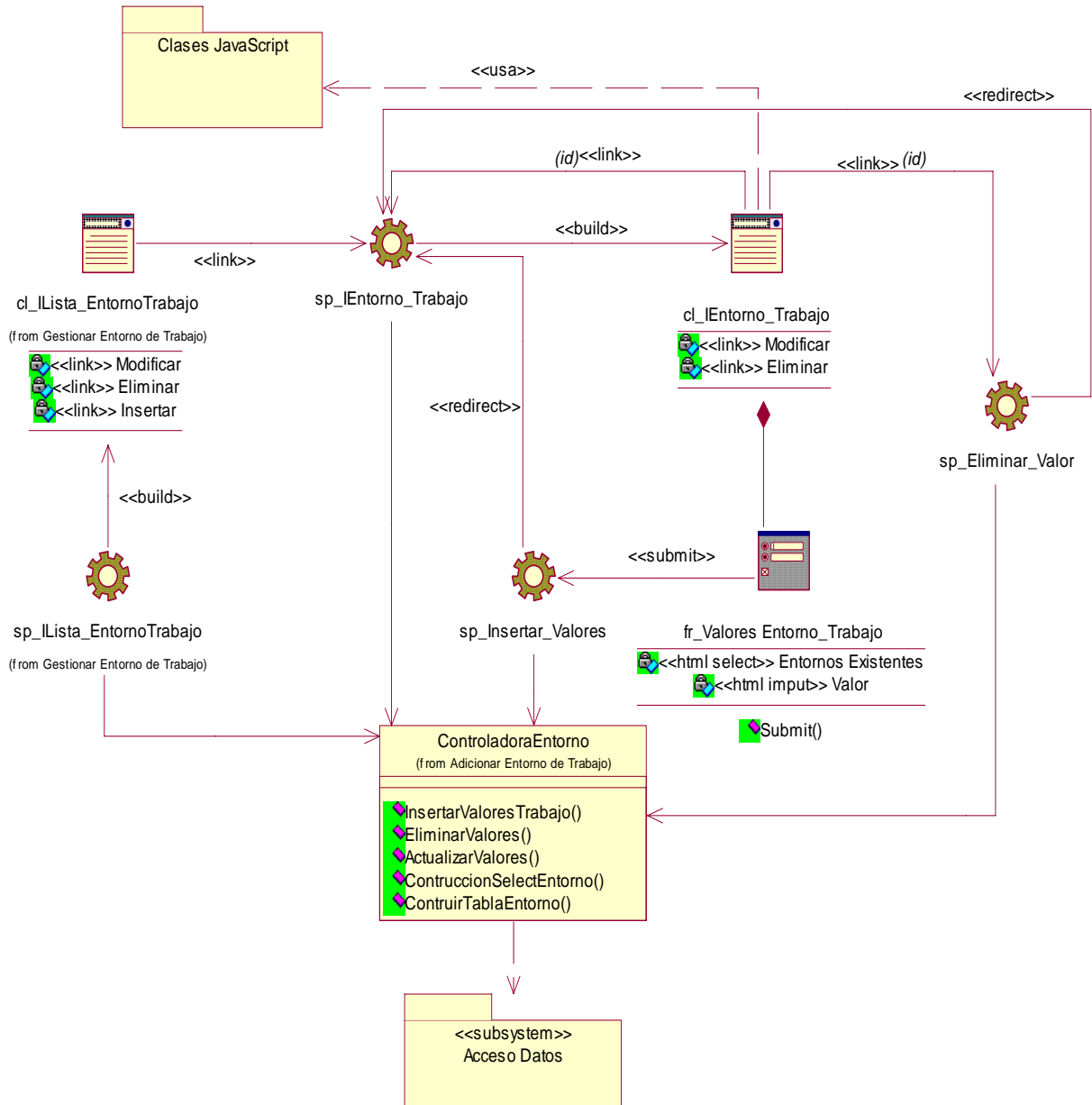


Diagrama de clases del diseño del CU Gestionar relación riesgo_factor.

CU<Gestionar relación riesgo_factor>

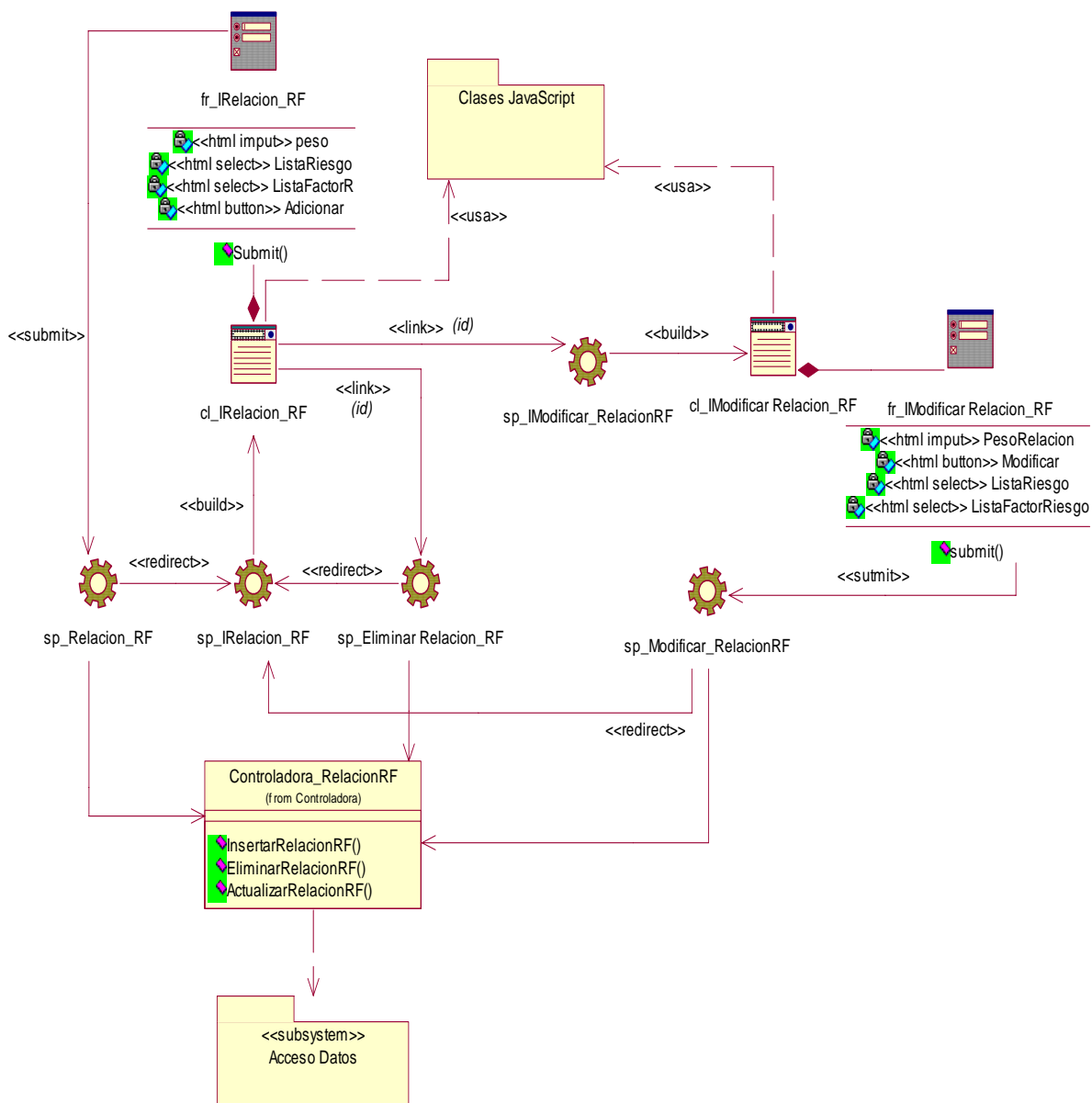


Diagrama de clases del diseño del CU Gestionar relación riesgo_entorno.

CU<Gestionar relación riesgo_entorno>

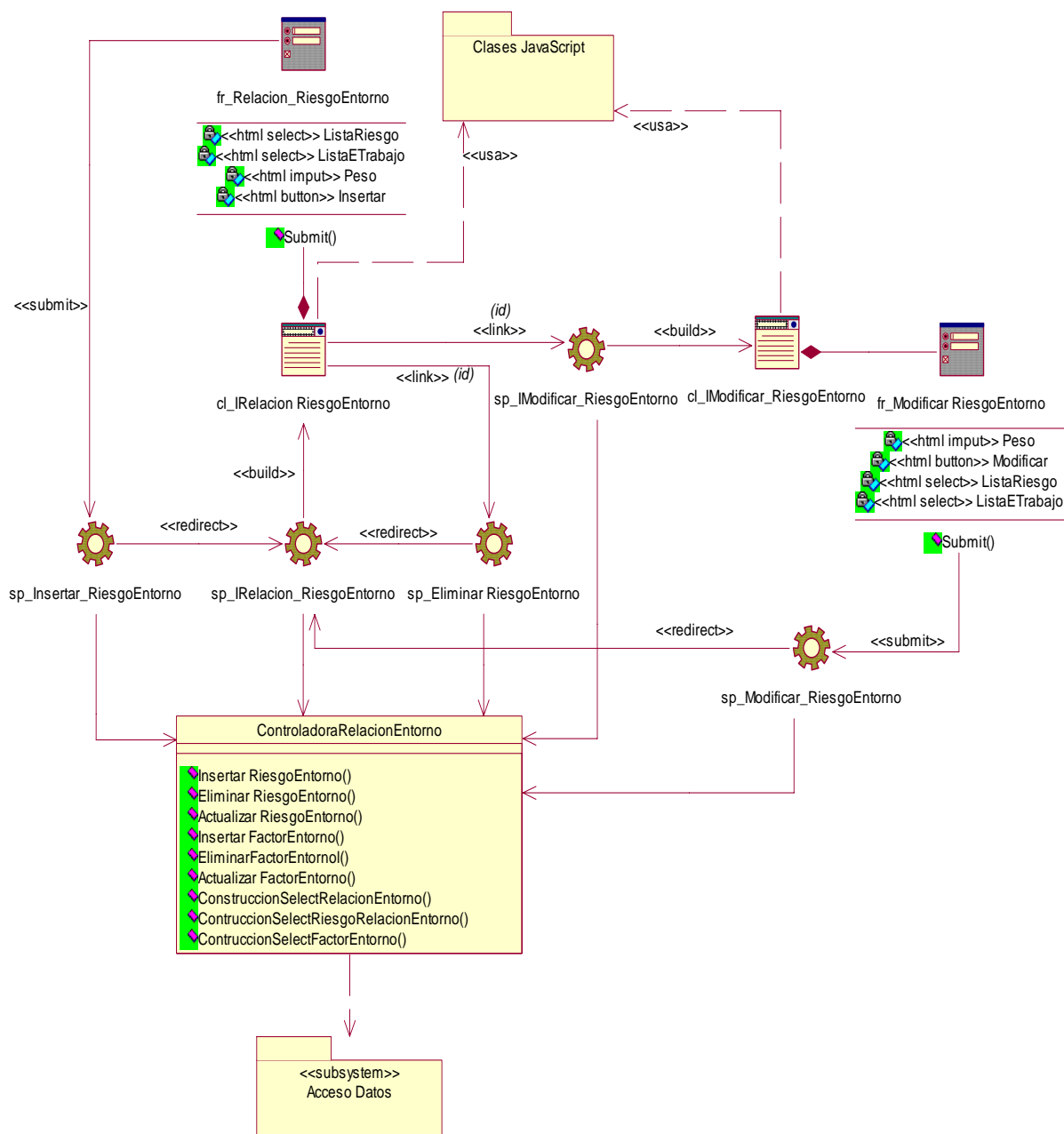


Diagrama de clases del diseño del CU Gestionar relación factor_entorno.

CU<Gestionar relación factor_entorno>

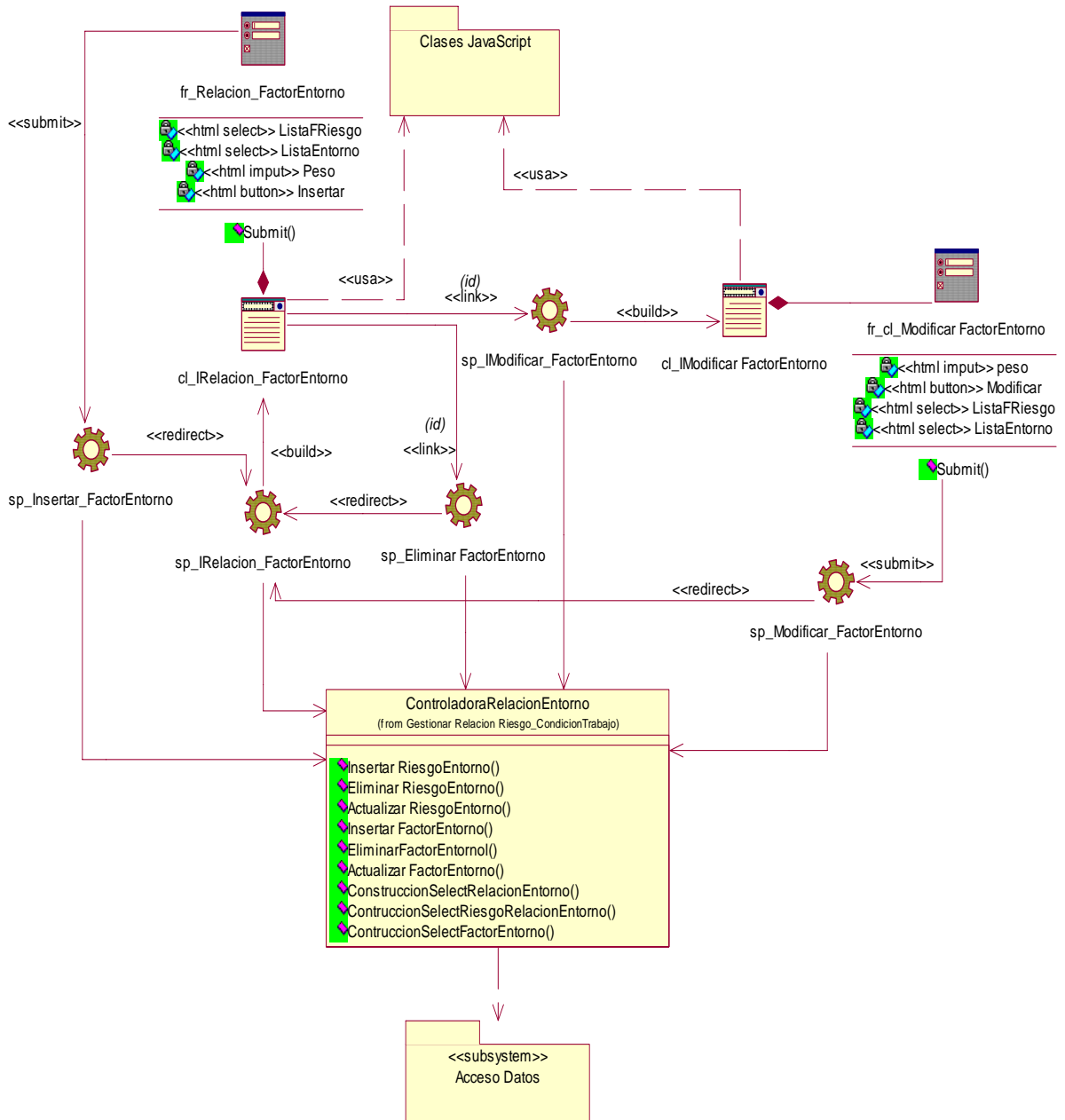


Diagrama de clases del diseño del CU Notificar disparo.

CU<Notificar disparo>

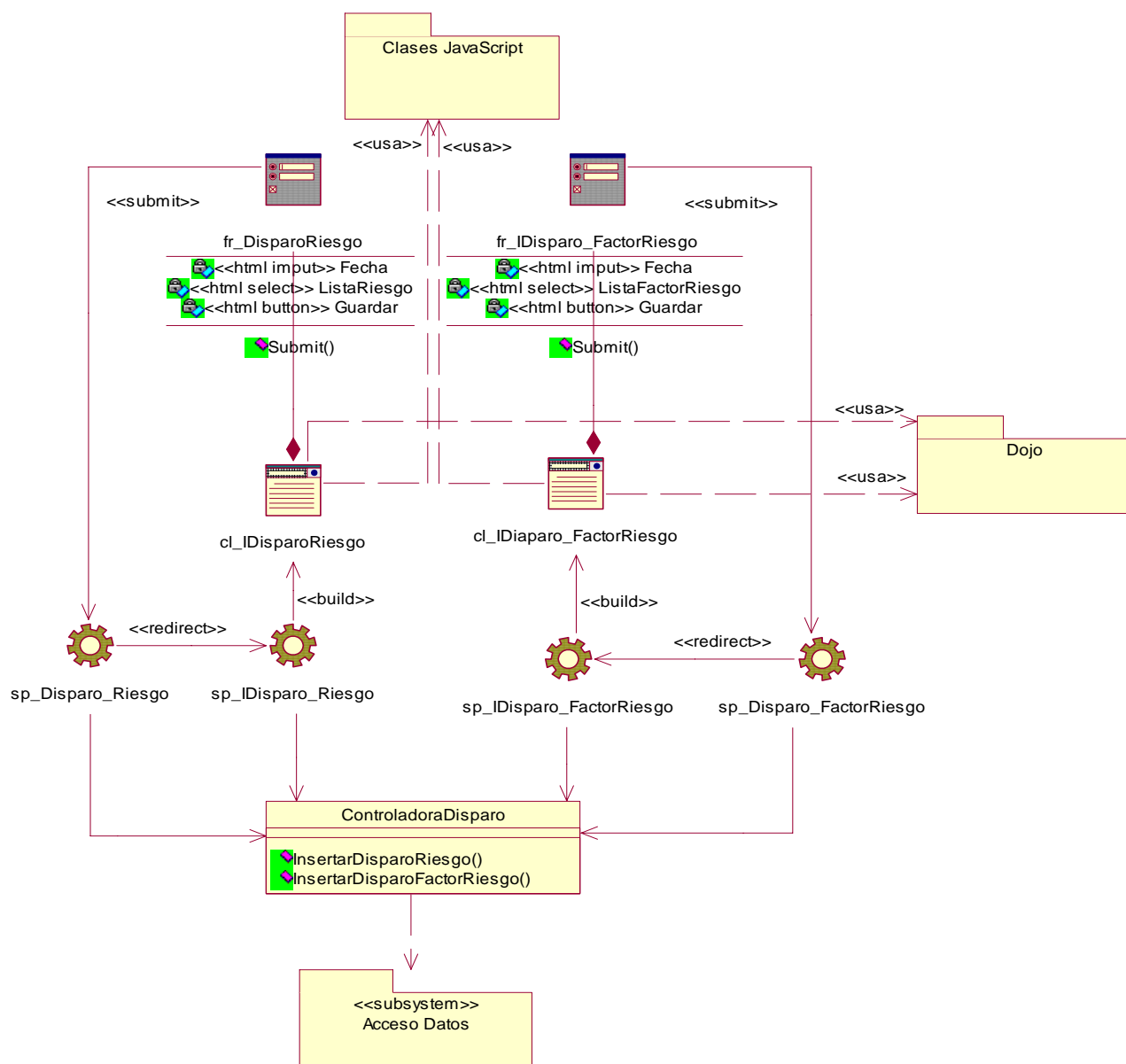


Diagrama de clases del diseño del CU Generar reportes.

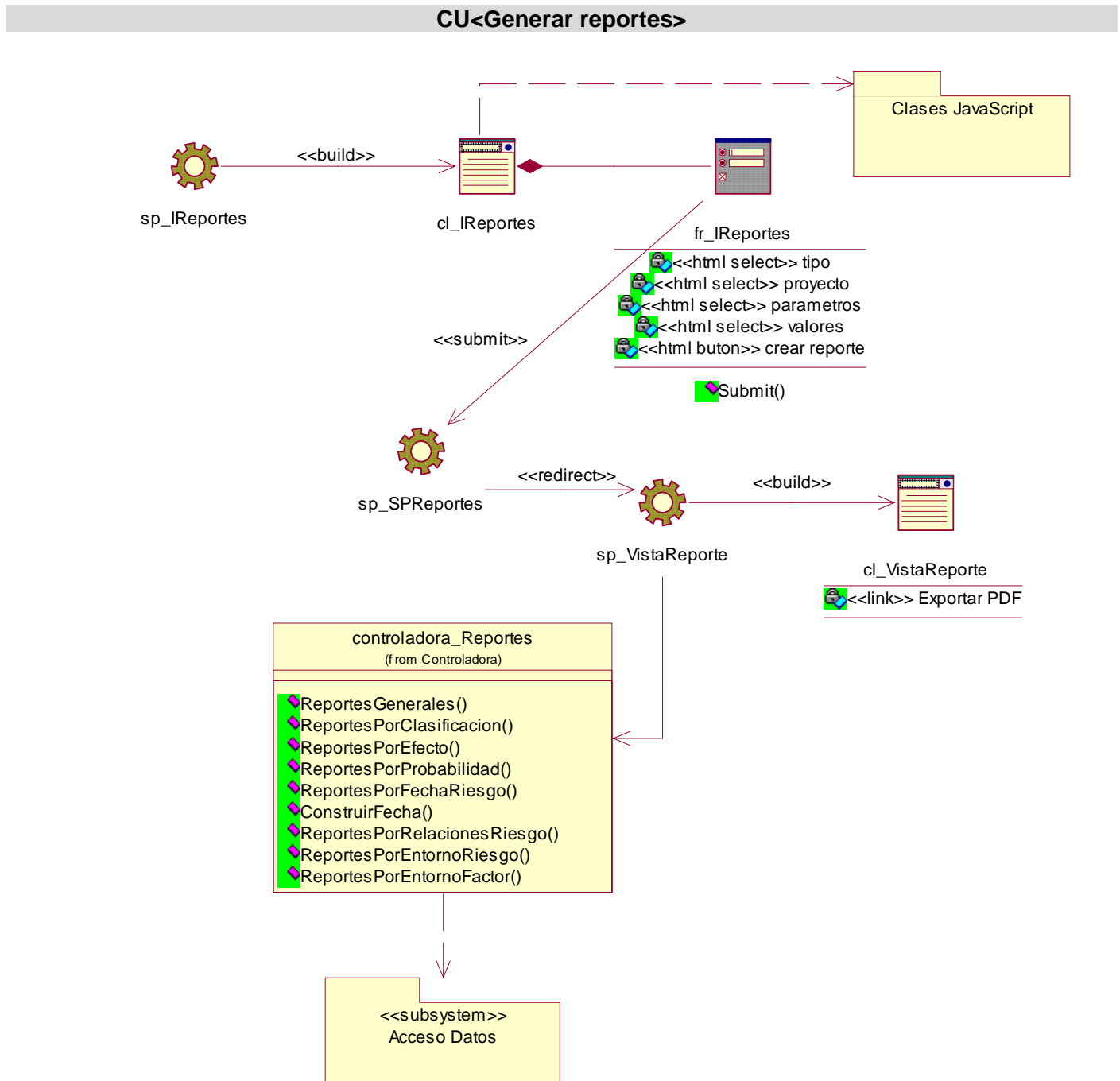


Diagrama de clases del diseño del CU Exportar a PDF.

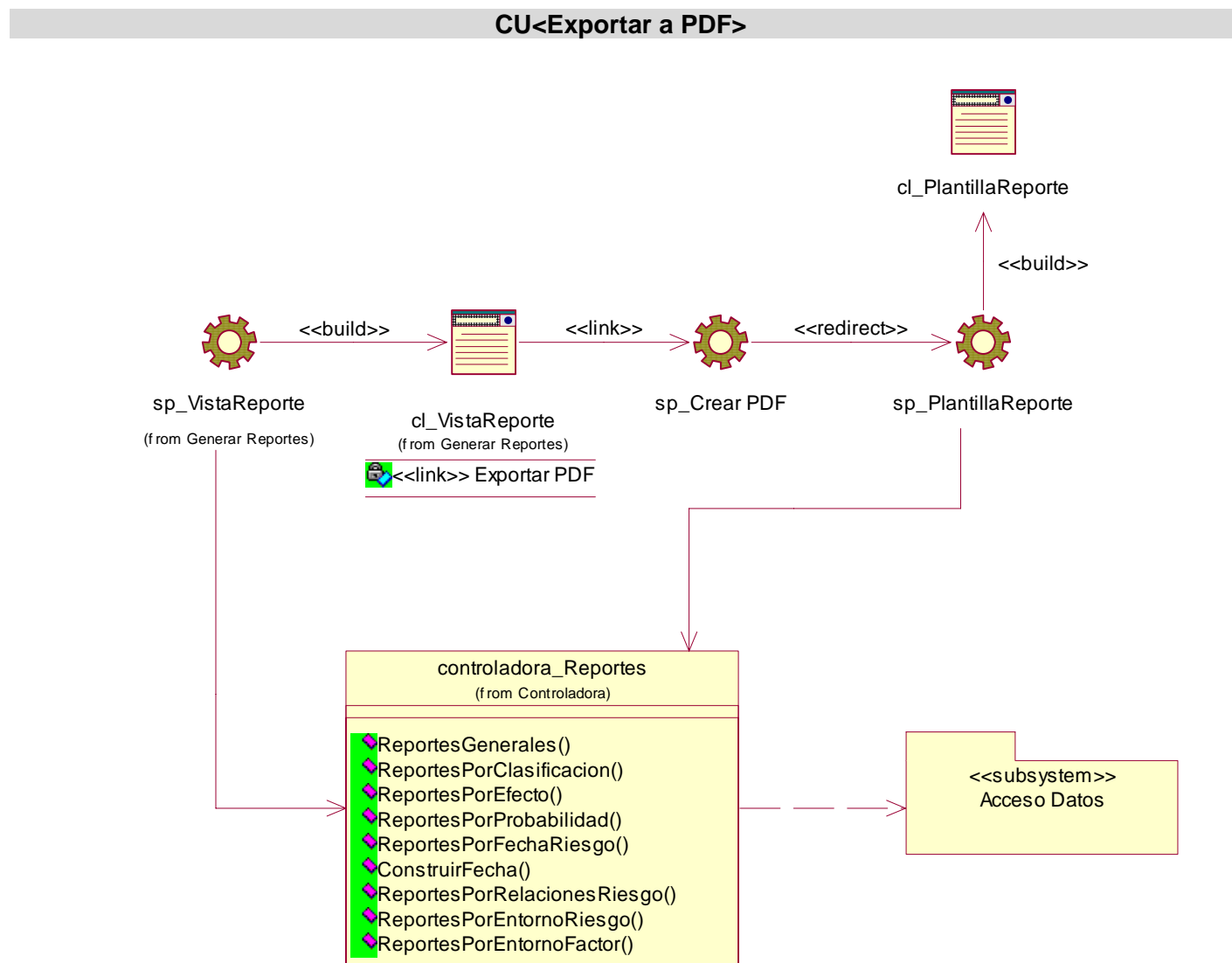


Diagrama de clases del diseño del CU Autenticar usuario.

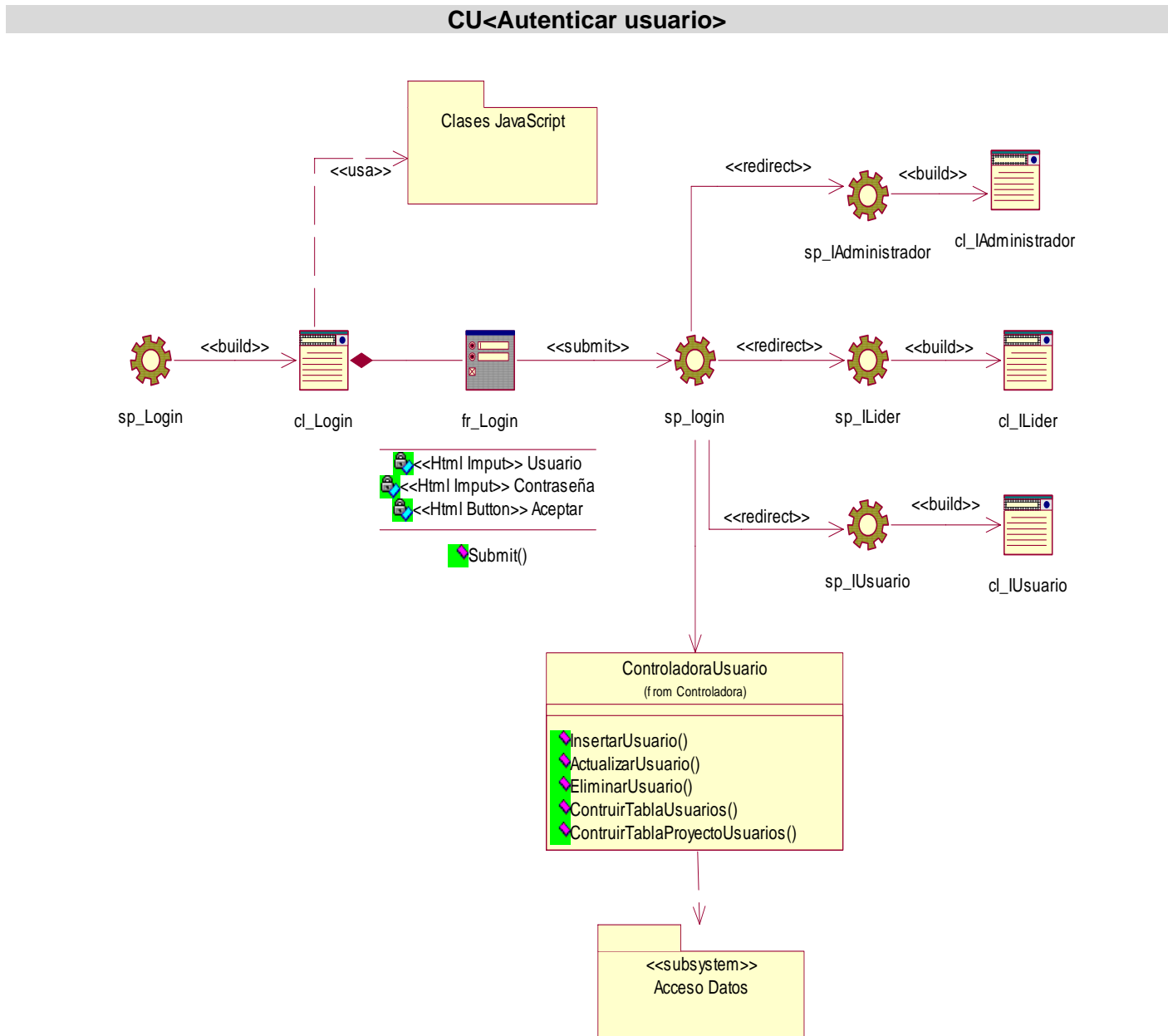
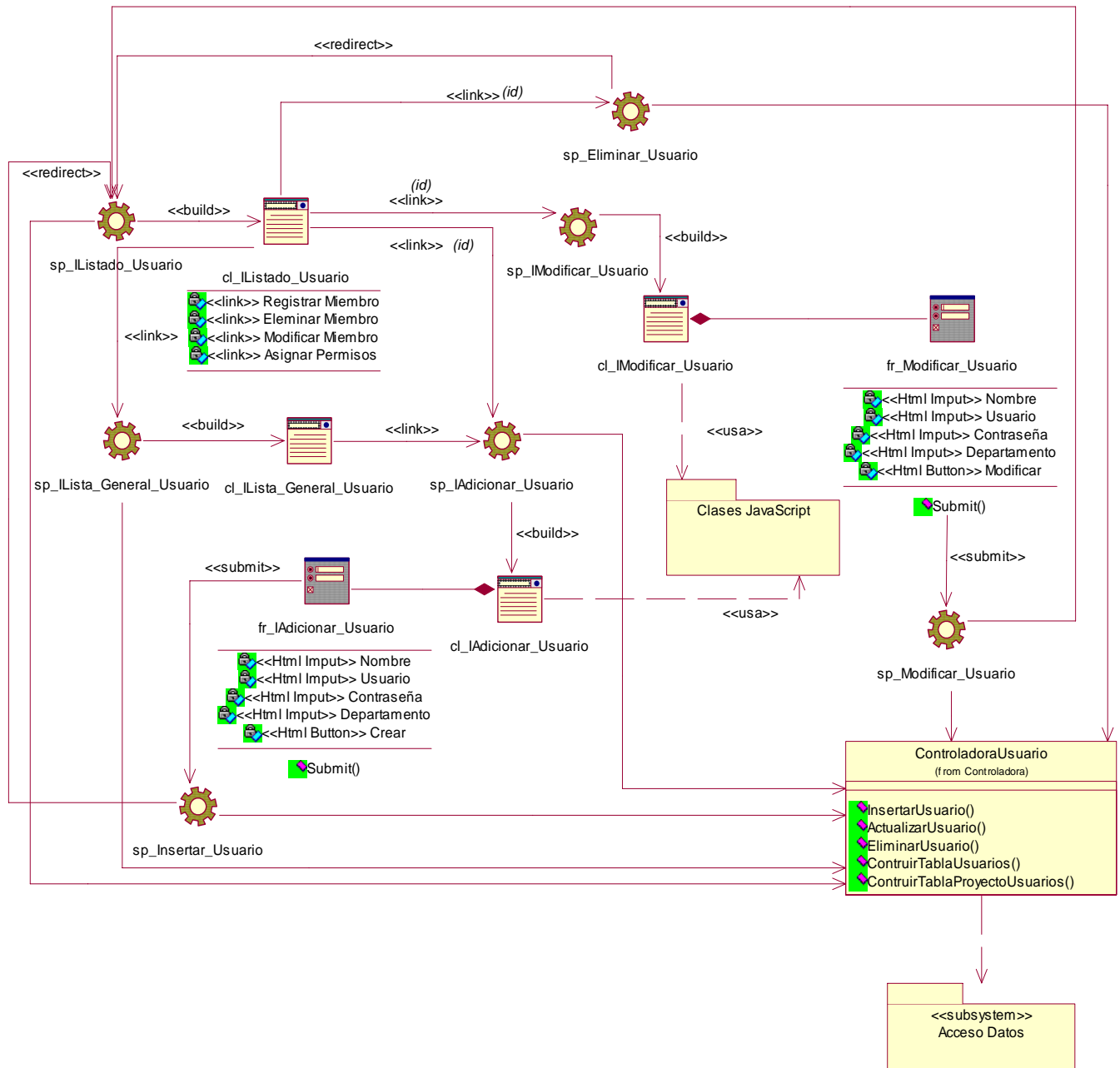


Diagrama de clases del diseño del CU Gestionar usuarios.

CU<Gestionar usuarios>



Anexo 7: Diagrama de clases del Paquete Acceso a Datos.

Diagrama de clases del subsistema Acceso a Datos(Primera parte)

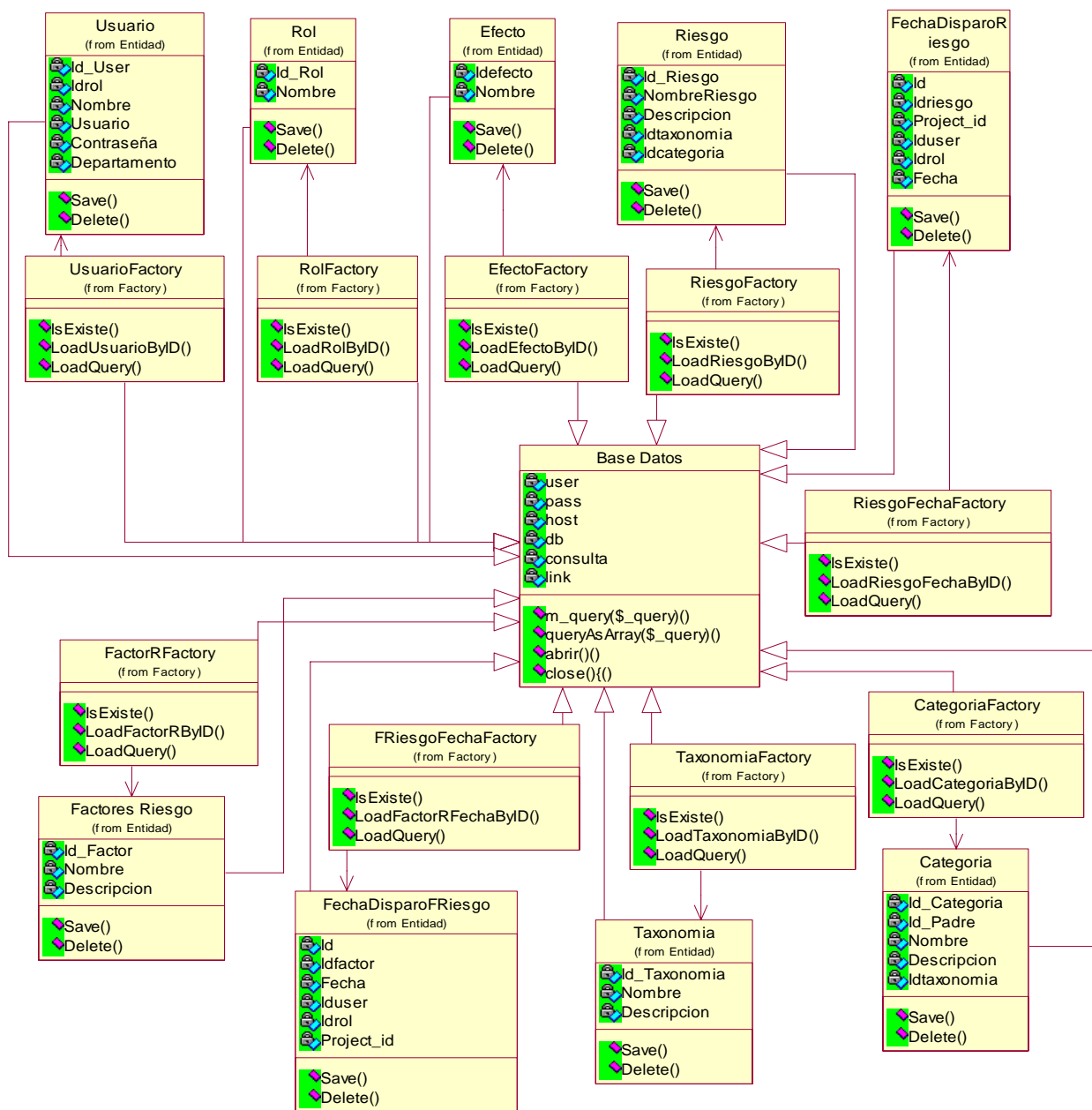
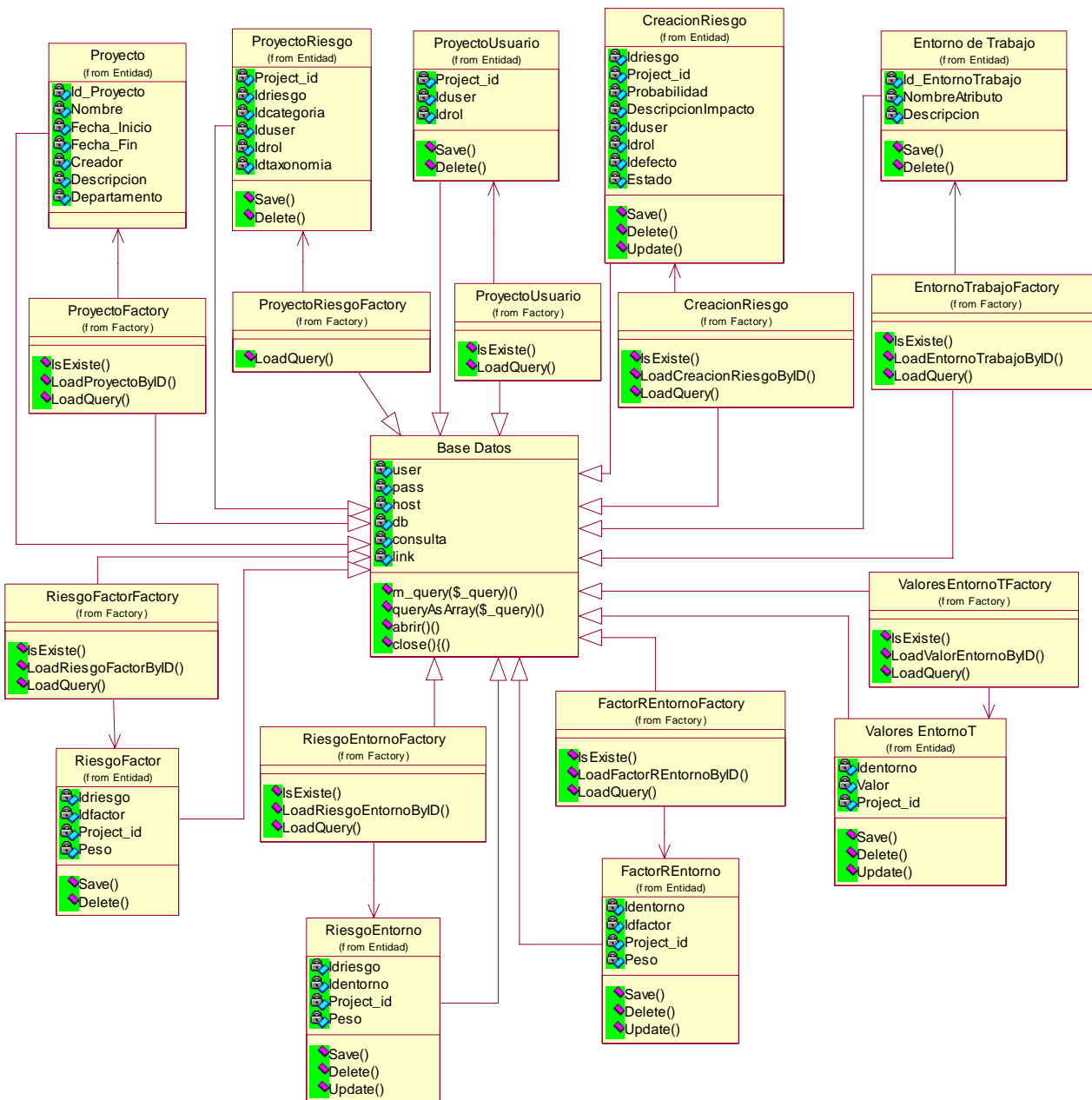


Diagrama de clases del Subsistema Acceso a Datos(Segunda Parte)



Anexo 8: Descripción de las tablas de la Base de Datos

Tabla: Riesgo

Nombre: riesgo		
Descripción: En esta tabla se almacenan los datos relacionados con los riesgos.		
Atributo	Tipo	Descripción
idriesgo	int(11)	Identificador de los riesgos existentes en el sistema.
nombre	varchar(20)	Nombre de los riesgos.
descripción	text	Breve descripción de cada riesgo.
idtaxonomía	int(11)	Identificador de la taxonomía.
idcategoría	int(11)	Identificador de las categorías existentes en el sistema.

Tabla: Factor

Nombre: factor		
Descripción: En esta tabla se almacenan los datos relacionados con los factores de riesgos.		
Atributo	Tipo	Descripción
idfactor	int(11)	Identificador de los factores de riesgos existentes en el sistema.
nombre	varchar(20)	Nombre de los factores de riesgos.
descripción	text	Breve descripción de cada factor de riesgo.

Tabla: Entorno_Trabajo

Nombre: entorno_trabajo		
Descripción: En esta tabla se almacenan los entornos de trabajo existentes en el sistema.		
Atributo	Tipo	Descripción
Idcondiciones_trabajo	int(11)	Identificador de los entornos de trabajo existentes en el sistema.
nombre	varchar (50)	Nombre del entorno de trabajo.
descripción	text	Breve descripción del entorno de trabajo.

Tabla: Project

Nombre: project		
Descripción: En esta tabla se almacena todo lo relacionado con los proyectos.		
Atributo	Tipo	Descripción
project_id	int(11)	Identificador de los proyectos que existen en la Base de Datos.
project_company	int(11)	Compañía a la que pertenece el proyecto.
project_company_internal	int(11)	Compañías internas del proyecto.
project_department	int(11)	Número del departamento.
project_name	varchar(255)	Nombre del proyecto.
project_short_name	varchar(10)	Sobrenombre del proyecto.
project_owner	int(11)	Dueño o líder de proyecto.
project_url	varchar(255)	Dirección URL del proyecto.
project_demo_url	varchar(255)	Dirección URL alternativa.
project_end_date	datetime	Fecha de culminación del proyecto.
project_start_date	datetime	Fecha de inicio del proyecto.
project_actual_end_date	datetime	Fecha actual en que está el proyecto.
project_status	int(11)	Estado en el cual se encuentra el proyecto.
project_percent_complete	tinyint(4)	Porcentaje de terminación del proyecto.
project_color_idenfier	varchar(6)	Color que identifica el proyecto.
project_description	text	Breve descripción del proyecto.
project_target_budget	decimal(10,2)	Presupuesto para cumplir la meta del proyecto.
project_actual_budget	decimal(10,2)	Presupuesto actual del proyecto.
project_creator	int(11)	Persona que crea el proyecto.
project_private	tinyint(4)	Número que te indica si es o no privado el proyecto.
project_departments	char(100)	Nombre del departamento.
project_priority	tinyint(4)	Prioridad del proyecto.
project_type	smallint(6)	Tipo de proyecto.

Tabla: Categoría

Nombre: categoría		
Descripción: En esta tabla se almacenan las categorías en las cuales se van a clasificar los riesgos del proyecto.		
Atributo	Tipo	Descripción
idtaxonomía	int(11)	Identificador de la taxonomía.
idcategoría	int(11)	Identificador de las categorías existentes en el sistema.
idcategoríapadre	int(11)	Identificador que te permite saber de quién es hija la categoría.
nombre	varchar(20)	Nombre de la categoría.
descripción	text	Breve descripción de la categoría.

Tabla: Taxonomía

Nombre: taxonomía		
Descripción: En esta tabla se almacenan los elementos que conforman la taxonomía.		
Atributo	Tipo	Descripción
idtaxonomía	int(11)	Identificador de la taxonomía.
nombre	varchar(20)	Nombre de los elementos.
descripción	text	Breve descripción de los elementos.

Tabla: Nomenclador_Efecto

Nombre: nomenclador_efecto		
Descripción: En esta tabla se almacenan los efectos que pueden causar los riesgos en los proyectos.		
Atributo	Tipo	Descripción
idefecto	int(11)	Identificador de los efectos que existen en el sistema.
nombre	varchar(20)	Nombre de los efectos existentes.

Tabla: Usuario

Nombre: usuario		
Descripción: En esta tabla se almacenan los datos relacionados con los usuarios que pueden acceder al sistema.		
Atributo	Tipo	Descripción
iduser	int(11)	Identificador de los usuarios existentes en el sistema.
nombre	varchar(50)	Nombre de los usuarios.
user	varchar(50)	Usuario correspondiente con las personas que pueden acceder al sistema.
pass	varchar(25)	Contraseña correspondiente a cada usuario.
dpto	varchar(50)	Departamento al que pertenece cada usuario.

Tabla: Rol

Nombre: rol		
Descripción: En esta tabla se almacenan los datos relacionados con los niveles de acceso de los usuarios.		
Atributo	Tipo	Descripción
idrol	int(11)	Identificador de los roles que tienen los usuarios del sistema.
nombre	varchar(50)	Nombre del rol que ocupa cada usuario.

Tabla: Creación_Riesgo

Nombre: creación_riesgo		
Descripción: En esta tabla se almacenan algunas características de los riesgos de acuerdo al proyecto.		
Atributo	Tipo	Descripción
idriesgo	int(11)	Identificador de los riesgos existentes en el sistema.
project_id	int(11)	Identificador de los proyectos existentes en el sistema.
iduser	int(11)	Identificador de los usuarios existentes en el sistema.

Idrol	int(11)	Identificador de los roles existentes en el sistema.
idefecto	int(11)	Identificador de los efectos que causan los riesgos.
estado	varchar(2)	Estado de los riesgos (creado o reutilizado).
descripción_impacto	text	Breve descripción del impacto que provoca el riesgo.

Tabla: Disp_Riesgo

Nombre: disp_riesgo		
Descripción: En esta tabla se almacenan las fechas en que se dispara un riesgo.		
Atributo	Tipo	Descripción
id	int(11)	Identificador de la fecha de disparo de un riesgo.
idriesgo	int(11)	Identificador de los riesgos existentes en el sistema.
project_id	int(11)	Identificador de los proyectos existentes en el sistema.
iduser	int(11)	Identificador de los usuarios existentes en el sistema.
idrol	int(11)	Identificador de los roles que tienen los usuarios del sistema.
fecha	date	Fecha en la cual se dispara un riesgo.

Tabla: Disp_Factor

Nombre: disp_factor		
Descripción: En esta tabla se almacenan las fechas en que se dispara un factor de riesgo.		
Atributo	Tipo	Descripción
id	int(11)	Identificador de la fecha de disparo de un factor de riesgo.
idfactor	int(11)	Identificador de los factores de riesgos existentes en el sistema.
project_id	int(11)	Identificador de los proyectos existentes en el

		sistema.
iduser	int(11)	Identificador de los usuarios existentes en el sistema.
idrol	int(11)	Identificador de los roles que tienen los usuarios.
fecha	date	Fecha en la cual se dispara un factor de riesgo.

Tabla: Factor_Entorno

Nombre: factor_entorno		
Descripción: En esta tabla se almacena la relación existente entre los entornos de trabajo y los factores de riesgos en un proyecto determinado.		
Atributo	Tipo	Descripción
idcondiciones_trabajo	int(11)	Identificador de los entornos de trabajo existentes en el sistema.
idfactor	int(11)	Identificador de los factores de riesgos existentes en el sistema.
project_id	int(11)	Identificador de los proyectos existentes en el sistema.
peso	double	Grado con el cual el entorno de trabajo puede influir en un factor de riesgo.

Tabla: Riesgo_Entorno

Nombre: riesgo_entorno		
Descripción: En esta tabla se almacena la relación existente entre los entornos de trabajo y los riesgos en un proyecto determinado.		
Atributo	Tipo	Descripción
idcondiciones_trabajo	int(11)	Identificador de los entornos de trabajo existentes en el sistema.
idriesgo	int(11)	Identificador de los riesgos existentes en el sistema.
project_id	int(11)	Identificador de los proyectos existentes en el sistema.

peso	double	Grado con el cual el entorno de trabajo puede influir en un riesgo.
------	--------	---

Tabla: Factor_Riesgo

Nombre: factor_riesgo		
Descripción: En esta tabla se almacena la relación existente entre los riesgos y los factores de riesgos en un proyecto determinado.		
Atributo	Tipo	Descripción
idriesgo	int(11)	Identificador de los riesgos existentes en el sistema.
idfactor	int(11)	Identificador de los factores de riesgos existentes en el sistema.
project_id	int(11)	Identificador de los proyectos existentes en el sistema.
peso	double	Grado con el cual un factor de riesgo puede influir en un riesgo.

Tabla: Projects_Usuario

Nombre: projects_usuario		
Descripción: En esta tabla se almacenan todos usuarios y los respectivos proyectos a los que pertenecen.		
Atributo	Tipo	Descripción
project_id	int(11)	Identificador de los proyectos existentes en el sistema.
iduser	int(11)	Identificador de los usuarios existentes en el sistema.
idrol	int(11)	Identificador de los roles que tienen los usuarios del sistema.

Tabla: Proyecto_Entorno

Nombre: proyecto_entorno		
Descripción: En esta tabla se almacenan los valores de los entornos de trabajo de los proyectos.		
Atributo	Tipo	Descripción
project_id	int(11)	Identificador de los proyectos existentes en el sistema.
idcondiciones_trabajo	int(11)	Identificador de los entornos de trabajo existentes en el sistema.
valor	varchar(20)	Valor que toma el entorno de trabajo en un proyecto.

Anexo 9: Pruebas de Caja Negra.

Pruebas de Caja Negra para el caso de uso <Insertar entorno de trabajo>

Condiciones de Entrada	Casos Válidos	Casos no Válidos
Nombre	Nombre	Dejar el campo vacío. Insertar números.
Descripción	Descripción	Dejar el campo vacío.

Caso de uso	Insertar entorno de trabajo
Caso de prueba	Adicionar un entorno de trabajo entrando los datos correctamente.
Entrada	
El líder introduce correctamente los datos para adicionar un entorno de trabajo. Nombre: "Local". Descripción: "Lugar en el cual se va a desarrollar el proyecto".	
Resultados	El sistema adiciona el entorno de trabajo a la Base de Datos y te muestra nuevamente el formulario de adicionar.
Condiciones	Datos del entorno de trabajo entrados correctamente.

Caso de uso	Insertar entorno de trabajo
Caso de prueba	Adicionar un entorno de trabajo dejando un campo vacío.
Entrada	
El líder introduce incorrectamente los datos para adicionar un entorno de trabajo. Nombre: "". Descripción: " Lugar en el cual se va a desarrollar el proyecto".	
Resultados	El sistema muestra un mensaje de alerta notificando el error ("Debe introducir el nombre del entorno de trabajo").
Condiciones	No se introducen datos en el campo nombre.

Pruebas de Caja Negra para el caso de uso <Gestionar entorno de trabajo>

Condiciones de Entrada	Casos Válidos	Casos no Válidos
Nombre	Nombre	Dejar el campo vacío. Insertar números.
Descripción	Descripción	Dejar el campo vacío.

Escenario <Modificar entorno de trabajo>

Caso de uso	Gestionar entorno de trabajo (Modificar entorno de trabajo)
Caso de prueba	Modificar un entorno de trabajo entrando los datos correctamente.
Entrada	
El administrador introduce correctamente los datos para modificar un entorno de trabajo. Nombre: "Local". Descripción: "Sitio donde se desarrollará el proyecto informático".	
Resultados	El sistema modifica los datos del entorno de trabajo en la Base de Datos.
Condiciones	Datos del entorno de trabajo entrados correctamente.

Caso de uso	Gestionar entorno de trabajo (Modificar entorno de trabajo)
Caso de prueba	Modificar un entorno de trabajo insertando números en el campo nombre.
Entrada	

El administrador introduce incorrectamente los datos para adicionar un entorno de trabajo. Nombre: "15344". Descripción: " Sitio donde se desarrollará el proyecto informático".	
Resultados	El sistema muestra un mensaje de alerta ("Ha introducido caracteres no admisibles en el nombre").
Condiciones	Se introducen datos incorrectos en el campo nombre.

Pruebas de Caja Negra para el caso de uso <Gestionar relaciones riesgo_entorno>

Condiciones de Entrada	Casos Válidos	Casos no Válidos
Riesgo	Seleccionar el riesgo.	No seleccionar el riesgo.
Entorno de trabajo	Seleccionar el entorno de trabajo.	No seleccionar el entorno de trabajo.
Peso	Números del 0 al 1.	Dejar el campo vacío. Números fuera del rango de 0 a 1. Insertar letras.

Escenario <Insertar relación riesgo_entorno>

Caso de uso	Gestionar relación riesgo_entorno(Insertar relación riesgo_entorno)
Caso de prueba	Adicionar el peso de la relación entrando los datos correctamente.
Entrada	
El líder introduce correctamente los datos para adicionar el peso de la relación entre el riesgo y el entorno de trabajo. Riesgo: "Fallas en el fluido eléctrico". Entorno de trabajo: "Docente 5". Peso: "0.2".	
Resultados	El sistema adiciona el peso de la relación entre el riesgo y el entorno de trabajo a la Base de Datos.
Condiciones	Datos de la relación entrados correctamente.

Caso de uso	Gestionar relación riesgo_entorno(Insertar relación riesgo_entorno)
Caso de prueba	Adicionar el peso de la relación sin haber seleccionado un entorno de trabajo.
Entrada	
El líder no selecciona un entorno de trabajo. Riesgo: " Fallas en el fluido eléctrico". Entorno de trabajo: "-----". Peso: "0.5".	
Resultados	El sistema muestra un mensaje de alerta notificando el error ("Debe seleccionar un entorno de trabajo").
Condiciones	No se seleccionó ningún entorno de trabajo.

Caso de uso	Gestionar relación riesgo_entorno (Insertar relación riesgo_entorno)
Caso de prueba	Insertar la relación dejando un campo obligatorio vacío.
Entrada	
El líder no introduce el peso de la relación entre el riesgo y el entorno de trabajo. Riesgo: "Fallas en el fluido eléctrico". Entorno de trabajo: "Docente 5". Peso: " "	
Resultados	El sistema muestra un mensaje de alerta notificando el error ("Debe introducir el peso de la relación").
Condiciones	No se introduce el dato en el campo peso.

Escenario <Modificar relación riesgo_entorno>

Caso de uso	Gestionar relación riesgo_entorno(Modificar relación riesgo_entorno)
Caso de prueba	Modificar el peso de la relación entrando los datos correctamente.
Entrada	

El líder introduce correctamente los datos para modificar el peso de la relación entre el riesgo y el entorno de trabajo. Riesgo: "Fallas en el fluido eléctrico". Entorno de trabajo: "Docente 5". Peso: "0.1".	
Resultados	El sistema modifica el peso de la relación entre el riesgo y el entorno de trabajo en la Base de Datos.
Condiciones	Datos de la relación entrados correctamente.

Caso de uso	Gestionar relaciones riesgo_entorno(Modificar relación riesgo_entorno)
Caso de prueba	Modificar una relación insertando letras en el campo peso.
Entrada	
El líder introduce incorrectamente los datos necesarios para adicionar una relación. Riesgo: "Fallas en el fluido eléctrico". Entorno de trabajo: "Docente 5". Peso: "prueba".	
Resultados	El sistema muestra un mensaje de alerta ("Ha introducido caracteres no admisibles en el Peso de la Relación").
Condiciones	Se introducen incorrectamente el dato en el campo peso.

Caso de uso	Gestionar relación riesgo_entorno(Modificar relación riesgo_entorno)
Caso de prueba	Modificar la relación entrando incorrectamente el peso.
Entrada	
El líder introduce incorrectamente el dato para modificar el peso de la relación entre el riesgo y el entorno de trabajo. Riesgo: "Fallas en el fluido eléctrico". Entorno de trabajo: "Docente 5". Peso: "15".	
Resultados	El sistema muestra un mensaje de alerta notificando el error ("El peso está en el

	rango de 0 a 1”).
Condiciones	Dato entrado fuera del rango de valores permitidos.

Pruebas de Caja Negra para el caso de uso <Gestionar relación factor_entorno>

Condiciones de Entrada	Casos Válidos	Casos no Válidos
Factor de riesgo	Seleccionar el factor de riesgo.	No seleccionar el factor de riesgo.
Entorno de trabajo	Seleccionar el entorno de trabajo.	No seleccionar el entorno de trabajo.
Peso	Números del 0 al 1.	Dejar el campo vacío. Números fuera del rango de 0 a 1. Insertar letras.

Escenario <Insertar relación factor_entorno>

Caso de uso	Gestionar relación factor_entorno(Insertar relación factor_entorno)
Caso de prueba	Adicionar el peso de la relación entrando correctamente el dato.
Entrada	
El líder introduce correctamente los datos para adicionar el peso de la relación entre el riesgo y el entorno de trabajo. Factor de riesgo: ” Fallas con el fluido eléctrico”. Entorno de trabajo: ”Docente 5”. Peso: ”0.4”.	
Resultados	El sistema adiciona el peso de la relación entre el factor de riesgo y el entorno de trabajo a la Base de Datos.
Condiciones	Datos de la relación entrados correctamente.

Caso de uso	Gestionar relación factor_entorno (Insertar relación factor_entorno)
Caso de prueba	Adicionar el peso de la relación sin haber seleccionado un factor de riesgo.
Entrada	

El líder no selecciona un factor de riesgo. Factor de riesgo: "-----". Entorno de trabajo: " Docente 5". Peso: "0.5".	
Resultados	El sistema muestra un mensaje de alerta notificando el error ("Debe seleccionar un factor de riesgo").
Condiciones	No se seleccionó ningún factor de riesgo.

Caso de uso	Gestionar relación factor_entorno(Insertar relación factor_entorno)
Caso de prueba	Adicionar una relación insertando letras en el campo peso.
Entrada	
El líder introduce incorrectamente un dato necesario para adicionar una relación. Factor de riesgo: " Fallas con el fluido eléctrico". Entorno de trabajo: "Docente 5". Peso: "prueba".	
Resultados	El sistema muestra un mensaje de alerta ("Ha introducido caracteres no admisibles en el Peso de la Relación").
Condiciones	Se introduce incorrectamente el dato en el campo peso.

Escenario <Modificar relación factor_entorno>

Caso de uso	Gestionar relación factor_entorno(Modificar relación factor_entorno)
Caso de prueba	Modificar la relación entrando correctamente el peso.
Entrada	
El líder introduce correctamente el peso para modificar la relación entre el factor de riesgo y el entorno de trabajo. Factor de riesgo: " Fallas con el fluido eléctrico". Entorno de trabajo: "Docente 5". Peso: "0.1".	
Resultados	El sistema modifica el peso de la relación entre el factor de riesgo y el entorno de

	trabajo en la Base de Datos.
Condiciones	Datos de la relación entrados correctamente.

Caso de uso	Gestionar relación factor_entorno(Modificar relación factor_entorno)
Caso de prueba	Modificar la relación dejando un campo obligatorio vacío.
Entrada	
El líder no introduce el peso de la relación entre el factor de riesgo y el entorno de trabajo. Factor de riesgo: ” Fallas con el fluido eléctrico”. Entorno de trabajo: ”Docente 5”. Peso: ” ”.	
Resultados	El sistema muestra un mensaje de alerta notificando el error (“Debe introducir el Peso de la relación”).
Condiciones	No se introduce el dato en el campo peso.

Caso de uso	Gestionar relación factor_entorno (Modificar relación factor_entorno)
Caso de prueba	Modificar la relación entrando incorrectamente el peso.
Entrada	
El líder introduce incorrectamente el peso de la relación entre el factor de riesgo y el entorno de trabajo. Factor de riesgo: ” Fallas con el fluido eléctrico”. Entorno de trabajo: ”Docente 5”. Peso: ”10”.	
Resultados	El sistema muestra un mensaje de alerta notificando el error (“El peso está en el rango de 0 a 1”).
Condiciones	Dato entrado fuera del rango de valores permitidos.

Pruebas de Caja Negra para el caso de uso <Gestionar valor entorno de trabajo>

Condiciones de Entrada	Casos Válidos	Casos no Válidos
Entorno	Seleccionar un entorno.	No seleccionar el entorno.

Valor	Pueden ser números o cadena de caracteres o ambos.	Dejar el campo vacío.
-------	--	-----------------------

Escenario <Insertar valor entorno de trabajo>

Caso de uso	Gestionar valor entorno de trabajo (Insertar valor entorno de trabajo)
Caso de prueba	Adicionar el valor de un entorno de trabajo entrando los datos del mismo correctamente.
Entrada	
El líder introduce correctamente los datos para adicionar el valor de un entorno de trabajo. Entornos existentes: "Local". Valor: "Laboratorio 208".	
Resultados	El sistema adiciona el valor del entorno de trabajo a la Base de Datos.
Condiciones	Datos entrados correctamente.

Caso de uso	Gestionar valor entorno de trabajo (Insertar valor entorno de trabajo)
Caso de prueba	Adicionar el valor de un entorno sin seleccionar un entorno.
Entrada	
El líder no selecciona un entorno de trabajo. Entornos existentes: "-----". Valor: "Laboratorio 201".	
Resultados	El sistema muestra un mensaje de alerta notificando el error ("Debe seleccionar un entorno de trabajo").
Condiciones	No se selecciona un entorno de trabajo.

Escenario <Modificar valor entorno de trabajo>

Caso de uso	Gestionar valor entorno de trabajo (Modificar valor entorno de trabajo)
Caso de prueba	Modificar datos del valor del entorno de trabajo entrando correctamente los datos.
Entrada	

El líder introduce correctamente el dato para modificar el valor del entorno de trabajo. Entornos existentes: "Local". Valor: "Laboratorio 207".	
Resultados	El sistema modifica el valor del entorno de trabajo en la Base de Datos.
Condiciones	Se introduce correctamente el valor.

Caso de uso	Gestionar valor entorno de trabajo (Modificar valor entorno de trabajo)
Caso de prueba	Modificar el valor del entorno de trabajo dejando un campo vacío.
Entrada	
El líder no introduce el valor del entorno de trabajo. Entornos existentes: "Local". Valor: "".	
Resultados	El sistema muestra un mensaje de alerta notificando el error ("Debe introducir el valor").
Condiciones	No se introduce el dato en el campo valor.

Pruebas de Caja Negra para el caso de uso <Gestionar usuarios>

Condiciones de Entrada	Casos Válidos	Casos no Válidos
Nombre	Nombre	Dejar el campo vacío. Insertar números.
Usuario	Usuario	Dejar el campo vacío.
Contraseña	Puede tener números, letras o ambos siempre que la cantidad de caracteres sea mayor que 2.	Dejar campo vacío.
Repetir contraseña	Igual que el campo contraseña.	Dejar el campo vacío.
Proyecto	Seleccionar un proyecto.	No seleccionar un proyecto.
Rol	Seleccionar un rol.	No seleccionar un rol.
Departamento	Puede tener letras, números o ambos.	Dejar el campo vacío.

Escenario <Insertar usuarios>

Caso de uso	Gestionar usuarios (Insertar usuarios)
Caso de prueba	Adicionar un usuario entrando los datos correctamente.
Entrada	
<p>El líder introduce correctamente los datos para adicionar un nuevo usuario.</p> <p>Nombre:"Ernesto".</p> <p>Usuario:"evargas".</p> <p>Contraseña:"*****".</p> <p>Repetir contraseña:"*****".</p> <p>Proyecto:"Multisaber".</p> <p>Rol:"Usuario".</p> <p>Departamento:"Gestión".</p>	
Resultados	El sistema adiciona un nuevo usuario a la Base de Datos.
Condiciones	Datos del usuario entrados correctamente.

Caso de uso	Gestionar usuarios (Insertar usuarios)
Caso de prueba	Adicionar un usuario dejando un campo vacío.
Entrada	
<p>El líder introduce incorrectamente los datos para adicionar un usuario.</p> <p>Nombre:".</p> <p>Usuario:"evargas".</p> <p>Contraseña:"*****".</p> <p>Repetir contraseña:"*****".</p> <p>Proyecto:"Multisaber".</p> <p>Rol:"Usuario".</p> <p>Departamento:"Gestión".</p>	
Resultados	El sistema muestra un mensaje de alerta notificando el error ("Debe introducir el nombre del usuario").
Condiciones	No se introduce el dato en el campo nombre.

Escenario <Modificar usuarios >

Caso de uso	Gestionar usuarios (Modificar usuarios)
Caso de prueba	Modificar un usuario entrando los datos correctamente.
Entrada	
<p>El líder introduce correctamente los datos para modificar a un usuario.</p> <p>Nombre:"Ernesto".</p> <p>Usuario:"evperez".</p> <p>Contraseña:"*****".</p> <p>Repetir contraseña:"*****".</p> <p>Proyecto:"Multisaber".</p> <p>Rol:"Líder".</p> <p>Departamento:"Gestión".</p>	
Resultados	El sistema modifica los datos del usuario en la Base de Datos.
Condiciones	Datos del usuario entrados correctamente.

Caso de uso	Gestionar usuario (Modificar usuarios)
Caso de prueba	Modificar los datos de un usuario insertando números en el campo nombre.
Entrada	
<p>El líder introduce incorrectamente los datos para modificar a un usuario.</p> <p>Nombre:"Ernesto555".</p> <p>Usuario:"evperez".</p> <p>Contraseña:"*****".</p> <p>Repetir contraseña:"*****".</p> <p>Proyecto:"Multisaber".</p> <p>Rol:"Líder".</p> <p>Departamento:"Gestión".</p>	
Resultados	El sistema muestra un mensaje de alerta notificando el error ("Ha introducido caracteres no admisibles en el nombre").
Condiciones	Se introducen datos de forma incorrecta.

Pruebas de Caja Negra para el caso de uso <Notificar disparo>

Escenario <Notificar disparo riesgo>

Condiciones de Entrada	Casos Válidos	Casos no Válidos
Riesgo	Seleccionar riesgo.	No seleccionar un riesgo.
Fecha de disparo	Seleccionar fecha de disparo.	No seleccionar fecha de disparo. Introducir caracteres no válidos (letras).

Caso de uso	Notificar disparo (Notificar disparo riesgo)
Caso de prueba	Notificar disparo de un riesgo entrando los datos correctamente.
Entrada	
El líder introduce correctamente los datos para notificar el disparo de un riesgo. Riesgo: "Falta de puestos de trabajo". Fecha de disparo: "20/05/2009".	
Resultados	El sistema adiciona la fecha del disparo de un riesgo a la Base de Datos.
Condiciones	Se introduce correctamente la fecha.

Caso de uso	Notificar disparo (Notificar disparo riesgo).
Caso de prueba	Notificar disparo de un riesgo sin seleccionar un riesgo.
Entrada	
El líder no selecciona un riesgo. Riesgo: "-----". Fecha de Disparo: "20/05/2009".	
Resultados	El sistema muestra un mensaje de alerta notificando el error ("Debe seleccionar un riesgo").
Condiciones	No se seleccionó un riesgo.

Caso de uso	Notificar disparo (Notificar disparo riesgo)
Caso de prueba	Notificar disparo de un riesgo entrando caracteres no permitidos.

Entrada	
El líder introduce incorrectamente los datos para adicionar una fecha de disparo. Riesgo: "Falta de puestos de trabajo". Fecha de Disparo: "prueba".	
Resultados	El sistema muestra un mensaje de alerta notificando el error ("El valor especificado no es válido").
Condiciones	No se introducen correctamente los datos.

Escenario <Notificar disparo factor riesgo>

Condiciones de Entrada	Casos Válidos	Casos no Válidos
Factor de riesgo	Seleccionar factor de riesgo.	No seleccionar un factor de riesgo.
Fecha de disparo	Seleccionar fecha de disparo.	No seleccionar fecha de disparo. Introducir caracteres no válidos (letras).

Caso de uso	Notificar disparo (Notificar disparo factor riesgo)
Caso de prueba	Notificar disparo de un factor de riesgo entrando la fecha correctamente.
Entrada	
El líder introduce correctamente la fecha para notificar el disparo de un factor de riesgo. Factor de Riesgo: "Falla en el fluido eléctrico". Fecha de Disparo: "19/06/2009".	
Resultados	El sistema adiciona la fecha del disparo de un factor de riesgo a la Base de Datos.
Condiciones	Datos insertado correctamente.

Caso de uso	Notificar disparo (Notificar disparo factor riesgo)
Caso de prueba	Notificar disparo de un factor de riesgo sin seleccionar un factor.
Entrada	
El líder no selecciona un factor de riesgo.	

Factor de riesgo: "-----".	
Fecha de disparo: "19/06/2009".	
Resultados	El sistema muestra un mensaje de alerta notificando el error ("Debe seleccionar un factor de riesgo").
Condiciones	No se seleccionó un factor de riesgo.

Caso de uso	Notificar disparo (Notificar disparo factor riesgo).
Caso de prueba	Notificar disparo de un factor de riesgo sin seleccionar una fecha de disparo.
Entrada	
El líder no selecciona una fecha. Riesgo: "Falla en el fluido eléctrico". Fecha de disparo: " "	
Resultados	El sistema muestra un mensaje de alerta notificando el error ("Debe seleccionar una fecha").
Condiciones	No se selecciona la fecha de disparo.

Pruebas de Caja Negra para el caso de uso <Generar reportes>

Condiciones de Entrada	Casos Válidos	Casos no Válidos
Tipo	Seleccionar tipo de reporte.	No seleccionar un tipo de reporte.
Parámetros	Seleccionar el parámetro.	No seleccionar un parámetro.
Valor	Seleccionar un valor. Introducir números del 0 al 1 con dos lugares después de la coma.	No seleccionar un valor. Introducir letras. Introducir número fuera de rango. Introducir un número con dos lugares después de la coma.
Operador	Seleccionar un operador.	No seleccionar un operador.

Caso de uso	Generar reportes
Caso de prueba	Generar reportes entrando los datos correctamente.
Entrada	
El usuario introduce correctamente los datos para generar un reporte. Tipo: "Riesgo". Proyecto: "Multisaber". Parámetros: "Clasificación". Valores: "Personal".	
Resultados	El sistema muestra el reporte deseado.
Condiciones	Se introducen todos los datos correctamente.

Caso de uso	Generar reportes
Caso de prueba	Generar reportes sin seleccionar el tipo de reporte.
Entrada	
El usuario no selecciona el tipo de reporte. Tipo: "-----". Proyecto: "Multisaber". Parámetros: "Clasificación". Valores: "Personal".	
Resultados	El sistema muestra un mensaje de alerta ("Debe seleccionar el tipo de reporte").
Condiciones	No se selecciona el tipo de reporte.

Caso de uso	Generar reportes.
Caso de prueba	Generar reportes entrando en el campo valor un número con más de dos lugares después de la coma.
Entrada	

El usuario introduce un número con tres lugares después de la coma. Tipo: "Riesgo". Proyecto: "Multisaber". Parámetros: "Probabilidad". Operador: ">=". Valores: "0.213".	
Resultados	El sistema muestra un mensaje de alerta ("No puede exceder dos lugares después de la coma").
Condiciones	Introduce un número con tres lugares después de la coma.