



**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**  
**Facultad 6**

**SPLRISK: Sistema para la Evaluación de Riesgos en las LPS**  
**del departamento de Geoinformática**

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS

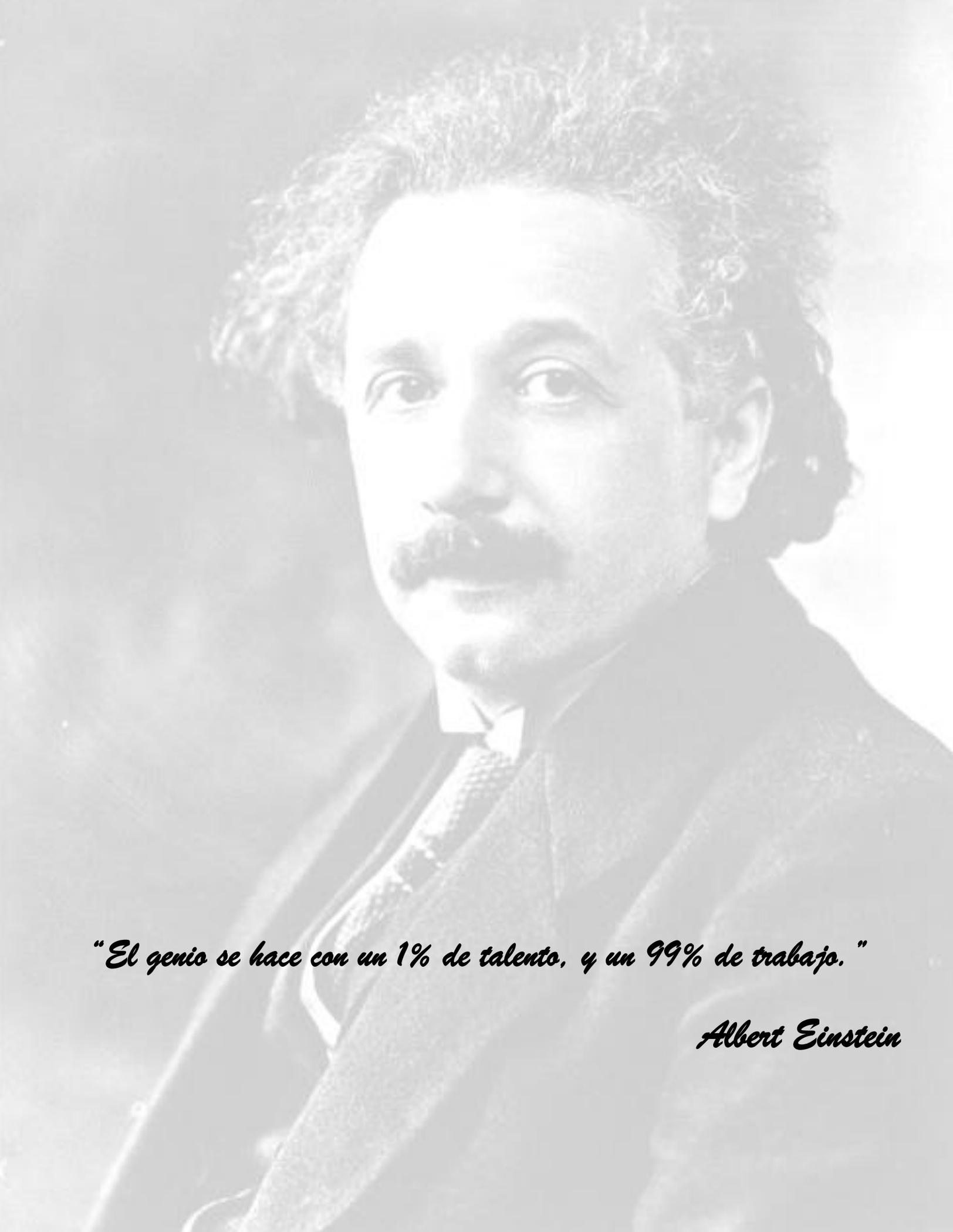
**AUTORES:**

Lázaro William Brito León  
Ariel Valladares Pérez

**TUTOR:**

M.Sc. Vladimir Martell Fernández

La Habana, mayo de 2014  
"Año 56 de la Revolución"



*"El genio se hace con un 1% de talento, y un 99% de trabajo."*

*Albert Einstein*

## **DECLARACIÓN DE AUTORÍA**

Declaramos por este medio que nosotros, Ariel Valladares Pérez, con carné de identidad 90091034484 y Lázaro W. Brito León, con carnet de identidad 90080938540, somos autores de este trabajo y que autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste, firmamos la presente declaración jurada de autoría, en La Habana, a los 16 días del mes de junio del año 2014.

---

**Ariel Valladares Pérez**

Autor

---

**Lázaro W. Brito León**

Autor

---

**M.Sc. Vladimir Martell Fernández**

Tutor

# AGRADECIMIENTOS

*Lázaro William*

*Agradezco a mi familia por estar siempre a mi lado y darme la educación que no se alcanza en una escuela...*

*Especialmente a mi papá Walfrido Brito Guerra, por ser un ejemplo de rectitud y principios, y demostrarme con sus actos que debemos luchar por nuestros sueños y por las personas que amamos...*

*A mis tíos Félix Raúl y Vladimir, y mis tías Carelia, Amarilis, Maritza y Sofía por convertirse con sus acciones desinteresadas en mis otros padres y madres...*

*A mi otra mamá María (Mary), por brindarme el mismo calor y amor de madre con que trata a su propio hijo, y porque, a pesar del poco tiempo que vivimos juntos y que por razones de la vida hoy no sea así, siempre serás mi otra mamá y tendrás un lugar especial en mi corazón...*

*A mis primos Anilka (Nana) y Robin (el Jabao) por ser mis hermanos incondicionales y nunca olvidar, a pesar de la distancia que nos impone la vida, todas las tormentas y festividades que vivimos juntos...*

*A mis primos Marielsy y Jorge (yoyo), por permitirme desinteresadamente aprovecharme de ese lazo que nos une para resolver mis propios problemas cada vez que lo necesito...*

*A mi madrina María Rondón, por ser mi confidente y consejera en los momentos de duda y desacuerdo...*

*A mi tutor Vladimir, por ser el máximo responsable de que hoy yo esté aquí, gracias al esfuerzo y empeño con que asumió la tarea de guiarnos, y ser a lo largo de esta dura jornada más que amigo, un padre para nosotros...*

*A mi jefe de proyecto y amigo Alain, por brindarme apoyo y consejos, y estar disponible para evacuar nuestras dudas con mucha paciencia, sacrificando tiempo y esfuerzo...*

*A mi compañero de tesis y "amigo" Ariel (el Flako), por brindarme su apoyo desde que nos conocimos en primer año, confiar en mí en todo momento, y ser mi pilar de apoyo y mano derecha en esta contienda...*

*A mi amiga de muchos años Arianna (Ari), por sobreponerse a todos sus problemas y ser mi paño de lágrimas en los momentos más duros de mi vida, y demostrarme que la verdadera amistad existe...*

*A los que han sido durante estos cinco años junto a Ariel mis "amigos y hermanos" Frank (el metra) y Yosvany (el cabezón), gracias por permitirme formar parte de sus vidas, espero que nuestra amistad no se pierda a pesar de que nos separemos al terminar la Universidad...*

*A todos los demás que hicieron soportables estos años de sacrificio, con los que compartí tantos momentos lindos dentro y fuera de la escuela. Mencionar nombres sería demasiado complicado y no me perdonaría dejar alguno fuera. Ustedes y yo sabemos quiénes son y los más cercanos están sin falta hoy aquí...*

# AGRADECIMIENTOS

*Ariel*

*A mi hermana y mi hermanito por ser motivos de mi inspiración en la vida.*

*A mi tías, mi tíos, mis primos y a toda mi familia en general por todo el apoyo, el cariño y la confianza que me han dado durante estos años de mi vida.*

*A mi tutor Vladimir por todo el esfuerzo, el apoyo, la ayuda y la preocupación durante la realización de este trabajo.*

*A todos mis amigos por el apoyo, la amistad y la ayuda brindada en estos años en la universidad, convirtiéndose en mi otra familia.*

*A mi jefe de proyecto Alain y al profesor Yudiel (figura) por la ayuda prestada durante la elaboración del trabajo.*

*A todos los profesores que de una forma u otra dieron su aporte en mi formación personal y profesional.*

*A todos Gracias...*

# DEDICATORIA

*Lázaro William*

*Dedicado a mis abuelos*

*Jesús Joaquín Brito Sánchez, por ser un ejemplo de padre y, porque a pesar de haber vivido casi toda mi vida alejado de la mayor parte de su familia ha sabido estar presente cada día...*

*Elsa Guerra Basulto y Félix Ramón León Rodríguez, que hoy no están presentes físicamente, pero siempre estarán en mi corazón por ser, con sus virtudes y defectos, los mejores padres y madres que cualquiera pudiera desear. A ella por ser madre y padre para todos sus hijos, y ser ejemplo de firmeza y sacrificio; y a él porque a pesar de todos sus problemas, siempre estuvo presente para su familia con el corazón abierto y lleno de amor...*

*Y muy en especial a mi abuela Deborah Ester Ramos Ruiz, por haber sido y ser una madre incondicional para mí, por compartir mis alegrías y triunfos y apoyarme en todas las dificultades, por ser mi luz y mi guía y darme fuerzas para seguir adelante. Eres y serás por siempre la dueña de mi corazón "mamá"...*

## DEDICATORIA

*Ariel*

*A mis padres:*

*Por todo el amor, sacrificio, apoyo y comprensión que me han ofrecido en todo este tiempo, en especial a mi mamá por transformar los malos momentos en gratos recuerdos.*

*A mi tío Paulino:*

*Por todo el amor, el apoyo y la confianza que de él he recibido, por saber conducirme por el camino correcto y saber sacar de mí lo mejor como persona.*

*A mi abuela María Caridad:*

*Por el apoyo, el sacrificio, la abnegación y el amor que me ha regalado durante mi vida, por saber aconsejarme y defenderme en todo momento.*

## RESUMEN

Actualmente los productos de software que se desarrollan a nivel mundial son más complejos y se requiere de una mayor rapidez, eficacia y calidad en el desarrollo de los mismos. Considerando estos aspectos de vital importancia para satisfacer las necesidades de los clientes surge la idea y el concepto de "reutilizar", reutilizar un componente ya previamente elaborado y probado en anteriores productos. Las Líneas de Productos de Software se han convertido en un modelo de producción muy seguido por las instituciones desarrolladoras de aplicaciones informáticas, atendiendo a las facilidades y beneficios en tiempo de producción y reutilización de código que brindan, lo que supone mejoras en el costo de ingeniería y en el tiempo de entrega. Siguiendo este marco de desarrollo, es necesario entonces que la gestión de proyectos se enfoque específicamente a este modelo de producción, centrando su atención en las afectaciones (riesgo u oportunidad), que pueden presentarse durante la ejecución de cada proyecto dentro de ella. La investigación que se presenta consiste en el análisis, diseño, implementación y pruebas de una aplicación informática para la evaluación de los riesgos en las Líneas de Productos de Software del Departamento de Geoinformática y persigue disminuir las limitaciones que este proceso presenta en cuanto al tiempo, la complejidad, inexactitud y seguridad.

**Palabras Clave:** Aplicación informática, evaluación, Líneas de productos de software, riesgos.

## ABSTRACT

Software products that are developed today are more complex and require greater speed, efficiency and quality in its development. Considering these aspects critical to meet the customer needs the concept of "reuse"; that is, to reuse a component already developed and tested in previous products. The Software Product Lines have become a production model followed by many institutions of applications developers, considering the facilities and benefits in production time and they provide code reuse, which means improvements in engineering costs and time delivery. Therefore, it is necessary then that the project management approaches specifically to this production model, focusing on the risks or opportunities that may arise during the execution of each project. The research presented is the analysis, design, implementation and testing of a software application for the evaluation of risks in Software Product Lines Department of Geoinformatics and aims to reduce the limitations that this process occurs in terms of time, complexity, inaccuracy and security.

**Keywords:** Computer application, evaluation, risks, Software Product Lines.

# ÍNDICE DE CONTENIDOS

INTRODUCCIÓN	1
Capítulo 1: “FUNDAMENTACIÓN DE LA INVESTIGACIÓN”	6
1.1. Introducción	6
1.2. Conceptos asociados al dominio del problema	6
1.2.1. <i>Aplicación informática</i>	6
1.2.2. <i>Líneas de Productos de Software</i>	6
1.2.3. <i>Riesgos</i>	7
1.2.4. <i>Gestión de riesgos</i>	8
1.3. El proceso de desarrollo de software	9
1.4. Caracterización del objeto de estudio	9
1.4.1. <i>El Proceso de Evaluación de Riesgos</i>	9
1.4.2. <i>La Evaluación de Riesgos en LPS</i>	11
1.4.3. <i>La Evaluación de Riesgos en las LPS del Departamento de Geoinformática</i>	12
1.5. Soluciones existentes	15
1.6. Metodologías, herramientas y tecnologías empleadas	18
1.6.1. <i>Metodología de desarrollo de software</i>	18
1.6.2. <i>Lenguaje de modelado como soporte a la metodología</i>	21
1.6.3. <i>Herramienta para el modelado</i>	21
1.6.4. <i>Lenguajes de programación</i>	22
1.6.5. <i>Framework de desarrollo</i>	24
1.6.6. <i>Sistema Gestor de Bases de Datos</i>	25
1.7. Conclusiones	27
Capítulo 2: “Descripción de la solución propuesta”	28
2.1. Introducción	28
2.2. Modelo de dominio	28
2.2.1. <i>Diagrama de clases del dominio</i>	28
2.3. Requisitos	29
2.3.1. <i>Requisitos Funcionales</i>	30
2.3.2. <i>Requisitos No Funcionales</i>	31

2.4.	Actores del Sistema -----	32
2.5.	Modelo de Sistema -----	33
2.5.1.	<i>Diagrama de Casos de Uso del Sistema</i> -----	33
2.5.2.	Descripción textual de los Casos de Uso del Sistema -----	34
2.5.3.	<i>Patrones de Casos de Uso utilizados</i> -----	35
2.6.	Elementos fundamentales de la arquitectura-----	36
2.6.1.	<i>Estilos y Patrones arquitectónicos</i> -----	36
2.6.2.	<i>Patrones de diseño</i> -----	37
2.7.	Modelo de Diseño-----	40
2.7.1.	<i>Diagrama de clases del diseño</i> -----	40
2.8.	<i>Modelo entidad-relación</i> -----	41
2.9.	Modelo de clases persistentes-----	42
2.10.	Conclusiones parciales -----	43
Capítulo 3: "Implementación y evaluación de la SPLRISK"-----		44
3.1.	Introducción -----	44
3.2.	Modelo de implementación -----	44
3.2.1.	<i>Diagrama de componentes</i> -----	44
3.3.	Modelo de despliegue-----	46
3.4.	El proceso de pruebas-----	48
3.4.1.	<i>Pruebas de Caja Negra</i> -----	48
3.4.2.	<i>Resultados de las pruebas</i> -----	51
3.5.	Conclusiones-----	52
Conclusiones Generales -----		53
RECOMENDACIONES-----		54

## ÍNDICE DE TABLAS

Tabla 1: Involucrados, entrada, acciones y salida de la propuesta. -----	13
Tabla 2: Descripción de los actores del sistema. -----	33
Tabla 3: Descripción textual del Caso de Uso Gestionar Riesgos. -----	34
Tabla 4: Descripción por escenarios. -----	49
Tabla 5: Variables del Caso de Prueba. -----	50
Tabla 6: Comprobación de variables. -----	51
Tabla 7: No Conformidades del Proceso de pruebas -----	52

## ÍNDICE DE FIGURAS

Fig. 1: Ubicación de la propuesta. -----	13
Fig. 2: Ciclo de vida de AUP. -----	20
Fig. 3: Diagrama de clases del dominio. -----	29
Fig. 4: Diagrama de Casos de Uso del Sistema. -----	33
Fig. 5: Representación visual del patrón CRUD. -----	36
Fig. 6: Representación visual del patrón Múltiples Actores. -----	36
Fig. 7: Diagrama de clases de Diseño del CU Gestionar Riesgo. -----	41
Fig. 8: Diagrama de Entidad - Relación. -----	42
Fig. 9: Diagrama de Clases Persistentes. -----	43
Fig. 10: Diagrama de Componentes. -----	44
Fig. 11: Representación de la Vista del Diagrama de Componentes. -----	45
Fig. 12: Representación del Controlador del Diagrama de Componentes. -----	45
Fig. 13: Representación del Modelo del Diagrama de Componentes. -----	46
Fig. 14: Diagrama de Despliegue. -----	47

## INTRODUCCIÓN

Tanto la informática como las telecomunicaciones, y en sentido general, todo lo relacionado con las **nuevas tecnologías**, tienen un **auge** imparable en el mercado; y es que todas las organizaciones empresariales se valen de alguna forma de estas, desde las simples computadoras en los puestos de trabajo convencionales, hasta llegar a las plataformas de redes sociales y servidores, o las nubes informáticas.

Las tecnologías del mundo digital están destinadas a triunfar, por tratarse de un sector que abarca un sin número de puestos de trabajo. Cada vez es mayor la necesidad de trabajadores con conocimientos en esta área, capaces de cubrir las demandas de las empresas. En la actualidad, los medios tradicionales se han visto obligados a apoyarse en sistemas digitales para crear su espacio virtual y así satisfacer las necesidades sociales y tener un impacto positivo en la población.

En una entrevista a Bill Gates hecha por Jorge Luis Cebrián para diario El País acota que *“el impacto generado por el mundo digital es equivalente al de la revolución industrial; no mayor, pero si equivalente. La diferencia que magnifica la situación actual, es que el industrialismo se desarrolló durante muchas generaciones, mientras que el cambio digital se está llevando a cabo en sólo una generación y media o dos”* (CEBRIÁN, 2011).

Actualmente los productos de software que se desarrollan a nivel mundial son más complejos y se requiere de una mayor rapidez, eficacia y calidad en el desarrollo de los mismos. Considerando estos aspectos de vital importancia para satisfacer las necesidades de los clientes surge la idea y el concepto de **“reutilizar”**, reutilizar un componente ya previamente elaborado y probado en anteriores productos.

Un componente reutilizable es considerado como *“Una pieza de software funcional que es liberada independientemente de otras y que proporciona acceso a sus servicios a través de sus interfaces”* (MONTILVA, 2006).

Siguiendo este principio de reutilización de componentes surgen las Líneas de Productos de Software (LPS), definidas por el Instituto de Ingeniería de Software (del inglés Software Engineering Institute, SEI) como un conjunto de sistemas de software compartiendo características comunes y administradas que

satisface las necesidades específicas de un segmento de mercado particular o misión y que son desarrolladas de forma prescrita a partir de un conjunto común de elementos clave” (SEI, 2011).

Las LPS se han convertido en un modelo de producción muy seguido por las instituciones desarrolladoras de aplicaciones informáticas, atendiendo a las facilidades y beneficios en tiempo de producción y reutilización de código que brindan, lo que supone mejoras en el costo de ingeniería y en el tiempo de entrega.

Siguiendo este marco de desarrollo, es necesario entonces que la **gestión de proyectos** se enfoque específicamente al concepto de LPS, centrando su atención en las afectaciones, dígase *riesgo u oportunidad*, que pueden presentarse durante la ejecución de cada proyecto dentro de ella.

La gestión o administración de proyectos se encarga de guiar todo el proceso de planificar, captar, dinamizar, organizar talentos y administrar recursos, el cual requiere *liderar, evaluar y regular* sistemáticamente las acciones necesarias y suficientes para terminar un producto, cumpliendo además con los límites de tiempo y costo definidos con anterioridad.

Un riesgo, es definido por la Real Academia de la Lengua Española (ESPAÑOLA, 2012) como:

- La contingencia o proximidad de un daño.
- En sentido estricto, el riesgo implica solamente la posibilidad de sufrir daño o pérdida.
- En el contexto del proyecto, la identificación del riesgo también se refiere a las oportunidades (resultados positivos) así como las amenazas (resultados negativos).
- La administración de riesgos son los medios a través de los cuales la incertidumbre se maneja de forma sistemática, para aumentar la probabilidad de lograr los objetivos del proyecto.

Se puede concluir entonces que la **Gestión de los Riesgos** en proyectos informáticos de desarrollo de software se evidencia como un proceso clave dentro de la Gestión de Proyectos, el cual se divide en tres partes, la identificación de los riesgos, la evaluación o análisis y el seguimiento y control.

Las actividades referidas a la evaluación de los riesgos, concretamente en las LPS del Departamento de Geoinformática de la Universidad de las Ciencias Informáticas, se desarrollan a partir de un modelo propio obtenido como parte de una investigación científica.

La utilización de este modelo actualmente presenta algunas limitantes asociadas a su ejecución de forma manual, la imposibilidad de obtener estadísticas profundas y a la seguridad y centralización de la información que se maneja.

Sobre la ejecución de forma manual:

1. Aumenta considerablemente el tiempo dedicado a su ejecución pues cada actividad es descrita a partir de técnicas, indicaciones metodológicas y acciones a realizar muy complejas y a menudo embarazosas para su seguimiento en papel.
2. Las operaciones matemáticas descritas a través de las fórmulas, con grados de exactitud hasta de tres lugares decimales, en la mayoría de las actividades se tornan muy complejas.
3. Introduce una alta posibilidad de incurrir en errores involuntarios al manejar el procesamiento matemático y metodológico a través de miembros del equipo de desarrollo (el conocido error humano).

Sobre la imposibilidad de obtener estadísticas:

1. Como consecuencia de lo anterior, es prácticamente imposible obtener información sobre el proceso de Evaluación de Riesgos referida a porcentos, cantidades, áreas de mayor exposición a los riesgos, análisis de tendencias a partir de comportamientos históricos.

Sobre la seguridad y centralización de la Gestión de Riesgos:

1. Los riesgos son manejados en formatos no apropiados para mantener el grado de confidencialidad necesario en una actividad tan sensible como el tratamiento y evaluación de vulnerabilidades.
2. Todos los miembros del equipo no disponen constantemente de la información asociada ni de facilidades para desarrollar actividades relacionadas con los riesgos desde cualquier lugar o en cualquier momento.

Luego de constatada la problemática anterior, se propone el siguiente problema a resolver: **¿Cómo disminuir las limitaciones de tiempo, complejidad, inexactitud y seguridad del proceso de Evaluación de Riesgos de las LPS del Departamento de Geoinformática?**

Esta investigación propone como objetivo general **desarrollar una aplicación informática para automatizar el proceso de Evaluación de Riesgos en las Líneas de Productos de Software del Departamento de Geoinformática**, teniéndose como objeto de estudio **el proceso de Gestión de Riesgos** y como campo de acción **la automatización del proceso de evaluación de los riesgos en las LPS del Departamento de Geoinformática**.

Para lograr el objetivo general se desarrollan una serie de tareas las cuales se enumeran a continuación:

1. Caracterizar el proceso de Gestión de Riesgos en LPS para comprender el área objeto de información.
2. Caracterizar el proceso de Evaluación de Riesgos en la LPS de Geoinformática para identificar sus fortalezas y debilidades.
3. Valorar las soluciones existentes que responden al problema de la investigación en alguna medida, sus limitaciones y fortalezas para enfocar la propuesta de solución o facilitar el proceso de construcción de la misma.
4. Caracterizar las principales herramientas, tecnologías, lenguajes y metodologías a utilizar para la construcción de la propuesta de solución con el objetivo de utilizar sus fortalezas en el proceso de desarrollo.
5. Realizar el análisis y el diseño de la solución propuesta.
6. Implementar la solución propuesta.
7. Desarrollar el proceso de pruebas al sistema implementado.

Una vez finalizado el proceso investigativo se esperan los siguientes resultados: Una aplicación informática que automatice el proceso de Evaluación de Riesgos en LPS y la documentación técnica del proceso ingenieril.

En el desarrollo de la investigación se hace uso de los métodos científicos siguientes:

Métodos teóricos:

- **El analítico-sintético**, ya que a partir del análisis de documentos se profundiza en las teorías y tendencias relacionadas con el tema para identificar y sintetizar los elementos más importantes y de mayor impacto para el desarrollo del trabajo.

- **El histórico-lógico**, que nos permite estudiar la trayectoria, evolución y desarrollo de diferentes fenómenos que dan paso a la Gestión de Proyectos, y por ende a la Gestión de Riesgos; así como analizar soluciones anteriores relacionadas con herramientas destinadas a la Gestión de Riesgos en otros proyectos productivos en el momento de proponer una solución acertada.

La presente investigación cuenta con Agradecimientos, Dedicatoria, Resumen e Introducción, además de las secciones de desarrollo fundamentadas en tres capítulos, sintetizados a continuación:

Capítulo 1: En este capítulo se describen de forma general los fundamentos teóricos relacionados con la Gestión de Riesgos, su definición y característica, así como el análisis de las principales herramientas disponibles en la actualidad, teniendo en cuenta el objetivo de llevar a cabo la elaboración de una aplicación informática. Finalmente, se definen, argumentan y valoran las principales herramientas, tecnologías, metodologías y lenguajes que se utilizan para la construcción de la solución.

Capítulo 2: En este capítulo se introduce la solución según la metodología de desarrollo seleccionada, se definen los requisitos funcionales y no funcionales, se describen los casos de uso del sistema, y se especifican los artefactos referidos a las etapas de diseño.

Capítulo 3: En este capítulo se define el diagrama de despliegue e implementación con su respectivo diagrama de componentes. Se detallan los elementos fundamentales de la arquitectura y finalmente, se diseñan y se aplican las pruebas del sistema.

## Capítulo 1: “FUNDAMENTACIÓN DE LA INVESTIGACIÓN”

### 1.1. Introducción

En el presente capítulo se profundiza en los aspectos fundamentales de la Evaluación de Riesgos en las LPS del Departamento de Geoinformática, así como las tendencias y metodologías utilizadas en la actualidad para llevar a cabo dicha Evaluación de Riesgos, finalmente se presentan posibles soluciones informáticas para la Evaluación de Riesgos y la caracterización del conjunto de metodologías, herramientas y tecnologías empleadas en la propuesta de solución.

### 1.2. Conceptos asociados al dominio del problema

#### 1.2.1. Aplicación informática

Programa informático que permite a un usuario utilizar una computadora con un fin específico. Las aplicaciones son parte del software de una computadora, y a menudo, suelen ejecutarse sobre el sistema operativo del que se dispone. Una aplicación de software suele tener un único objetivo: navegar en la web, revisar el correo, explorar el disco duro, editar textos, jugar (un juego es un tipo de aplicación), etc.

Son ejemplos de aplicaciones Internet Explorer, Outlook, Word, Excel, Winamp, etc. De acuerdo a su tipo se pueden clasificar en: aplicaciones Web, aplicaciones de escritorio, bases de datos, aplicaciones gráficas 3D (ALEGSA, 2007).

#### 1.2.2. Líneas de Productos de Software

**La Universidad Carnegie Mellon**, define una Línea de Productos de Software como *“un conjunto de software de uso intensivo de sistemas que comparten un común, gestionado conjunto de características que satisfacen las necesidades específicas de un segmento especial del mercado o misión y que se desarrollan a partir de un conjunto común de activos esenciales de una manera prescrita”* (SEI, 2011).

La definición más comúnmente aceptada de una LPS procede de (CLEMENT and NORTHRHOP, 2002) donde se definen las líneas del producto de software como *“un conjunto de sistemas de software, que comparten un conjunto común de características las cuales satisfacen las necesidades específicas de un*

*dominio o segmento particular de mercado, y que se desarrollan a partir de un sistema común de activos base de una manera preestablecida”.*

Otra definición conocida es citada por (ROSSEL *et al.*, 2009), y pertenece a (FERNÁNDEZ and HIDALGO, 2008) y plantea que *“las LPS proporcionan un enfoque sistemático para la creación de una diversidad de productos similares a bajo costo, en poco tiempo y con alta calidad. Se puede decir que constituyen un portafolio de productos de software similares, dirigidos a un dominio en particular. Generalmente usan un modelo de características que expresan los requerimientos teniendo en cuenta la variabilidad de los mismos y proporcionan una vía para el desarrollo de productos particulares a partir de los activos de software reutilizables”.*

Finalmente el autor coincide con la definición que propone (PESTANO, 2011) y que plantea que una LPS es un conjunto de sistemas de software de un entorno de negocios determinado, que tienen funcionalidades en común. El desarrollo basado en líneas de productos busca aprovechar esta parte común para desarrollar de forma eficiente y sistemática nuevos miembros de la familia de productos.

### **1.2.3. Riesgos**

Según la Guía de los Fundamentos para la Dirección de Proyectos (del inglés Project Management Body of *Knowledge*, PMBOK) el Riesgo de un proyecto es un evento o condición incierto que, si se produce, tiene un efecto positivo o negativo sobre al menos uno de los objetivos del proyecto, en tiempo, coste, alcance o calidad (PMI, 2009).

Pressman, define al riesgo como la posibilidad que un evento adverso, desgracia o contratiempo pueda manifestarse produciendo una pérdida (PRESSMAN, 2005a). Según **Robert Charette**, un riesgo afecta a los futuros conocimientos, un riesgo implica cambios y elecciones además de la incertidumbre que acompaña a la misma (CHARETTE, 1989).

Tres factores, según Pressman, afectan a las consecuencias probables de un riesgo si ocurre: su alcance, su naturaleza y cuándo ocurre.

El alcance de un riesgo combina la severidad (¿cómo de serio es el problema?) con su distribución general (¿qué proporción del proyecto se afectará y cuántos clientes se perjudicarán?).

La naturaleza del riesgo indica los problemas probables que aparecerán si ocurre. Por ejemplo, una interfaz externa mal definida para el hardware del cliente (un riesgo técnico) impedirá un diseño y pruebas tempranas y probablemente lleve a problemas de integración más adelante en el proyecto.

La temporización de un riesgo considera cuándo y por cuánto tiempo se dejará sentir el impacto (PRESSMAN, 2005a).

El autor coincide con la definición que proporciona el **PMBOK** debido al enfoque al riesgo positivo que la diferencia del resto de las definiciones.

#### **1.2.4. Gestión de riesgos**

La Gestión de Riesgos en proyectos informáticos tiene como principales objetivos aumentar la probabilidad e impacto de los eventos positivos y disminuir la probabilidad e impacto de los eventos negativos. Al decir de (LLANES, 2003), el objetivo de la gestión y análisis de riesgos en los proyectos informáticos consiste en adelantarse a aquellos imprevistos que puedan desviar al proyecto de sus metas, para ello es necesario disponer de una herramienta informática que permita cuantificar el riesgo de cada uno de los factores de riesgo que amenazan la implantación del proyecto informático con todas sus especificaciones. De acuerdo con el PMBOK, está dividida en tres partes fundamentales, la identificación, el análisis y el seguimiento y control de los riesgos (PMI, 2009).

La Identificación de los riesgos consiste en conocer e identificar qué acciones podrían suponer un problema o riesgo, y también aquellas partes que, por el contrario, puedan suponer una oportunidad de éxito en las distintas partes del proyecto. Es conveniente que los riesgos sean advertidos lo antes posible en el desarrollo del proyecto, para así poder modificar todos los aspectos necesarios para suponer mejoras. Para poder identificar los riesgos a tiempo hay que revisar toda la documentación del proyecto, así como los antecedentes históricos que puedan precederlo sacando los riesgos potenciales del proyecto.

La Evaluación de los riesgos incluye las actividades asociadas a la asignación de la probabilidad de ocurrencia, al impacto sobre los objetivos y a la priorización de los riesgos y son llevadas a cabo en los procesos de Análisis cualitativo y cuantitativo de los riesgos (FERNÁNDEZ, 2013a).

El Seguimiento y Control de los riesgos consiste en implementar planes de respuesta a los riesgos, seguir los riesgos identificados y los riesgos residuales, se identifican nuevos riesgos y se evalúa la efectividad del proceso contra riesgos a través del proyecto (PMI, 2009).

### 1.3. El proceso de desarrollo de software

Un **proceso de desarrollo de software** es la descripción de una secuencia de actividades que deben ser seguidas por un equipo de trabajadores para generar un conjunto coherente de productos. El objetivo básico del proceso es hacer predecible el trabajo que se requiere: predecir el costo, mantener un nivel de calidad, predecir el tiempo de desarrollo.

El objetivo de un proceso de desarrollo de programas es la formalización de las actividades relacionadas con el desarrollo del software de un sistema informático. No existe un proceso de desarrollo universal. Debe configurarse de acuerdo con la naturaleza del producto y de la experiencia de la empresa (DRAKE, 2008). Según lo descrito en (MUESTRAS, 2009) Pressman define el proceso de desarrollo de software como un conjunto estructurado de actividades y resultados requeridos para desarrollar un sistema de software a través de las siguientes actividades:

- **Especificación:** establecer los requisitos y restricciones del sistema.
- **Diseño:** Producir un modelo en papel del sistema.
- **Implementación:** construcción del sistema de software.
- **Validación:** verificar (por ejemplo mediante pruebas) que el sistema cumple con las especificaciones requeridas.
- **Instalación:** entregar el sistema al usuario y asegurar su funcionamiento.
- **Evolución y mantenimiento:** cambiar/adaptar el software según las demandas; reparar fallos en el sistema cuando sean descubiertos.

Las actividades varían dependiendo de la organización y del tipo de sistema a desarrollar. Debe estar explícitamente modelado si va a ser bien administrado.

### 1.4. Caracterización del objeto de estudio

#### 1.4.1. El Proceso de Evaluación de Riesgos

De acuerdo con MAGERIT, en el proceso de Evaluación de Riesgos se identifican las relaciones entre los activos a tratar y la valoración que merecen; se identifican las amenazas significativas y se valoran en términos de frecuencia de ocurrencia y degradación que causan sobre el valor del activo afectado. Según esta propuesta se entiende como activo a cualquier recurso de interés para la organización y que por lo tanto representa el foco principal de atención y protección por parte del equipo de riesgos (MAP, 2006).

Según CMMI, la Evaluación de Riesgos es necesaria para asignar la importancia relativa a cada riesgo identificado, y se usa en la determinación de cuándo se requiere la atención apropiada de la gerencia, se consideran la probabilidad de ocurrencia del riesgo, la gravedad o el impacto ocasionado y los umbrales para disparar las acciones de gestión (CHRISSIS *et al.*, 2009).

En PMBOK la evaluación de los riesgos es referida con el término análisis y se divide en Análisis Cuantitativo y Cualitativo. Se desarrolla a partir de técnicas y herramientas.

El *Análisis cualitativo* de los riesgos evalúa la prioridad de los riesgos identificados usando la probabilidad de ocurrencia, el impacto correspondiente sobre los objetivos del proyecto si los riesgos efectivamente ocurren, así como otros factores como el plazo y la tolerancia al riesgo de las restricciones u objetivos del proyecto como costo, cronograma, alcance y calidad.

El *Análisis cuantitativo* se centra en la cuantificación y priorización de los riesgos de forma objetiva. Se pueden resaltar sus principales funciones: determinar la probabilidad de realizar un objetivo específico del proyecto, cuantificar el riesgo del proyecto y determinar el tamaño del costo.

Las técnicas y herramientas empleadas para su evaluación son la evaluación de la probabilidad y el impacto, la utilización de la matriz del mismo nombre y la evaluación de la urgencia del riesgo (PMI, 2009).

La propuesta de PMBOK solamente utiliza los conceptos de probabilidad e impacto a través de la matriz para priorizar los riesgos identificados, no incluye la definición de umbral definida en CMMI aunque propone en la evaluación de la urgencia de los riesgos incluir en la priorización los síntomas y señales de advertencia para dar respuesta a los riesgos.

Según (FERNÁNDEZ, 2013a), en el proceso de Evaluación de Riesgos actual no se considera la influencia o dependencia de un activo sobre otro en el cálculo del impacto del riesgo sobre los activos, la

reestimación se desarrolla a partir de una nueva evaluación de su impacto sobre los activos y su probabilidad de ocurrencia, y no se provechan los resultados históricos que permitan desarrollar la evaluación considerando la evolución real del riesgo.

Los autores agregan, además de lo anterior, que en los parámetros para obtener la exposición al riesgo con su consecuente priorización se observa una utilización recurrente de la probabilidad y el impacto de ellos únicamente.

#### **1.4.2. La Evaluación de Riesgos en LPS**

En la conceptualización de la Evaluación de Riesgos en LPS se revisaron los enfoques propuestos por las LPS siguientes: Proceso Ingenieril Evolutivo, ESPELEP, TWIN, CAFÉ, Modelo propuesto por la UCI, WATCH, Modelo del SEI.

ESPELEP, CAFÉ y WATCH no contemplan actividades relacionadas con la gestión o evaluación de los riesgos explícita ni implícitamente en sus definiciones, TWIN por su parte, sí enuncia las actividades referidas a los riesgos aunque no los detalla, ni proporciona un mecanismo o una guía para desarrollarlas (SAMETINGER, 2007). El Modelo propuesto por el SEI es llamado Paradigma de Gestión de Riesgos Continuo y en él los riesgos se identifican por primera vez y luego se analizan para determinar su probabilidad de ocurrencia y el impacto (exposición al riesgo) sobre la organización, los planes de mitigación se desarrollan en las áreas de riesgo más importantes, y las decisiones de control (cierre, re-planificación, u otras) están debidamente documentadas.

Este enfoque de riesgos para LPS requiere una íntima asociación entre los desarrolladores de activos de software y los desarrolladores de productos, de ahí que esta propuesta introduzca el concepto de Equipo de Gestión de Riesgos que afecte a toda la línea y “escuche” todas las partes (SEI, 2011).

En la propuesta del SEI la evaluación de los riesgos no queda definida explícitamente aún cuando se evidencia una tendencia a la utilización de la probabilidad y el impacto para su cálculo. Aún siendo una propuesta concreta y sólida no contiene las actividades de Evaluación de Riesgos formalmente descritas, además, no incluye las evaluaciones históricas ni considera el impacto de los activos y su importancia dentro de la LPS (FERNÁNDEZ, 2013a).

Finalmente en el modelo propuesto por el Laboratorio de Gestión de Proyectos de la UCI las actividades referidas a los riesgos sí son explícitamente detalladas, y se identifican acciones para la planificación, identificación análisis, seguimiento y control de los riesgos en cada una de las fases propuestas. Propone la definición de reportes en la toma de decisiones a nivel de proyecto y de criticidad de los riesgos, las actividades referidas a la evaluación de los riesgos se desarrollan de acuerdo a PMBOK y sin un enfoque a LPS.

En general, según (FERNÁNDEZ, 2013a), la evaluación de los riesgos no comprende un enfoque claro hacia activos generales, que permita evaluarlos según su interés e importancia dentro del proyecto en cuestión; tampoco incluyen la posibilidad de lograr una priorización real en función del impacto del riesgo sobre un activo determinado y la importancia de este activo para la LPS que permitan situar al riesgo en el lugar que le corresponde con mayor certeza.

Además, cuando se evalúa un riesgo, al momento de la re-estimación, ninguna de las propuestas caracterizadas aprovecha la información histórica referida a las evaluaciones anteriores como un posible punto de partida para la nueva evaluación, se limitan a calcularlas nuevamente según los mismos supuestos de la primera vez.

### **1.4.3. La Evaluación de Riesgos en las LPS del Departamento de Geoinformática**

En las LPS del Departamento de Geoinformática las actividades asociadas a la Gestión de riesgos se desarrollan de acuerdo al modelo realizado por (FERNÁNDEZ, 2013a) para obtener el Máster en Gestión de Proyectos Informáticos.

De acuerdo a su autor, la propuesta se basa en:

1. La propuesta realizada por el Ministerio de Administraciones Públicas de Madrid denominada MAGERIT (CRESPO *et al.*, 2006).
2. La revisión del área de procesos de Gestión de Riesgos que se define en CMMI (CHRISIS *et al.*, 2009).
3. La aplicación del área del conocimiento referida a la GR del PMBOK y propuesta por el PMI (PMI, 2009).

El modelo es el producto de la síntesis y reestructuración de estas tres propuestas, enfocadas a la reutilización de componentes y al modelo de producción basado en LPS, utiliza de manera particular las actividades propuestas por PMBOK y contiene las actividades propias de la evaluación de los riesgos como resultado fundamental de la investigación debido al interés de dirigirla hacia el trabajo con los activos de la LPS.

Es importante destacar que este modelo propone desarrollar las actividades de caracterización de los riesgos (Identificación de riesgos) y las actividades de definición y desarrollo de las acciones de mitigación y contingencia (Seguimiento y Control de los riesgos) según especifica el PMBOK. Ver Figura 1.



Fig. 1: Ubicación de la propuesta (FERNÁNDEZ, 2013a).

La propuesta está dividida en cuatro actividades fundamentales, cada una de ellas con entradas, salidas e involucrados así como las acciones que son necesarias para que sean ejecutadas sin dificultades. Ver Tabla 1.

Tabla 1: Involucrados, entrada, acciones y salida de la propuesta (FERNÁNDEZ, 2013a).

	Involucrados	Entrada	Acciones	Salida
<b>Identificación de los Riesgos</b>	El resultado fundamental de esta Fase lo constituye el listado de los riesgos asociados al proyecto. Se propone que se registren en el documento denominado “Análisis de los riesgos del proyecto”.			
<b>Evaluación de los Riesgos</b>	EGR Analista de riesgos	Análisis de riesgos	Determinación de la facilidad de detección de cada riesgo	Análisis de riesgos (actualizado)

	EGR Líder de la LPS	Análisis de riesgos	Determinación de la probabilidad de ocurrencia de cada riesgo	Análisis de riesgos (actualizado)
	EGR Analista de riesgos	Análisis de riesgos	↓ Estimación del impacto de los activos del proyecto	Análisis de riesgos (actualizado)
	Analista de riesgos	Análisis de riesgos	↓ Estimación del impacto de los riesgos sobre los activos del proyecto	Análisis de riesgos (actualizado)
	Analista de riesgos	Análisis de riesgos	↓ Obtención de la exposición a cada riesgo	Análisis de riesgos (actualizado)
	Analista de riesgos Líder de la LPS	Análisis de riesgos	↓ Determinación de los umbrales de atención a los riesgos	Análisis de riesgos (actualizado)
<b>Seguimiento y Control de los Riesgos</b>	El resultado fundamental de esta Fase lo constituye la ejecución de las acciones de mitigación o contingencia según proceda, para desarrollarlas es necesario el listado priorizado de los riesgos del proyecto obtenido como resultado fundamental de la Fase de Evaluación de los Riesgos.			

Las cuatro actividades fundamentales son:

**Actividad 1:** Estimación de la facilidad de detección del riesgo.

**Actividad 2:** Estimación de la probabilidad de ocurrencia de los riesgos.

**Actividad 3:** Estimación del impacto de los riesgos sobre los activos.

**Actividad 4:** Priorización de los riesgos.

Se propone un Equipo de Gestión de Riesgos (EGR) dirigido por un Analista de riesgos que de conjunto se encarguen del tratamiento de todos los riesgos en la LPS. Se considera como entrada de la propuesta de evaluación a la documentación que se obtuvo en la fase -etapa, momento- de identificación de los riesgos y que es necesaria para comenzar las actividades de evaluación de los riesgos. Asimismo, se considera una salida a la documentación obtenida en su ejecución y que sirve de punto de partida a la fase –etapa, momento- de seguimiento y control de los riesgos (FERNÁNDEZ, 2013a).

Una acción debe ser entendida como la tarea concreta a realizar para finalizar exitosamente una actividad, cada una cuenta con su descripción, las herramientas y las técnicas que se utilizan para desarrollarla. Para obtener el éxito en la ejecución del método se propone que la información que se genere o manipule se registre en un solo documento denominado Análisis de los Riesgos. Este documento contendrá entre sus principales acápites lo relacionado con la facilidad de detección de cada riesgo, su probabilidad de ocurrencia, el impacto sobre los activos y la priorización final de cada uno, así como los valores posteriores de re-evaluación (FERNÁNDEZ, 2013a).

### **1.5. Soluciones existentes**

#### Sistema de Razonamiento Basado en Casos para la identificación de riesgos de software

Sistema desarrollado en el Centro de Informatización de la Seguridad Ciudadana (ISEC) de la Facultad 2 de la UCI dedicado al desarrollo de aplicaciones informáticas para diversos órganos e instituciones que brindan seguridad a los ciudadanos. El sistema brinda la posibilidad de que a partir de un conjunto de características relevantes del proyecto que le son introducidas, se logre obtener de forma automática los riesgos potenciales a incidir a lo largo del ciclo de desarrollo, así como una posible mitigación o aprovechamiento de los mismos. Esto permite tener una visión adelantada, agilizando el proceso de toma de decisiones (RIQUELME and MORALES, 2010).

La solución desarrollada constituye una herramienta capaz de ayudar y apoyar el proceso de toma de decisiones, pues a partir de ciertos datos de entrada correspondientes a características de los proyectos se pueden predecir posibles riesgos a incidir así como algún tipo de tratamiento que se le puede dar a estos.

Para ello se definieron los procesos fundamentales relacionados con la Gestión de Riesgos de software y tras el análisis detallado de los métodos que abordan el tema, se seleccionó MOGERI como la metodología adecuada de acuerdo con las características del centro de desarrollo de software ISEC.

### XEDRO GESPRO

Suite de Gestión de Proyectos, desarrollada por el Laboratorio de Gestión de Proyectos de la UCI que se presenta como un modelo de negocios basado en servicios donde se combina el uso de una solución informática para la gestión de proyectos y un sistema de formación especializada en gestión de proyectos. Esta combinación posibilita no sólo la informatización de la gestión de proyectos en las organizaciones, sino también la mejora continua de sus procesos de planificación, seguimiento y control.

La solución informática es desarrollada bajo un ecosistema de software basado en tecnologías libres, y permite entre otras: La planificación del alcance y el tiempo, la gestión de recursos humanos y sus competencias, la Gestión de Riesgos, así como la financiera de los proyectos. Gestión logística y gestión de recursos compartidos. La gestión de los riesgos se implementa a partir de la propuesta de (PESTANO, 2011) descrita en el apartado anterior. (ECURED, 2011).

### Herramienta para la Gestión de Riesgos en Proyectos de Software

Esta herramienta fue desarrollada en la Facultad de Informática del Instituto de Investigación en Informática de La Plata en Argentina. La herramienta permite gestionar el conocimiento generado para enriquecer el análisis de proyectos subsiguientes. Se gestionan los riesgos de un proyecto desde la etapa de licitación de requerimientos y durante todo el ciclo de vida. De esta forma se puede administrar de manera proactiva cada acontecimiento presentado durante el análisis, diseño, codificación, prueba y mantenimiento del software, y además es posible disponer de suficiente material para generar planes alternativos de contingencia.

### Herramienta de Evaluación de Riesgo-CRAMM

CRAMM es la metodología de análisis de riesgos desarrollado por el Centro de Informática y la Agencia Nacional de Telecomunicaciones (CCTA) del gobierno del Reino Unido.

El significado del acrónimo proviene de CCTA *Risk Analysis and Management Method*. Su versión inicial data de 1987 y la versión vigente es la 5.2.

CRAMM incluye una amplia gama de herramientas de evaluación de riesgo que son totalmente compatibles con la ISO- 27001 y se ocupa de tareas como:

- Activos de modelado de dependencia.
- Evaluación de impacto empresarial.
- Identificación y evaluación de amenazas y vulnerabilidades.
- Evaluar los niveles de riesgo.
- La identificación de los controles necesarios y justificados sobre la base de la evaluación del riesgo.
- Un enfoque flexible para la Evaluación de Riesgos.

CRAMM es aplicable a todo tipo de sistemas y redes de información y se puede aplicar en todas las etapas del ciclo de vida del sistema de información, desde la planificación y viabilidad, a través del desarrollo e implementación del mismo.

CRAMM, toma en cuenta las medidas de los riesgos determinados durante la etapa anterior y los compara con el nivel de seguridad (un nivel de umbral asociado con cada contramedida) con el fin de identificar si los riesgos son suficientemente grandes para justificar la instalación de una determinada contramedida. CRAMM ofrece una serie de servicios de ayuda, incluyendo marcha atrás. Las funciones de priorización y presentación de informes para ayudar en la implementación de las contramedidas y la gestión activa de los riesgos identificados.

Las herramientas anteriores, si bien se relacionan con la investigación en curso, presentan algunas limitaciones que les impiden ser consideradas soluciones al problema planteado.

1. Ninguna de las herramientas presenta un enfoque hacia LPS o activos de software, que permita incluir la gestión y Evaluación de Riesgos de varios proyectos a la vez.
2. Aún cuando las propuestas de la UCI son desarrolladas sobre software libre, las tecnologías empleadas no se adecuan con la arquitectura de las LPS de Geoinformática.

3. La herramienta XEDRO GESPRO solamente realiza la identificación de los riesgos dentro del proceso de Gestión de Riesgos.
4. No existe la posibilidad de descargar las herramientas internacionales que permita evaluar su utilización práctica.

## 1.6. Metodologías, herramientas y tecnologías empleadas

### 1.6.1. Metodología de desarrollo de software

Una metodología no es más que un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software (PATÓN, 2007), (PÉREZ *et al.*, 2008), por su parte especifican que la Metodología de Desarrollo de Software surge ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto (software). Se clasifican en **metodologías pesadas** y **metodologías ágiles**, siendo muy populares la Programación Extrema (XP), Scrum y el Proceso Unificado de Desarrollo (RUP, por sus siglas en inglés).

#### Metodologías Pesadas

Son las más tradicionales, se centran en la definición detallada de los procesos y tareas a realizar, herramientas a utilizar, y requiere una extensa documentación, ya que pretenden prever todo de antemano. Este tipo de metodologías son más eficaces y necesarias cuanto mayor es el proyecto que se pretende realizar respecto a tiempo y recursos que son necesarios emplear, donde una gran organización es requerida.

No existen dos proyectos de desarrollo de software que sean iguales, pues cada uno tiene prioridades, requerimientos, y tecnologías muy diferentes. Sin embargo, en todos los proyectos, se debe minimizar el riesgo, garantizar la predictibilidad de los resultados y entregar software de calidad (PÉREZ *et al.*, 2008).

Rational Unified Process (RUP), es una plataforma flexible de procesos de desarrollo de software que ayuda brindando guías consistentes y personalizadas de procesos para todo el equipo de proyecto. Describe cómo utilizar de forma efectiva reglas de negocio y procedimientos comerciales probados en el desarrollo de software para equipos de desarrollo de software, conocidos como “mejores prácticas”.

Captura varias de las mejores prácticas en el desarrollo moderno de software en una forma que es aplicable para un amplio rango de proyectos y organizaciones. Provee a cada miembro del equipo fácil acceso a una base de conocimiento con guías, plantillas y herramientas para todas las actividades críticas de desarrollo (EUMET, 2011). Como una plataforma de procesos que abarca todas las prácticas de la industria, RUP permite seleccionar fácilmente el conjunto de componentes de proceso que se ajustan a las necesidades específicas del proyecto (ACUÑA, 2009). Otras metodologías trabajan de manera similar a RUP en el tratamiento de los casos de uso como es ICONIX pero no con la profundidad de RUP, además no puede ser usado para proyectos grandes y necesita información rápida y puntual de los requisitos, el diseño y las estimaciones.

MSF es otra metodología tradicional pero está basado en desarrollo con tecnología Microsoft lo que limita las opciones del cliente en lo que se refiere a herramientas de desarrollo. Esta metodología hace un análisis de riesgos demasiado exhaustivo y solicita demasiada documentación, lo que puede frenar el avance del proyecto y afectar el tiempo de entrega del producto.

## Metodologías Ágiles

Proponen como principales ideas:

- Valorar al individuo y las iteraciones más que a las herramientas o los procesos utilizados.
- Enfatizar más en un producto software que funcione, que escribir mucha documentación.
- El cliente está en todo momento colaborando en el proyecto.
- Es más importante la capacidad de respuesta ante un cambio que el seguimiento estricto de un plan.

### Metodología seleccionada: RUP Ágil

El **Agile UP (AUP)** es una versión simplificada de RUP. Describe un enfoque simple y fácil de entender para el desarrollo de software usando técnicas y conceptos que aún se mantienen vigentes en RUP. La descripción es simple, con vínculos de detalles en la web, si es necesario abundar en el tema. Los enfoques aplican técnicas ágiles incluidas en el Desarrollo Dirigido por Pruebas (TDD), Desarrollo Dirigido por Modelado Ágil (AMDD), administración de cambios ágil, y refactorización de bases de datos para mejorar la productividad. Muchas organizaciones dudan de XP por parecer muy liviana.

XP no muestra explícitamente cómo crear algunos de los productos de trabajo que la administración desea ver. Por el otro lado del espectro está RUP, en el cual la administración parece ser muy adorable, pero los desarrolladores dudan del gran número de entregables. Esto es desafortunado porque RUP tiene mucho que ofrecer, y se puede reducir a algo muy útil (lo cual es exactamente lo que IBM Rational le recomienda hacer). AUP se encuentra entre los dos, adopta muchas de las técnicas ágiles de XP y otros procesos ágiles que aún mantienen la formalidad de RUP (DTIC, 2009). El ciclo de vida para un proyecto de software que propone AUP se muestra en la figura siguiente:

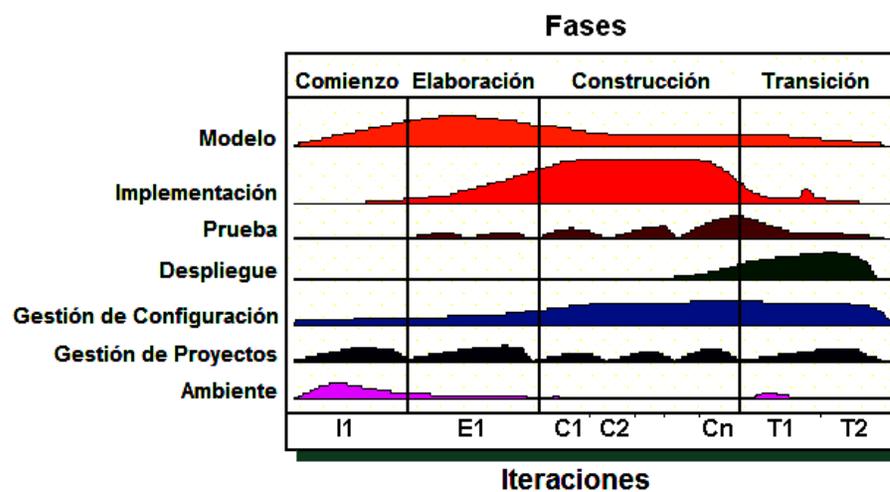


Fig. 2: Ciclo de vida de AUP. Tomado de (Informática, 2006).

Asimismo, se proponen las siguientes actividades fundamentales:

- Comienzo o iniciación: Definir alcance del proyecto, estimar costes y programación, definir riesgos, determinar viabilidad del proyecto, preparar entorno del proyecto.
- Elaboración: Identificar arquitectura, validar arquitectura, desarrollar entorno del proyecto, determinar el equipo.
- Construcción: Modelar, construir y probar el sistema, desarrollar documentación de soporte.
- Transición: Pruebas del sistema, pruebas de usuario, integración, despliegue.

### Ventajas de AUP

- El personal sabe lo que está haciendo: no obliga a conocer detalles.
- Simplicidad: apuntes concisos.

- Agilidad: procesos simplificados del RUP.
- Centrarse en actividades de alto valor: esenciales para el desarrollo.
- Herramientas independientes: a disposición del usuario (FLORES, 2013).

### **1.6.2. Lenguaje de modelado como soporte a la metodología**

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el Object Management Group (OMG). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables (CONML, 2013).

Se resalta que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional), pero no especifica en sí mismo qué metodología o proceso usar (ZAMITIZ, 2011).

Se utilizó UML como lenguaje para el modelado en la solución propuesta por ser sencillo y simple su uso. Además da la posibilidad de modelar todos los componentes del desarrollo de la aplicación propuesta. Es flexible a los cambios que se puedan realizar en el modelado de algún componente en específico.

### **1.6.3. Herramienta para el modelado**

Para el trabajo con software es necesario utilizar herramientas de modelado UML en computadora, también llamadas herramientas de Ingeniería de Software asistida por computadoras (CASE por sus siglas en inglés). Incluso algunas de estas herramientas pueden generar, a partir de modelos UML, el esqueleto del sistema en el lenguaje de programación que se requiera (SYSTEMS, 2007).

Visual Paradigm como herramientas CASE

Visual Paradigm for UML es una herramienta CASE que soporta el modelado mediante UML y proporciona asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un Software. Algunas de sus ventajas son:

- Dibujo. Facilita el modelado de UML, ya que proporciona herramientas específicas para ello. Esto también permite la estandarización de la documentación, ya que la misma se ajusta al estándar soportado por la herramienta.
- Corrección sintáctica. Controla que el modelado con UML sea correcto.
- Coherencia entre diagramas. Al disponer de un repositorio común, es posible visualizar el mismo elemento en varios diagramas, evitando duplicidades.
- Integración con otras aplicaciones. Permite integrarse con otras aplicaciones, como herramientas ofimáticas, lo cual aumenta la productividad.
- Trabajo multiusuario. Proporciona herramientas de compartición de trabajo.
- Reutilización. Facilita la reutilización, ya que disponemos de una herramienta centralizada donde se encuentran los modelos utilizados para otros proyectos.
- Generación de código. Permite generar código de forma automática, reduciendo los tiempos de desarrollo y evitando errores en la codificación del software.
- Generación de informes. Permite generar diversos informes a partir de la información introducida en la herramienta (GROUP, 2009).

Se seleccionó como herramienta CASE para el modelado Visual Paradigm por la extensa documentación que presenta y el dominio que tiene el equipo de desarrollo sobre la misma. La universidad cuenta con una licencia para su uso, también posee una buena capacidad para graficar los componentes fundamentales, utiliza el lenguaje UML y brinda una serie de facilidades en la generación automática de código ahorrando tiempo a los desarrolladores.

#### **1.6.4. Lenguajes de programación**

Un lenguaje de programación es un lenguaje artificial que puede ser usado para controlar el comportamiento de una máquina, especialmente una computadora. Estos se componen de un conjunto de reglas sintácticas y semánticas que permiten expresar instrucciones que luego serán interpretadas (PROGRAMACIÓN, 2009).

Debe distinguirse de “lenguaje informático”, que es una definición más amplia, puesto estos incluyen otros lenguajes como son el HTML o PDF que dan formato a un texto y no es programación en sí misma (ALEGSAONLINE.COM, 2006).

### **Del lado del cliente: HTML, CSS, JavaScript**

El lenguaje **HTML** indica al navegador dónde colocar cada texto, cada imagen o cada video y la forma que tendrán estos al ser colocados en la página. El lenguaje consta de etiquetas que se utilizan para definir la forma o el estilo que se quiere aplicar a una determinada Web. Las Hojas de estilo en Cascada (del inglés Cascading Style Sheets, **CSS**) son una tecnología que permite crear páginas web de una manera más exacta. Producto de las CSS, un usuario es mucho más administrador de los resultados finales de una página web, pudiendo desarrollar acciones que no son posibles solamente con HTML: incluir márgenes, tipos de letra, fondos y colores.

**JavaScript** es un lenguaje de programación utilizado para crear programas encargados de realizar acciones dentro del ámbito de una página web. Se define como lenguaje del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Su uso se basa fundamentalmente en la creación de efectos especiales en las páginas y la definición de interactividades con el usuario. Las sentencias escritas en javascript se encapsulan entre las etiquetas `<script>` y `</script>` (TORRES, 2009). No se puede desarrollar un programa con JavaScript que se ejecute fuera de un Navegador, aunque en este momento comienza a expandirse a otras áreas como la programación en el servidor con Node.js Es un lenguaje interpretado que se embebe en una página web HTML (JAVASCRIPTYA, 2012).

Se utilizaron los lenguajes de programación HTML, CSS y JavaScript en la solución propuesta con un objetivo específico cada uno. HTML se utilizó para la elaboración de las páginas web que conforman la vista de la aplicación web, definiendo una estructura básica en las mismas. Este lenguaje brinda la posibilidad de ser creado y editado con cualquier editor de texto básico. HTML es compatible con todo sistema operativo.

CSS por su parte se utilizó para el estilo de las páginas web de la solución propuesta. Este lenguaje agrega muchas más unidades de medidas que las que brindaba HTML para definir atributos. Por otra parte nos da la posibilidad de dar estilo una única vez a varios elementos.

JavaScript fue empleado para realizar acciones determinadas dentro de las páginas web, siendo un lenguaje que presenta una buena documentación y un gran dominio por parte del equipo de desarrollo. Es uno de los lenguajes actualmente más usados, además permite el manejo de objetos dentro de la página web y poder definir eventos sobre este objeto.

### **Del lado del servidor: PHP**

PHP (acrónimo recursivo de **Hypertext Preprocessor**), es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

Puede usarse en todos los principales sistemas operativos, incluyendo Linux, muchas variantes de Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS y probablemente otros más. PHP admite la mayoría de servidores web de hoy día, incluyendo Apache y IIS. De modo que con PHP se tiene la libertad de elegir el sistema operativo y el servidor web. Además, se tiene la posibilidad de utilizar programación por procedimientos o programación orientada a objetos (POO), o una mezcla de ambas.

Lo que distingue a PHP de algo como Javascript del lado del cliente es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibe el resultado de ejecutar el script, aunque no se sabría el código subyacente que contenía (GROUP, 2009).

Se selecciona el lenguaje PHP en la solución propuesta ya que cuenta con una amplia documentación y el equipo de desarrollo posee cierto dominio sobre el mismo. Es muy adecuado en el desarrollo de la solución siendo esta una aplicación web de contenido dinámico. Este lenguaje es libre brindando la posibilidad de acceso de todos. Presenta una capacidad de conexión con el sistema gestor de base de datos PostgreSQL, además su la programación en este lenguaje es segura y confiable.

#### **1.6.5. Framework de desarrollo**

Un framework de desarrollo tiene como objetivo simplificar el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes y proporcionar estructura al código fuente, obligando al desarrollador a crear código más legible y más fácil de mantener.

Además facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas.

### Symfony 2.3 como framework de desarrollo seleccionado

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Inicialmente, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases para reducir el tiempo de desarrollo. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Symfony2 ha sido ideado para exprimir al límite todas las nuevas características de PHP 5.3 y por eso es uno de los frameworks PHP con mejor rendimiento. Es además el framework que más ideas incorpora del resto de frameworks, incluso de aquellos que no están programados con PHP. Su arquitectura interna está completamente desacoplada, lo que permite reemplazar o eliminar fácilmente aquellas partes que no encajan en tu proyecto (EGUILUZ, 2011). Symfony2.3 se publicó en junio de 2013, casi dos años después que Symfony2.0. Esta nueva versión mejora todos los aspectos de la versión original, al tiempo que mantiene una alta compatibilidad.

Se seleccionó Symfony2.3 como herramienta para el desarrollo de la solución porque brinda muchas facilidades para el trabajo con el lenguaje PHP y nos brinda un motor de plantilla web muy potente, buscando organización dentro de la vista de la aplicación web. Symfony2.3 también nos brinda una completa documentación y su arquitectura interna permite estructurar la solución siguiendo el patrón arquitectónico Modelo-Vista-Controlador.

#### **1.6.6. Sistema Gestor de Bases de Datos**

Un sistema gestor de base de datos (SGBD) es una agrupación de programas que sirven para definir, construir y manipular una base de datos. Según (RIQUELME and MORALES, 2010) un Sistema Gestor de Bases de Datos no es más que un conjunto de programas, procedimientos y lenguajes que proporcionan las herramientas necesarias para trabajar con una base de datos. Incorpora una serie de funciones que permite definir los registros, sus campos, sus relaciones, insertar, suprimir, modificar y consultar los datos. Se clasifican en:

- **Relacional (SGBDR):** representa a la base de datos como una colección de tablas. Estas bases de datos suelen utilizar SQL como lenguaje de consultas de alto nivel.
- **Orientado a objetos:** define a la base de datos en términos de objetos, sus propiedades y sus operaciones. Todos los objetos que tienen la misma estructura y comportamiento pertenecen a una clase y las clases se organizan en jerarquías.
- **Objeto-relacional o relacional extendido:** son los sistemas relacionales con características de los orientados a objetos.
- **Jerárquico:** representa los datos en estructuras jerárquicas como árboles (ALEGSA.COM, 1998).

Algunos ejemplos de sistemas gestores de bases de datos muy utilizados en la actualidad son MySQL, SQL Server, PostgreSQL y Oracle.

#### El SGBD seleccionado: PostgreSQL

**PostgreSQL** es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema.

Entre sus principales características están:

- Es una base de datos 100% ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad).
- Copias de seguridad en caliente (Online/hot backups).
- Múltiples métodos de autenticación.
- Completa documentación.
- Disponible para Linux, UNIX en todas sus variantes y Windows 32/64bit.
- Acceso encriptado vía SSL (FERNÁNDEZ, 2013b).

Se seleccionó en la solución propuesta el sistema de gestión de bases de datos PostgreSQL ya que utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. Debido a la información que se maneja en la solución propuesta es necesaria una alta seguridad de la misma, por su parte PostgreSQL brinda un extenso conjunto de protocolos de seguridad y opciones de configuración para el control de acceso a los datos almacenados.

### 1.7. Conclusiones

La Evaluación de Riesgos en la actualidad, y específicamente en el Departamento de Geoinformática, no se desarrolla dirigida hacia activos, ni incluye la posibilidad de lograr una priorización real del impacto del riesgo sobre un activo determinado y la importancia de este activo para cualquier LPS que permitan situar al riesgo en el lugar que le corresponde con mayor precisión.

La propuesta de solución se desarrolla de acuerdo a la propuesta de (FERNÁNDEZ, 2013a) debido a que está dirigida con un enfoque claro hacia activos de software y fue desarrollada para LPS específicamente. Las herramientas analizadas no constituyen solución del problema que se propone en esta investigación debido a varias limitaciones que impiden su utilización, por otra parte, la selección de las herramientas y tecnologías satisface las necesidades en el desarrollo al mismo tiempo que impulsa los principios de soberanía tecnológica defendidos por la Universidad.

## Capítulo 2: “Descripción de la solución propuesta”

### 2.1. Introducción

En un nivel técnico, la ingeniería del software empieza con una serie de tareas de modelado que llevan a una especificación completa de los requisitos y a una representación del diseño general del software a construir. (PRESSMAN, 2005b). En el capítulo que se presenta se describe la solución a partir de sus principales elementos ingenieriles, modelos y descripción arquitectónica general.

### 2.2. Modelo de dominio

Un Modelo de Dominio es una representación visual de clases conceptuales o de objetos reales en un dominio de interés. Consiste en un conjunto de clases, sin definición de operaciones. El modelo de dominio proporciona una perspectiva conceptual en cuanto a objetos del dominio o clases conceptuales y las asociaciones entre estas clases conceptuales (LARMAN, 1999).

#### 2.2.1. Diagrama de clases del dominio

En la Figura 3 se muestra el diagrama de clases del dominio propuesto el cual presenta los siguientes conceptos en forma de clases.

- Departamento de Geoinformática: representa el máximo nivel, o sea, la máxima dirección del departamento, que contiene tres Líneas de Productos de Software.
- LPS: (Líneas de Productos de Software). Representa las dos LPS del Departamento de Geoinformática, a saber, Aplicativos SIG y SIG-Desktop.
- Proyectos: representa los distintos proyectos de desarrollo en que trabaja una LPS.
- Activos: representa los activos que existen en una LPS, cualquier recurso de interés para la organización, estos activos pueden ser afectados por uno, muchos o ningún riesgo.
- Riesgos: representa la posibilidad de que un evento ocurra, afectando a uno o más activos.
- EGR: representa los trabajadores encargados de la Gestión de Riesgos en una LPS, denominado Equipo de Gestión de Riesgos.
- Gestión de Riesgos: representa el proceso de identificación, evaluación, y seguimiento y control de riesgos.

- Identificación: representa la actividad referida a la identificación de los riesgos.
- Evaluación: representa la actividad referida a la evaluación de los riesgos identificados.
- Seguimiento y Control: representa la actividad referida al seguimiento y control de los riesgos evaluados.
- Líder: Jefe de Proyecto. Representa al líder de una LPS.
- Analista: representa al analista de riesgos de una LPS.
- Otros miembros: representa a los demás miembros del EGR de una LPS.

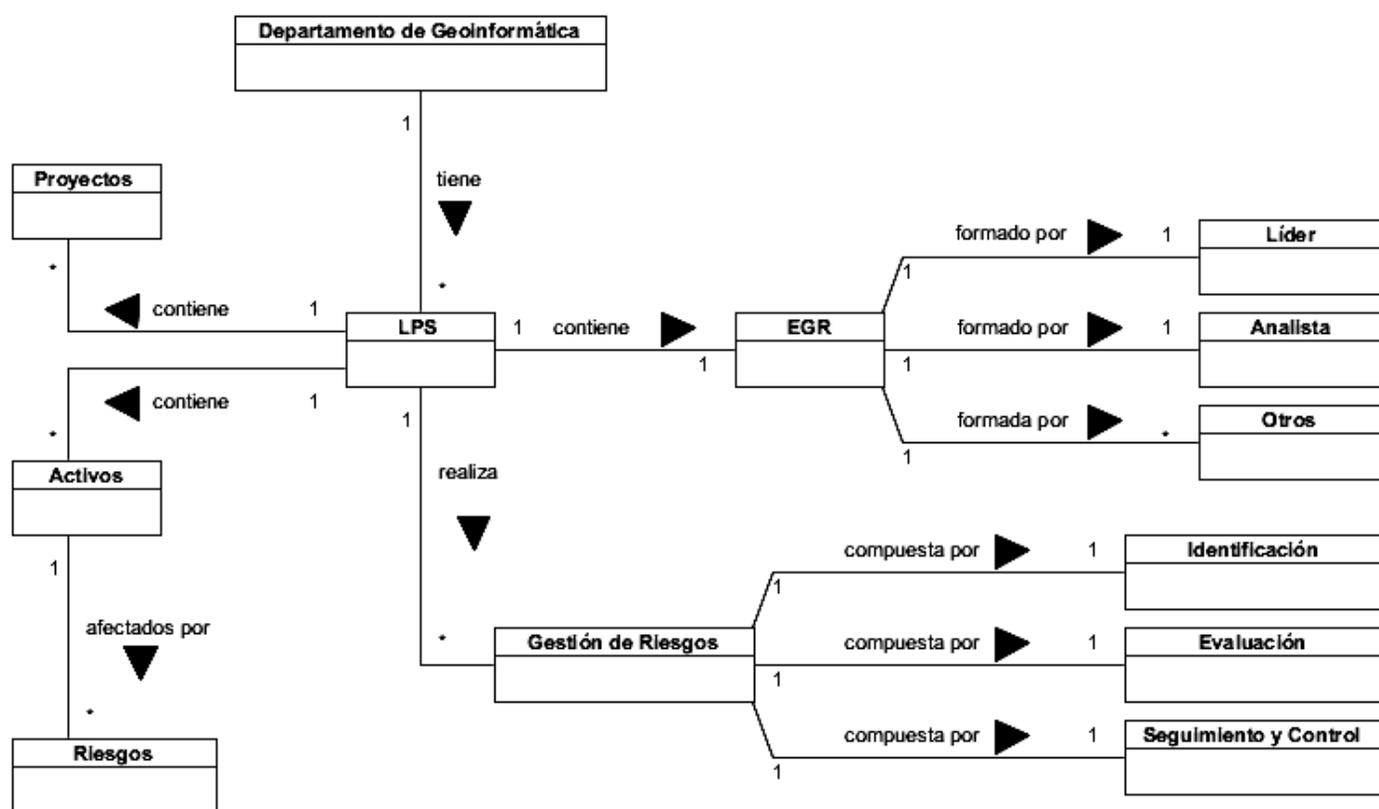


Fig. 3: Diagrama de clases del dominio.

### 2.3. Requisitos

Según la Real Academia de la Lengua Española un requisito es la circunstancia o condición necesaria para algo.

En aplicaciones de software y hardware, los requerimientos de software son las características (requisitos no funcionales) y las capacidades (requisitos funcionales) que el software debe tener y cumplir para poder soportar y/o ejecutar una aplicación o un dispositivo específico (PRESSMAN, 2005b).

### **2.3.1. Requisitos Funcionales**

**El sistema debe ser capaz de:**

- RF1. Autenticar usuario.
- RF2. Cerrar Sesión.
- RF3. Agregar un usuario al sistema.
- RF4. Modificar un usuario del sistema.
- RF5. Eliminar un usuario del sistema.
- RF6. Listar los usuarios del sistema.
- RF7. Agregar una LPS al sistema.
- RF8. Modificar una LPS del sistema.
- RF9. Eliminar una LPS del sistema.
- RF10. Listar las LPS del sistema.
- RF11. Agregar un proyecto a una LPS del sistema.
- RF12. Modificar un proyecto de una LPS del sistema.
- RF13. Eliminar un proyecto de una LPS del sistema.
- RF14. Listar los proyectos de una LPS del sistema.
- RF15. Agregar un activo a una LPS del sistema.
- RF16. Modificar un activo de una LPS del sistema.
- RF17. Eliminar un activo de una LPS del sistema.
- RF18. Listar los activos de una LPS del sistema.
- RF19. Agregar un riesgo en un proyecto de una LPS del sistema.
- RF20. Modificar un riesgo en un proyecto de una LPS del sistema.
- RF21. Eliminar un riesgo en un proyecto de una LPS del sistema.
- RF22. Listar los riesgos de un proyecto de una LPS del sistema.
- RF23. Asociar un riesgo a un activo de esa LPS.

- RF24. Determinar la facilidad de detección (**Fd**) de cada riesgo en un proyecto de una LPS del sistema.
- RF25. Determinar de la probabilidad de ocurrencia (**Po**) de cada riesgo en un proyecto de una LPS del sistema.
- RF26. Estimar el impacto de los activos (**Ia**) de una LPS del sistema.
- RF27. Determinar el impacto de los riesgos (**Ira**) sobre los activos en un proyecto de una LPS del sistema.
- RF28. Determinar el peso de la **Fd** de cada riesgo en un proyecto de una LPS del sistema.
- RF29. Determinar el peso de la **Po** de cada riesgo en un proyecto de una LPS del sistema.
- RF30. Determinar el peso del **Ira** en un proyecto de una LPS del sistema.
- RF31. Obtener la exposición a cada riesgo en un proyecto de una LPS del sistema.
- RF32. Determinar de los umbrales de atención a los riesgos en un proyecto de una LPS del sistema.
- RF33. Generar reportes de identificación de riesgos en un proyecto de una LPS del sistema.
- RF34. Generar reportes de Evaluación de Riesgos en un proyecto de una LPS del sistema.
- RF35. Generar reportes por umbral en un proyecto de una LPS del sistema.

### 2.3.2. Requisitos No Funcionales

#### De Software

##### Servidor

- Se utilizará un servidor con sistema operativo **Ubuntu 12.04** o superior.
- Como servidor web se utilizará **Apache versión 2.2.22** o superior.
- Se utilizará **PostgreSQL versión 9.1.13** como gestor de bases de datos.

##### Ciente

- Se requiere versiones de **Windows XP** o superior, o cualquier distribución de **Linux**.
- En las computadoras de los clientes sólo se requiere de un navegador para Internet (**Internet Explorer versión 11.0.3** o superior, **Mozilla Firefox versión 20.0** o superior).

#### De Hardware

##### Cientes

- Se requiere tengan al menos **512 MB de RAM**.

- Se requiere una velocidad del **procesador de 800 MGz** como mínimo.

#### Servidores

- Se requiere que tengan al menos **1 GB de RAM**.
- La capacidad del disco duro puede variar atendiendo al tamaño de la base de datos (BD) y de la cantidad de LPS con que se trabaje.
- Se requiere una velocidad del **procesador de 2.0 GHz** como mínimo.

#### **Restricciones en el diseño y la implementación**

- El análisis y diseño del sistema estará basado en la metodología de desarrollo AUP con el uso del lenguaje de modelado UML.
- Se utilizará como IDE de desarrollo el **Sublime Text 3**.

#### **Apariencia o interfaz externa**

- La interfaz debe ser lo más sencilla posible, para que los miembros del equipo de desarrollo que no estén familiarizados con la herramienta no necesiten tanto tiempo de preparación para aprender a trabajar con el sistema.
- Debe regirse por los principios de diseño establecidos por la Universidad de las Ciencias Informáticas, específicamente para la marca XEDRO.

#### **Usabilidad**

- Debe ser de fácil y de rápido manejo para todos los usuarios, para facilitar y agilizar el proceso de Gestión de Riesgos en la LPS.
- Cualquier persona que tenga conocimientos básicos en computación o que haya interactuado anteriormente con la web, podrá usar este sistema.

#### **2.4. Actores del Sistema**

Los actores del sistema son aquellas personas, entidades, sistemas o cualquier agente externo que interactúe con el mismo. Cada actor puede interactuar con una o más funcionalidades. En la Tabla N se muestra la relación de todos los actores del sistema con sus respectivas descripciones (JACOBSON *et al.*, 2000).

Tabla 2: Descripción de los actores del sistema.

Actor	Descripción
Administrador	Es el encargado de realizar las acciones de administración y configuración del sistema así como la gestión de las LPS y de sus respectivos proyectos.
Analista de riesgos	Es el rol fundamental dentro del sistema, encargado de desarrollar todo el proceso de Evaluación de Riesgos, desde la determinación de la probabilidad de ocurrencia, impacto y facilidad de detección hasta la obtención de los valores de exposición y los umbrales, así como la gestión de los riesgos y los activos.
Usuario autorizado	Desarrolla fundamentalmente acciones de visualización de la información general y estadística del sistema.

## 2.5. Modelo de Sistema

Un modelo del sistema es una abstracción del sistema que se está estudiando en lugar de una representación alternativa de ese sistema, se utiliza como ayuda para el pensamiento al organizar y clasificar conceptos confusos e inconsistentes (SOMMERVILLE, 2005).

### 2.5.1. Diagrama de Casos de Uso del Sistema

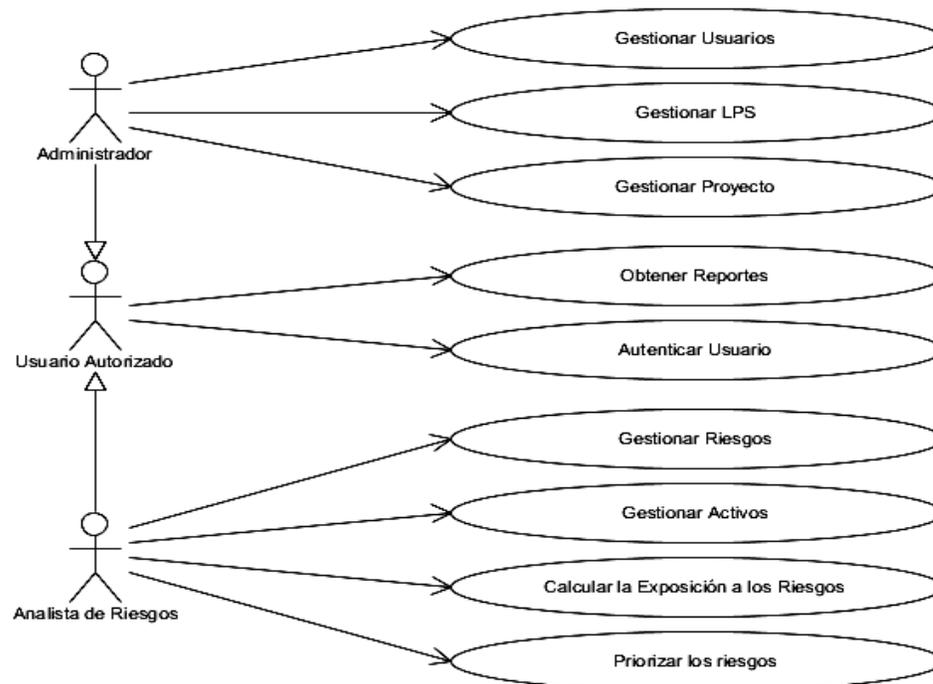


Fig. 4: Diagrama de Casos de Uso del Sistema.

### 2.5.2. Descripción textual de los Casos de Uso del Sistema

Para lograr una mejor comprensión de la funcionalidad de cada Caso de Uso del Sistema, se realiza la descripción textual de los mismos. A continuación se presenta la descripción textual del Caso de Uso Gestionar Riesgos, el resto de las descripciones se encuentran en el **Anexo #1**.

Tabla 3: Descripción textual del Caso de Uso Gestionar Riesgos.

<b>Caso de Uso:</b>	Gestionar Riesgos
<b>Actores:</b>	Analista de riesgos
<b>Propósito:</b>	Este caso de uso se realiza con el objetivo de gestionar los riesgos de las LPS.
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario desea agregar, modificar o eliminar riesgos en un proyecto de una LPS.
<b>Precondiciones:</b>	El usuario debe haber accedido al sistema con el rol de analista de riesgos.
<b>Referencias:</b>	RF19, RF20, RF21, RF22
<b>Prioridad:</b>	Crítico.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a la interfaz correspondiente a la Gestión de Riesgos.	2. Muestra en el área de trabajo una lista de los riesgos que existen actualmente registrados en el sistema, así como las opciones correspondientes a la Gestión de Riesgos: <ul style="list-style-type: none"> <li>▪ Agregar un riesgo.</li> <li>▪ Mostrar información de un riesgo.</li> <li>▪ Modificar un riesgo.</li> <li>▪ Eliminar un riesgo.</li> </ul>
3. Selecciona una de las opciones correspondientes a la Gestión de Riesgos.	4. El sistema muestra la interfaz correspondiente a la opción seleccionada. Si selecciona la opción Agregar un riesgo ir a la sección "Agregar un riesgo". Si selecciona la opción Mostrar información de un riesgo ir a la sección "Mostrar información de un riesgo". Si selecciona la opción Modificar un riesgo ir a la sección "Modificar un riesgo". Si selecciona la opción Eliminar un riesgo ir a la sección "Eliminar un riesgo".
<b>Sección: "Agregar un riesgo"</b>	

<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Selecciona la opción correspondiente a Agregar un riesgo.	2. Muestra el formulario requerido para agregar un riesgo.
3. Inserta los datos en el formulario y acepta la operación.	4. Adiciona el riesgo en la Base de Datos y muestra la lista de riesgos actualizada.
<b>Flujo alternativo de los eventos</b>	
4. Si los datos insertados son incorrectos o existen campos vacíos el sistema muestra un mensaje de rectificación y no se realiza la acción de adición.	
<b>Sección: “Mostrar información de un riesgo”</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Selecciona la opción correspondiente a Mostrar información de un riesgo.	2. Muestra una descripción del riesgo seleccionado, y sus datos generales.
3. Acepta la información que brinda el sistema.	4. Se muestra la lista de riesgos nuevamente.
<b>Sección: “Modificar un riesgo”</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Selecciona la opción correspondiente a Modificar un riesgo.	2. Muestra un formulario con los datos del riesgo seleccionado con la posibilidad de modificar cualquier campo.
3. Modifica los datos deseados en el formulario y acepta la operación.	4. Se actualizan los datos en la Base de Datos y se muestra la lista de riesgos actualizada.
<b>Flujo alternativo de los eventos</b>	
4. Si los datos a modificar son incorrectos o existen campos vacíos el sistema muestra un mensaje de rectificación y no se realiza la acción de modificación.	
<b>Sección: “Eliminar un riesgo”</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
5. Selecciona la opción correspondiente a eliminar un riesgo.	6. Elimina el riesgo seleccionado de la base de datos y muestra la lista de riesgos actualizada.
<b>Postcondiciones:</b>	Un riesgo es adicionado, eliminado, modificado o visualizado.

### 2.5.3. Patrones de Casos de Uso utilizados

#### Patrón CRUD total

CRUD, (del inglés, Creating, Reading, Updating, Deleting), consta de un caso de uso, llamado Información CRUD o Gestionar información, y se encarga de modelar todas las operaciones que pueden ser realizadas sobre una parte de la información de un tipo específico, tales como creación, lectura, actualización y eliminación.

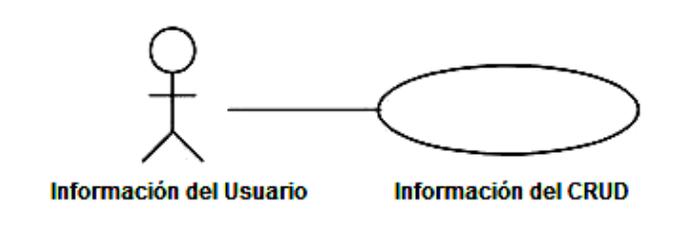


Fig. 5: Representación visual del patrón CRUD.

## Múltiples Actores

Roles comunes: Cuando dos actores juegan el mismo rol sobre un caso de uso, este rol es representado por otro actor, heredado por los actores que comparten este rol (OVERGAARD and PALMKVIST, 2004).

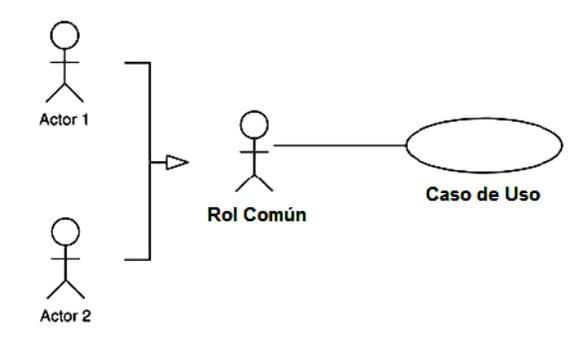


Fig. 6: Representación visual del patrón Múltiples Actores.

## 2.6. Elementos fundamentales de la arquitectura

### 2.6.1. Estilos y Patrones arquitectónicos

La diferencia entre estilos y patrones arquitectónicos no ha sido aclarada, (Bengtsson, 1999) plantea la existencia de dos grandes vertientes, que surgen de la discusión de los términos. Shaw y Garlan en el 1996 utilizan indistintamente los términos estilo arquitectónico y patrón arquitectónico. Por otro lado, Buschmann et al. (1996) establece diferencias sutiles entre ambos conceptos.

De cualquier forma, los estilos y los patrones establecen un vocabulario común, y brindan soporte a los ingenieros para conseguir una solución que haya sido aplicada con éxito anteriormente, ante ciertas situaciones de diseño. Además, su aplicación en el diseño de la arquitectura del sistema es determinante para la satisfacción de los requerimientos de calidad.

Buschmann et al. (1996) definen *estilo arquitectónico* como una familia de sistemas de software en términos de su organización estructural. Shaw y Garlan (1996) definen *estilo arquitectónico* como una familia de sistemas de software en términos de un patrón de organización estructural, que define un vocabulario de componentes y tipos de conectores y un conjunto de restricciones de cómo pueden ser combinadas.

De acuerdo a las clasificaciones dadas por Bass et al. (1998), se utiliza el estilo arquitectónico Llamada y Retorno: El sistema se constituye de un programa principal que tiene el control del sistema y varios subprogramas que se comunican con éste mediante el uso de llamadas.

Por otra parte, Buschmann et al. (1996) plantea que los *patrones arquitectónicos* expresan el esquema de organización estructural fundamental para sistemas de software. De acuerdo a las clasificaciones dadas, se utilizará Modelo-Vista-Controlador (del inglés, Model-View-Controller, MVC): Divide una aplicación interactiva en tres componentes. El modelo (model) contiene la información central y los datos. Las vistas (view) despliegan información al usuario. Los controladores (controllers) capturan la entrada del usuario. Las vistas y los controladores constituyen la interfaz del usuario (CAMACHO, 2004).

### 2.6.2. Patrones de diseño

#### Patrones GRASP

Patrones de Principios Generales para Asignar Responsabilidades. Estos patrones describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades.

- Experto en Información (o Experto):

Se utiliza con frecuencia en la asignación de responsabilidades; es un principio de guía básico que se utiliza continuamente en el diseño de objetos. El Experto no pretende ser una idea oscura o

extravagante; expresa la "intuición" común de que los objetos hacen las cosas relacionadas con la información que tienen.

Symfony2 por su parte al mapear la base de datos crea las entidades correspondientes al modelo entidad-relación de la misma, cada entidad tiene las funcionalidades necesarias para manipular la información de cada una de las tablas de la base de datos. Este patrón se evidencia en la clase Riesgo.php, la cual maneja toda la información referente a esa entidad.

- Creador:

Este patrón se utiliza cuando se pretende asignar a una clase B la responsabilidad de crear una instancia de clase A.

Symfony2 para cada entidad crea una clase controladora en la cual se crean instancias de las entidades, por tanto, cada clase controladora actúa como creadora de la entidad a la cual instancia. Un ejemplo en donde se utiliza este patrón es en la clase RiesgoController.php, donde se hace una instancia a la entidad riesgo.

- Alta Cohesión:

Un elemento con responsabilidades altamente relacionadas, y que no hace una gran cantidad de trabajo, tiene alta cohesión. Estos elementos pueden ser clases, subsistemas, etcétera.

En la aplicación propuesta se evidencia claramente este patrón. Todas las clases tienen una función específica, y en todas se realizan solamente tareas altamente relacionadas con su función. Un ejemplo es la clase Activo.php, la cual maneja solamente la información referente a los activos.

- Bajo Acoplamiento:

El acoplamiento es una medida de la fuerza con que un elemento está conectado a, tiene conocimiento de, confía en, otros elementos. Un elemento con bajo (o débil) acoplamiento no depende demasiado de otros elementos. Estos elementos pueden ser clases, subsistemas, etcétera.

Symfony2 utiliza el bajo acoplamiento en la inyección de dependencias, la cual permite estandarizar y centralizar la forma en que se construyen los objetos en tu aplicación. Un ejemplo del uso de este patrón es en la clase `ActivoController.php` cuando se hace una instancia a la entidad `Activo.php` no se define la forma de construir este objeto, ni interesa la dependencias que tenga esa clase con otras, solamente se le indica a la variable que se va a comportar como un objeto de la entidad `Activo.php`.

- Controlador:

Un Controlador es un objeto que no pertenece a la interfaz de usuario, responsable de recibir o manejar un evento del sistema.

Un Controlador define el método para la operación del sistema. Symfony2 utiliza el archivo `app_dev.php` como un controlador frontal, encargándose de atender las peticiones del usuario.

## Patrones GOF

Los patrones denominados “GoF” se descubren como una forma indispensable de enfrentarse a la programación a raíz del libro “Design Patterns—Elements of Reusable Software” de Erich Gamma, Richard Helm, Ralph Jonson y John Vlissides, a partir de entonces estos patrones son conocidos como los patrones de la pandilla de los cuatro (GoF, gang of four).

Se dividen en tres grandes grupos: Patrones de Creación, Patrones Estructurales, Patrones de Comportamiento (GAMMA *et al.*, 1980).

### Creacionales:

- Abstract Factory (Fábrica abstracta): Permite trabajar con objetos de distintas familias de manera que las familias no se mezclen entre sí y haciendo transparente el tipo de familia concreta que se esté usando. Cuando el framework necesita por ejemplo crear un nuevo objeto para una petición, busca en la definición de la factoría el nombre de la clase que se debe utilizar para esta tarea. Ejemplo de este patrón se ve en la clase `ProyectoController.php` cuando se crea un objeto de tipo `Proyecto.php` para trabajar con su información, solamente le indicamos el nombre de la clase a la cual va ser referencia.

### Estructurales:

- **Decorator (Envoltorio):** Añade funcionalidad a una clase, dinámicamente. El archivo principal.html.twig, que también se denomina plantilla global, almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. Entonces todas las páginas que heredan de ella utilizan todo el código HTML anteriormente utilizado.
- **Composite (Objeto compuesto):** Permite tratar objetos compuestos como si de uno simple se tratase. Sirve para construir objetos complejos a partir de otros más simples y similares entre sí, gracias a la composición recursiva y a una estructura en forma de árbol. Esto simplifica el tratamiento de los objetos creados, ya que al poseer todos ellos una interfaz común, se tratan todos de la misma manera.

## **2.7. Modelo de Diseño**

Un modelo de diseño contiene las especificaciones de las clases e interfaces de software en una aplicación. Entre la información general que muestran se incluyen:

- Clases, asociaciones y atributos.
- Interfaces, con sus operaciones y constantes.
- Métodos.
- Información acerca del tipo de los atributos.
- Navegabilidad.
- Dependencias (LARMAN, 1999).

A diferencia de las clases conceptuales del Modelo del Dominio, las clases de diseño muestran las definiciones de las clases de software en lugar de los conceptos del mundo real.

### **2.7.1. Diagrama de clases del diseño**

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. Los diagramas de clases permiten modelar la vista de diseño del sistema y el mejoramiento del modelo conceptual o de dominio (SOMMERVILLE, 2005).

El resto de los diagramas de clases del diseño pueden ser consultados en el **Anexo #2**.

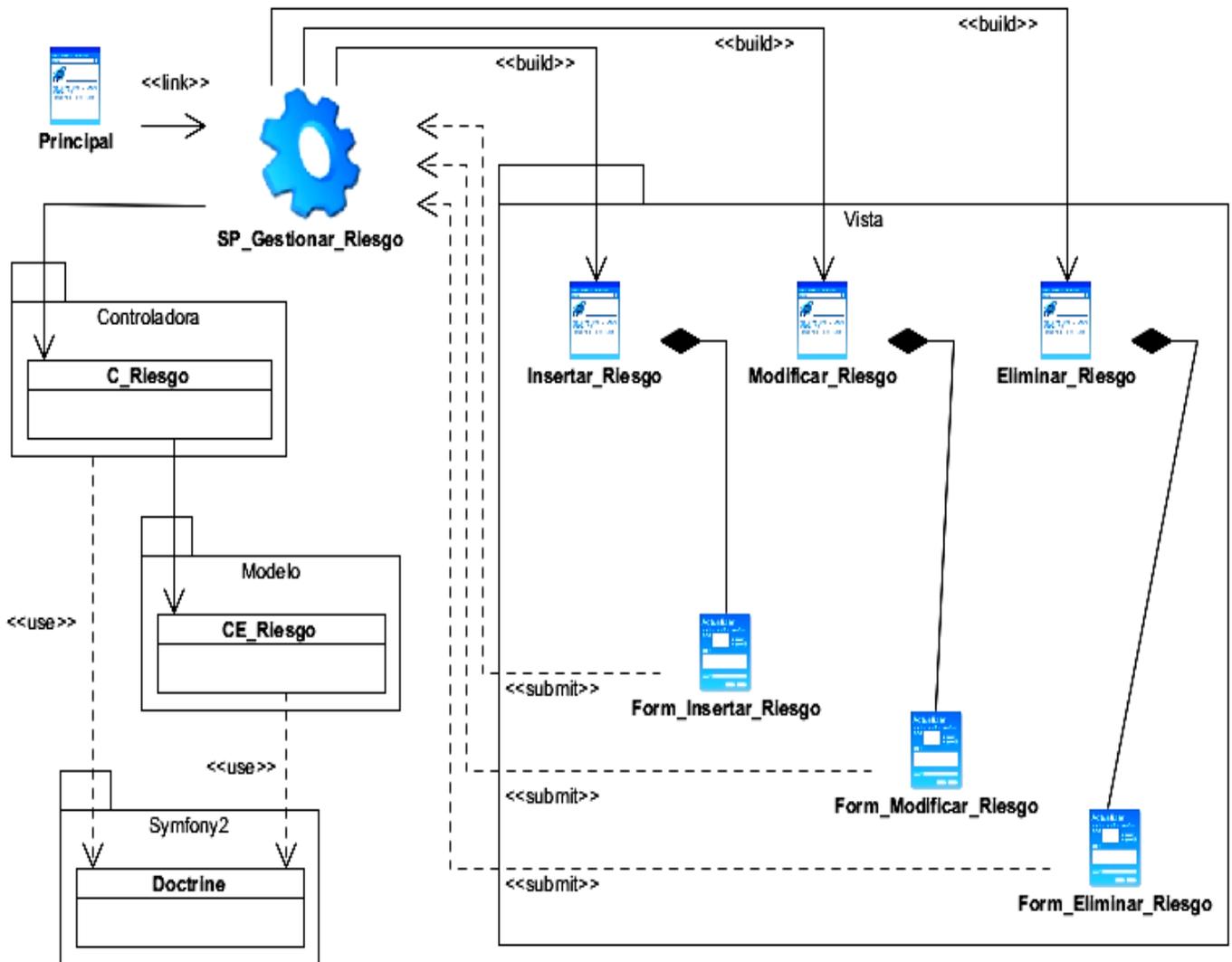


Fig. 7: Diagrama de clases de Diseño del CU Gestionar Riesgo.

### 2.8. Modelo entidad-relación

El diseño de Bases de Datos se utiliza para definir y después para especificar la estructura de los objetos de negocios que se emplean en el sistema cliente-servidor (PRESSMAN, 2001). El modelo de datos más extendido es el denominado ENTIDAD/RELACIÓN (E/R) En el modelo E/R se parte de una situación real a partir de la cual se definen entidades y relaciones entre dichas entidades (HERNÁNDEZ, 2002).

En la Figura 8 se muestra el Modelo entidad-relación de la propuesta de solución.

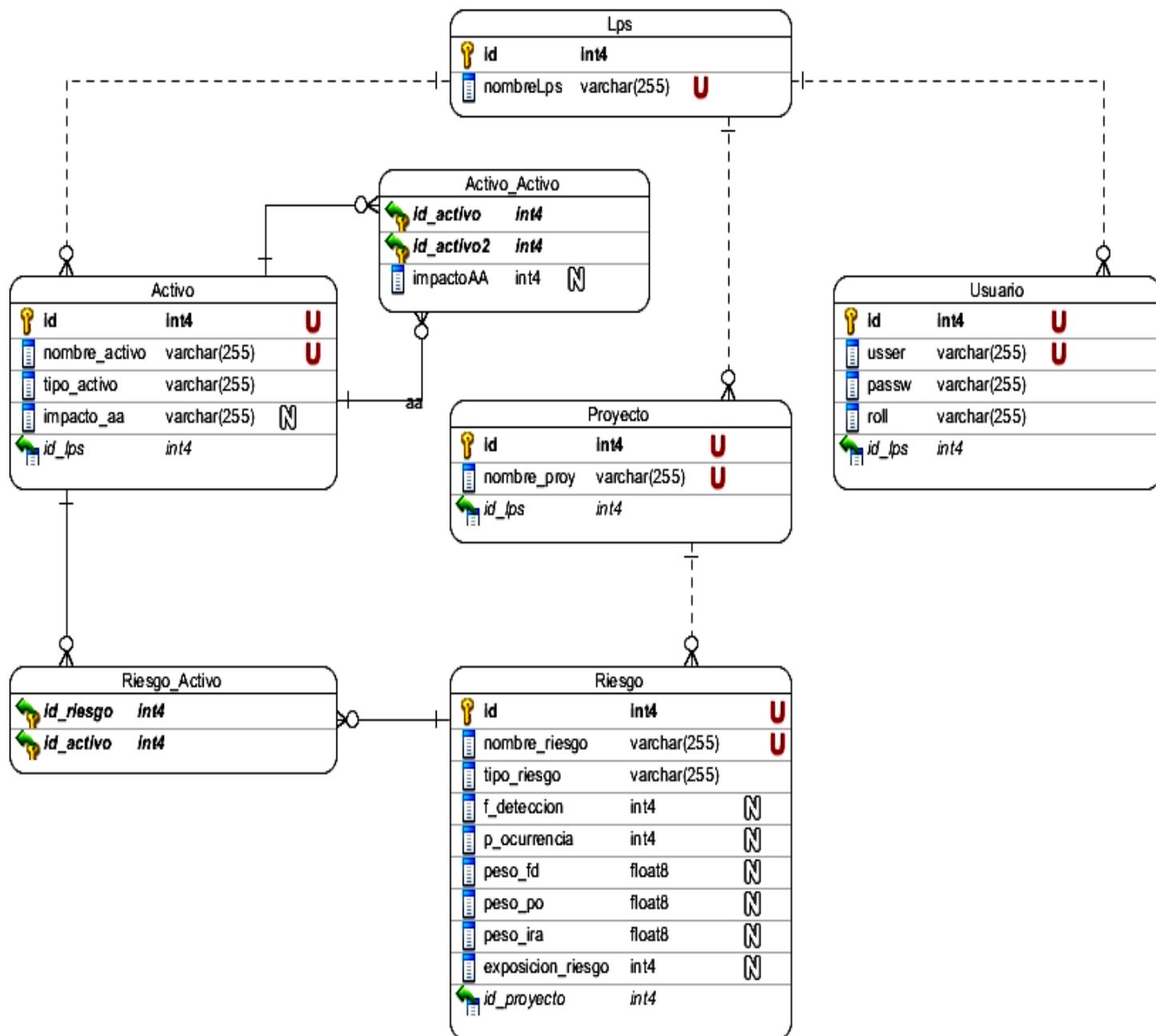


Fig. 8: Diagrama de Entidad - Relación.

### 2.9. Modelo de clases persistentes

El diagrama de clases persistentes muestra todas las clases capaces de mantener su valor en el espacio y en el tiempo. En la Figura 9 se muestra el Diagrama de Clases persistentes propuesto.

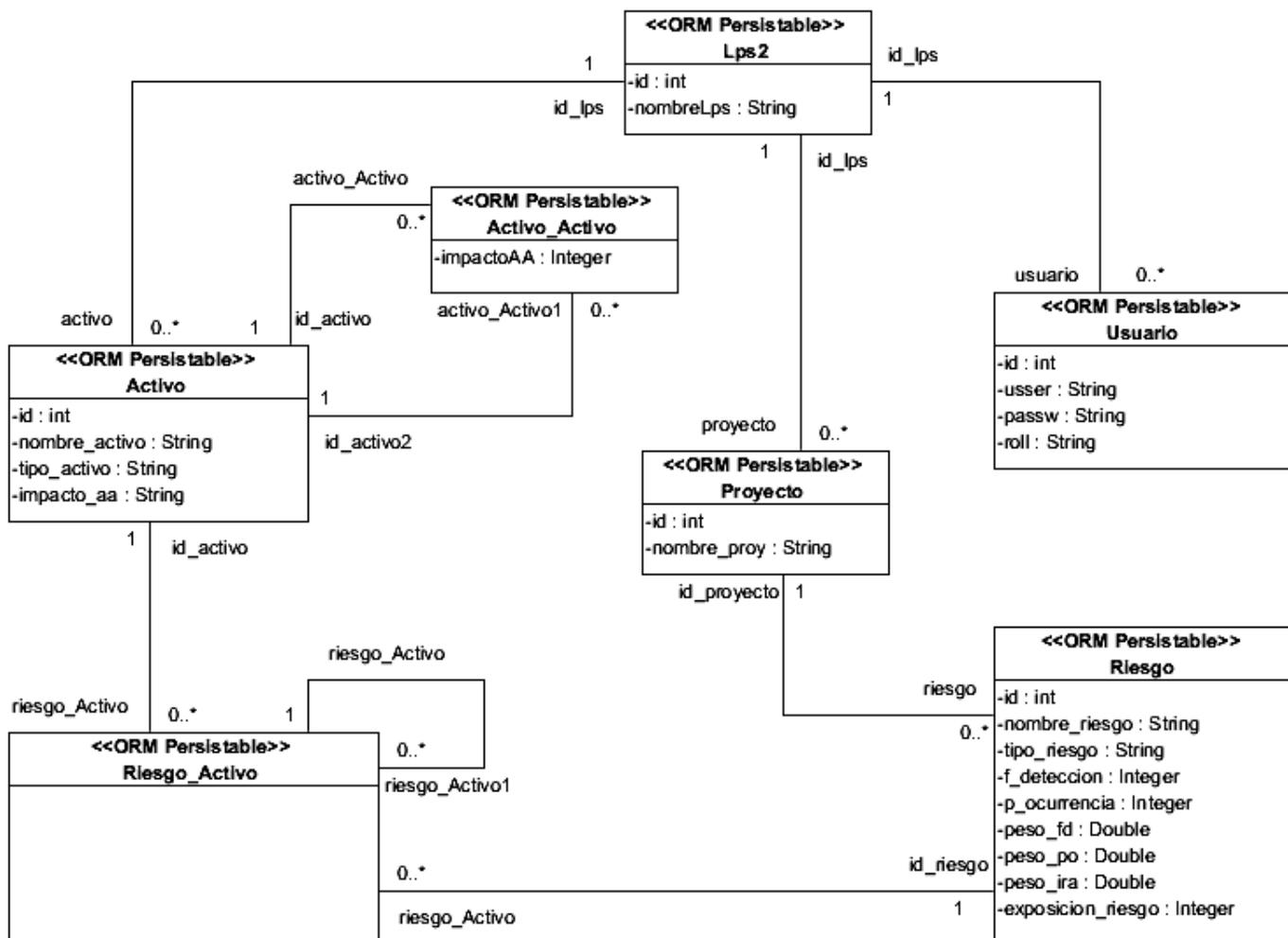


Fig. 9: Diagrama de Clases Persistentes.

### 2.10. Conclusiones parciales

La modelación del dominio y el sistema contribuyen al entendimiento hacia dentro del equipo de desarrollo. La documentación generada producto de la realización de las actividades del análisis y el diseño aseguran la continuidad y la ampliación de la solución en algún momento posterior a esta investigación. Los requisitos funcionales y no funcionales ubican al equipo de desarrollo en los objetivos de la solución y proporcionan una guía para su desarrollo y posterior validación. La utilización de patrones de diseño en la implementación de la solución propicia mayor organización, comprensión y menor tiempo de desarrollo.

## Capítulo 3: “Implementación y evaluación de la SPLRISK”

### 3.1. Introducción

En el presente capítulo se describen los elementos referidos a la implementación y la evaluación del sistema propuesto, se presentan los diagramas fundamentales de componentes y despliegue y luego se detalla el proceso de evaluación a través de las pruebas de software.

### 3.2. Modelo de implementación

El modelo de implementación describe cómo los elementos del modelo de diseño se implementan en términos de componentes. Describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración disponibles en el entorno de implementación y en el lenguaje de programación utilizado y cómo dependen los componentes unos de otros (ACUÑA, 2009).

#### 3.2.1. Diagrama de componentes

El Diagrama de componentes, según la metodología propuesta, es un diagrama que forma parte del Modelo de implementación (ACUÑA, 2009). A continuación se muestra el diagrama de componentes en términos de estructura de paquetes y luego una descripción por cada parte del patrón arquitectónico.

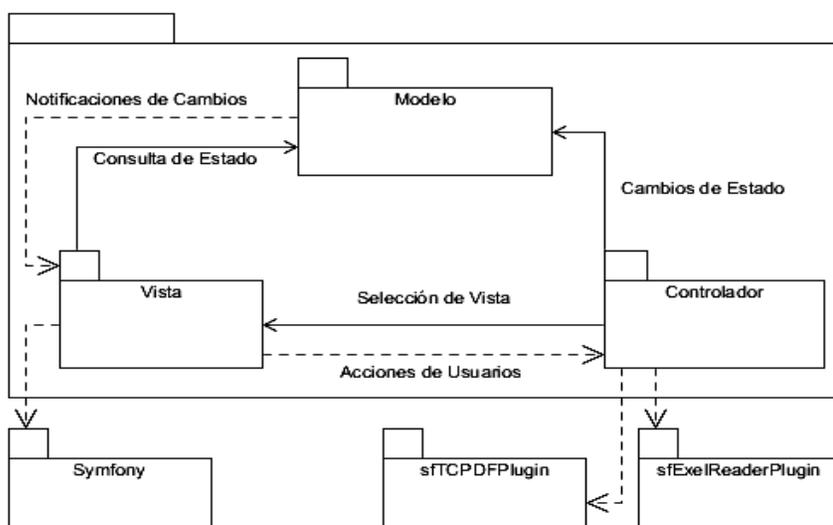


Fig. 10: Diagrama de Componentes.

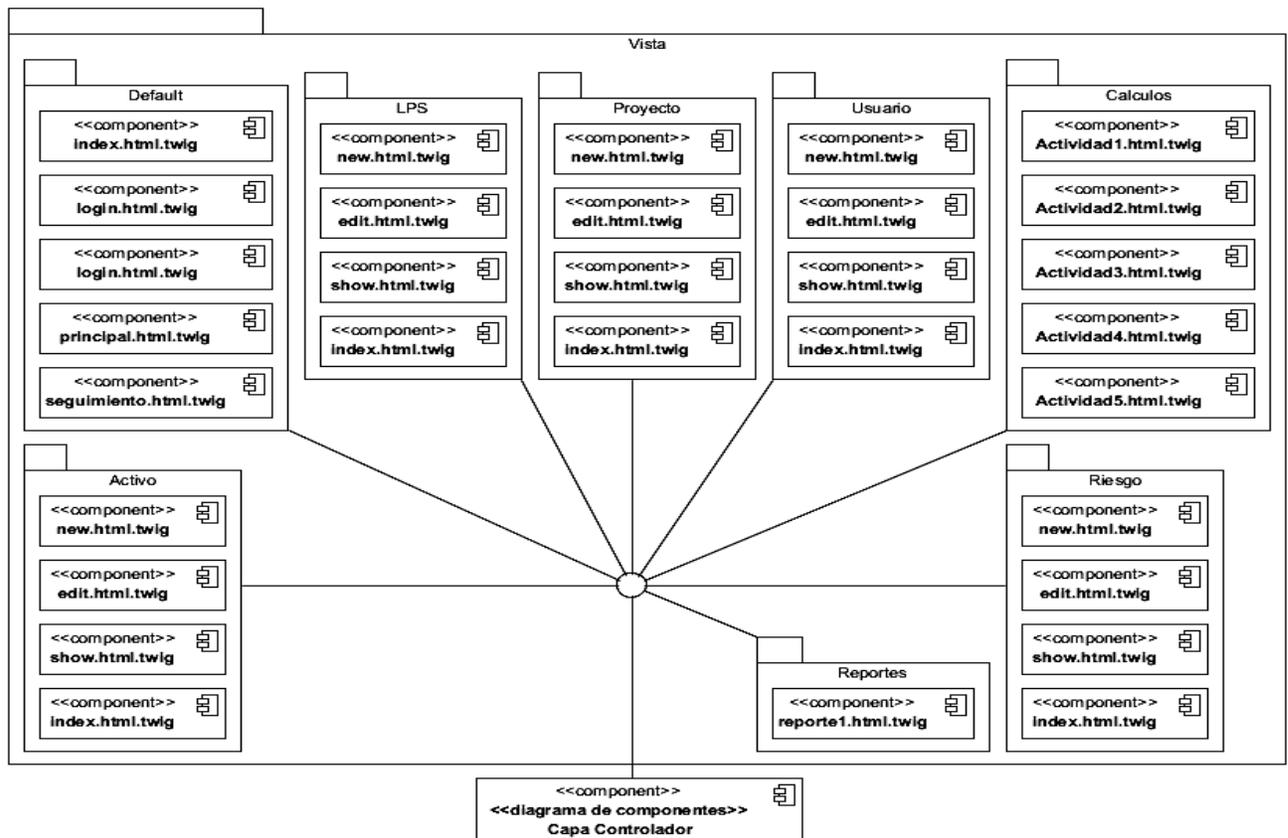


Fig. 11: Representación de la Vista del Diagrama de Componentes.

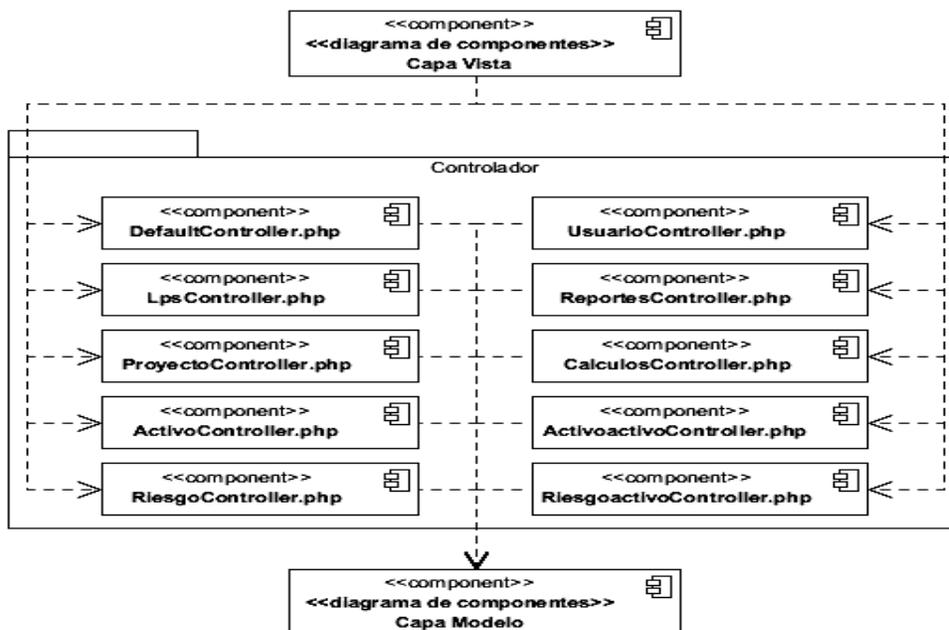


Fig. 12: Representación del Controlador del Diagrama de Componentes.

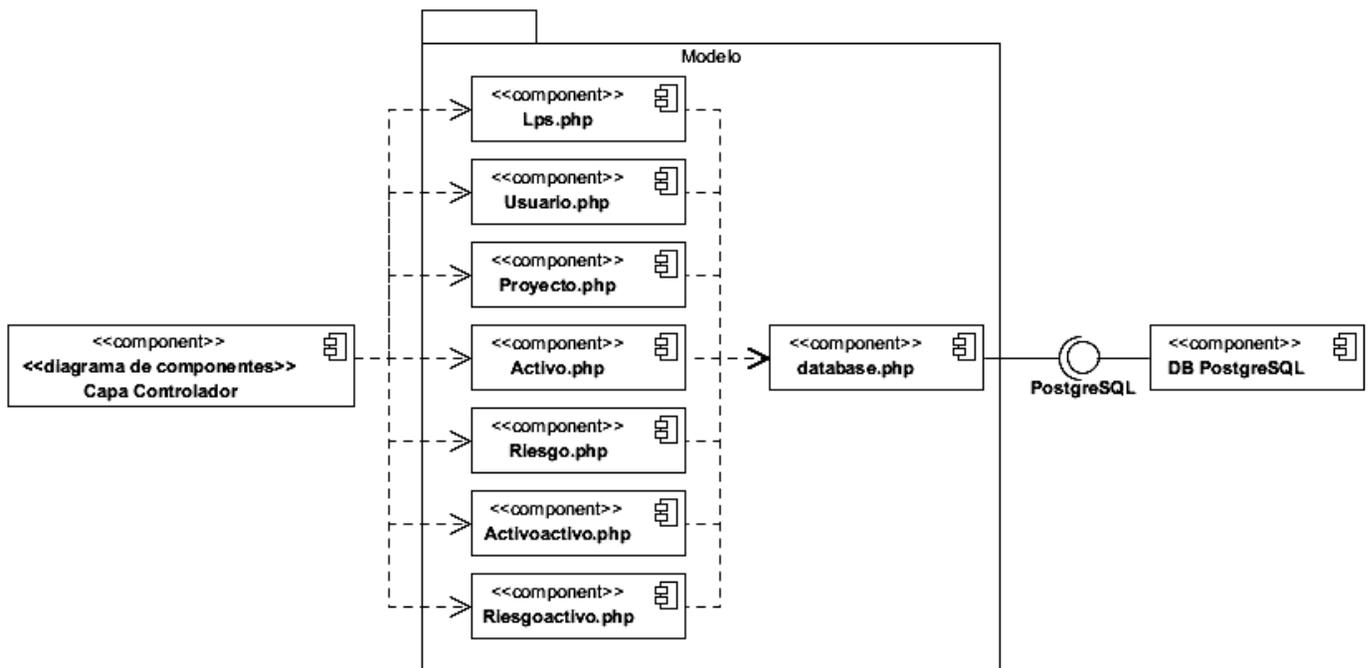


Fig. 13: Representación del Modelo del Diagrama de Componentes.

### 3.3. Modelo de despliegue

El modelo de despliegue define la arquitectura física del sistema por medios de nodos interconectados. Estos nodos son elementos de hardware sobre los cuales pueden ejecutarse los elementos de software.

El Diagrama de Despliegue es un tipo de diagrama del Lenguaje Unificado de Modelado que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes (JACOBSON *et al.*, 2000).

#### 3.3.1. Protocolos

**HTTP, (HyperText Transfer Protocol).** Protocolo usado para acceder a la Web (WWW), que generalmente utiliza el puerto 80. Se encarga de procesar y dar respuestas a las peticiones para visualizar una página web. Además sirve para el envío de información adicional como el envío de formularios con mensajes. Luego de finalizada la transacción, HTTP no guarda ninguna información sobre la misma, por lo tanto es considerado un protocolo "sin estado". Para guardar la información entre distintas peticiones, los webmasters suelen utilizar cookies o pasos de parámetros.

El HTTP está basado en el modelo cliente-servidor, en donde un cliente HTTP (un navegador por ejemplo) abre una conexión y realiza una solicitud al servidor. Este responde a la petición con un recurso (texto, gráficos, otros) o un mensaje de error, y finalmente se cierra la conexión. Uno de los más famosos mensajes de error HTTP es el “404 Not found” (INFORMÁTICA, 2009).

**HTTPS, (*Hypertext Transfer Protocol Secure*)**, es una combinación del protocolo HTTP y protocolos criptográficos. Se emplea para lograr conexiones más seguras en la WWW, generalmente para transacciones de pagos o cada vez que se intercambie información sensible (por ejemplo, claves) en internet. De esta manera la información sensible, en el caso de ser interceptada por un ajeno, estará cifrada.

Es recomendado para el sistema, por tratarse de un software que maneja información referente a un tema tan sensible como los riesgos y afectaciones de los proyectos (INFORMÁTICA, 2009).

**ADO, (*Microsoft ActiveX Data Objects*)**. En otras palabras, ADO es un mecanismo que utilizan los programas para comunicarse con las bases de datos. Provee una capa entre los lenguajes de programación y las bases de datos, lo que permite a los programadores escribir programas que accedan a datos, sin saber cómo está implementada la base de datos (sólo se debe tener cuidado en la conexión a la misma) (INFORMÁTICA, 2009).

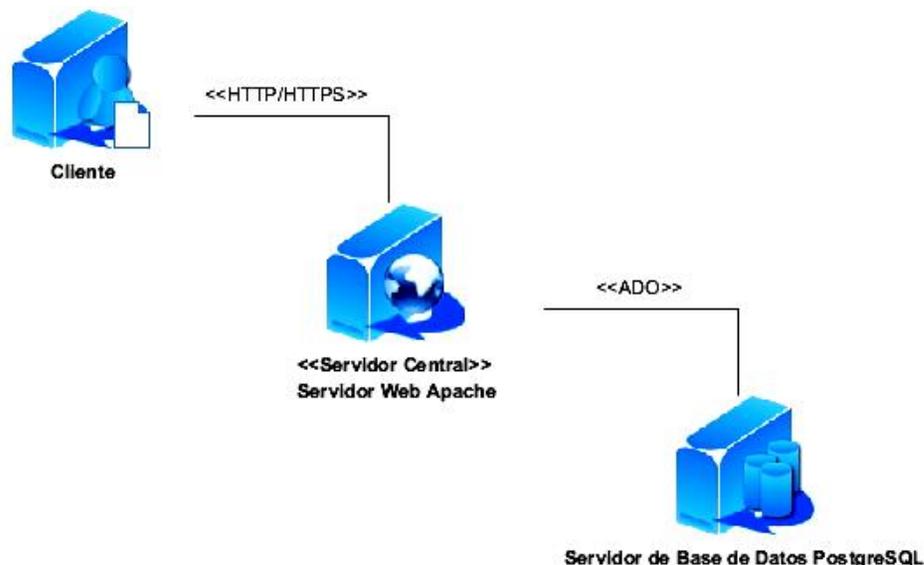


Fig. 14: Diagrama de Despliegue.

### 3.4. El proceso de pruebas

Según el sitio web “*pruebasdesoftware.com*” el único instrumento adecuado para determinar el status de la calidad de un producto software es el proceso de pruebas. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos. En las pruebas se usan casos de prueba, especificados de forma estructurada mediante técnicas de prueba. El proceso de pruebas, sus objetivos y los métodos y técnicas usados se describen en el plan de prueba.

Es recomendable que se realicen varios tipos de pruebas, siempre las pruebas funcionales tienen que estar completas y cubrir todas las funcionalidades posibles, solo así, se conseguirá una verdadera calidad en el software (PRUEBASDESFTWARE, 2005).

#### 3.4.1. Pruebas de Caja Negra

Las pruebas de **caja negra** son *pruebas funcionales dedicadas a “mirar” en el exterior de lo que se prueba*. Estas pruebas se denominan de varias formas, entre ellas, pruebas de caja “opaca”, pruebas de entrada/salida y pruebas inducidas por datos.

##### Descripción

Se centran principalmente en lo que “*se quiere*” de un módulo o sección específica de un software, es decir, es una manera de encontrar casos específicos en ese módulo, que atiendan a su especificación.

Se limitan a que el “probador” pruebe con “*datos*” de entrada y estudie cómo salen, sin preocuparse de lo que ocurre en el interior. Éstas, principalmente, se centran en módulos de interfaz de usuario (pantalla, ficheros y canales de comunicación) pero suelen ser útiles en cualquier módulo ya que todos o la mayoría tienen datos de entrada y salida que se pueden comprobar y verificar.

Como cualquier otra prueba, las de caja negra se apoyan y basan en la especificación de requisitos y documentación funcional. Para el desarrollo de las pruebas de caja negra existen diferentes técnicas, entre ellas:

- **Técnica de la Partición de Equivalencia:** divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- **Técnica del Análisis de Valores Límites:** prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
- **Técnica de Grafos de Causa-Efecto:** permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones (ADÁN, 2012).

En el desarrollo del proceso de pruebas se selecciona la técnica de **partición de equivalencia** y en lo siguiente se describirán las actividades asociadas al Caso de Uso Gestionar Riesgo. El resto de los diagramas puede ser consultado en el Anexo #3.

### Diseño del Caso de prueba

Tabla 4: Descripción por escenarios.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
SC1: Agregar un riesgo	EC 1.1: Agregar un riesgo satisfactoriamente.	El sistema agrega un riesgo y muestra la lista actualizada.	<ul style="list-style-type: none"> <li>• Seleccionar en el menú horizontal la opción “Identificación” y dentro escoger la opción “Gestionar Riesgo”.</li> <li>• Seleccionar la sección “Adicionar”.</li> <li>• Llenar los datos correspondientes.</li> <li>• Seleccionar la opción “Aceptar”.</li> </ul>
	EC 1.2: Se introducen datos incorrectos.	El sistema muestra un mensaje de error señalando que campos son incorrectos.	<ul style="list-style-type: none"> <li>• Seleccionar en el menú horizontal la opción “Identificación” y dentro escoger la opción “Gestionar Riesgo”.</li> <li>• Seleccionar la sección “Adicionar”.</li> <li>• Llenar los datos correspondientes.</li> <li>• Seleccionar la opción “Aceptar”.</li> </ul>
SC2: Mostrar información de un riesgo	EC 2.1: Mostrar información de un riesgo satisfactoriamente.	El sistema muestra los datos de un riesgo en una ventana.	<ul style="list-style-type: none"> <li>• Seleccionar en el menú horizontal la opción “Identificación” y dentro escoger la opción “Gestionar Riesgo”.</li> </ul>

			<ul style="list-style-type: none"> <li>• Seleccionar la sección “Mostrar”.</li> </ul>
SC3: Modificar un riesgo.	EC 3.1: Modificar un riesgo satisfactoriamente.	El sistema modifica los datos del riesgo y muestra la lista actualizada.	<ul style="list-style-type: none"> <li>• Seleccionar en el menú horizontal la opción “Identificación” y dentro escoger la opción “Gestionar Riesgo”.</li> <li>• Seleccionar la sección “Modificar”.</li> <li>• Modificar los datos deseados.</li> <li>• Seleccionar la opción “Aceptar”.</li> </ul>
	EC 3.2: Se introducen datos incorrectos.	El sistema muestra un mensaje de error señalando que campos son incorrectos.	<ul style="list-style-type: none"> <li>• Seleccionar en el menú horizontal la opción “Identificación” y dentro escoger la opción “Gestionar Riesgo”.</li> <li>• Seleccionar la sección “Modificar”.</li> <li>• Modificar los datos deseados.</li> <li>• Seleccionar la opción “Aceptar”.</li> </ul>
SC4: Eliminar un riesgo	EC 4.1: Eliminar un riesgo satisfactoriamente.	El sistema elimina el riesgo seleccionado y muestra la lista actualizada.	<ul style="list-style-type: none"> <li>• Seleccionar en el menú horizontal la opción “Identificación” y dentro escoger la opción “Gestionar Riesgo”.</li> <li>• Seleccionar la sección “Eliminar”.</li> </ul>

**Tabla 5: Variables del Caso de Prueba.**

ID del escenario	Escenario	V1	V2	Resultado de la prueba
EC 1.2	Se introducen datos incorrectos.	I (23)	V (Amenaza)	Error.
EC 2.1	Mostrar información de un riesgo satisfactoriamente.	NA	NA	Satisfactorio.
EC 3.1	Modificar un riesgo satisfactoriamente.	V (fallo eléctrico)	V (Amenaza)	Satisfactorio.
EC 4.1	Eliminar un riesgo satisfactoriamente.	NA	NA	Satisfactorio.

**Tabla 6: Comprobación de variables.**

ID del escenario	Escenario	V1	V2	V3	V4	Respuesta del Sistema	Resultado de la prueba
EC 1.1	Agregar un riesgo satisfactoria mente.	V (riesgo)	V (Amenaza)	V (1)	V (act1)	El sistema agrega un riesgo y muestra la lista actualizada.	Satisfactorio.
EC 1.2	Se introducen datos incorrectos.	I (23)	V (Amenaza)	V (1)	V (act1)	El sistema muestra un mensaje de error señalando que campos son incorrectos.	Error.
EC 2.1	Mostrar información de un riesgo satisfactoria mente.	NA	NA	NA	NA	El sistema muestra los datos de un riesgo en una ventana.	Satisfactorio.
EC 3.1	Modificar un riesgo satisfactoria mente.	V (fallo eléctrico)	V (Amenaza)	V (1)	V (act1)	El sistema modifica los datos del riesgo y muestra la lista actualizada.	Satisfactorio.
EC 3.2	Se introducen datos incorrectos.	V (45)	V (Amenaza)	V (1)	V (act1)	El sistema muestra un mensaje de error señalando que campos son incorrectos.	Error.
EC 4.1	Eliminar un riesgo satisfactoria mente.	NA	NA	NA	NA	El sistema elimina el riesgo seleccionado y muestra la lista actualizada.	Satisfactorio.

**3.4.2. Resultados de las pruebas**

Una vez diseñados todos los Casos de prueba a partir de las descripciones textuales de los Casos de Uso del Sistema se desarrollaron tres iteraciones.

**Primera Iteración:** Se obtuvieron cinco No Conformidades de acuerdo a como se muestra en la tabla 7 a continuación.

**Segunda Iteración:** Una vez corregidas las No Conformidades anteriores, se ejecutaron nuevamente los diseños elaborados, obteniéndose dos No Conformidades en el caso de uso que modela el algoritmo para el cálculo de la exposición al riesgo. Luego de corregidas las mismas, se decide no realizar la tercera iteración.

**Tabla 7: No Conformidades del Proceso de pruebas**

No. de Iteraciones	Total de no conformidades	No procede	Resueltas
1ra Iteración	5	0	5
2da Iteración	2	0	2

### **3.5. Conclusiones**

Los elementos identificados como parte de las interfaces visuales así como la arquitectura de información de SPLRISK están adecuados a la propuesta de diseño elaborada en la Universidad, lo cual promueve un producto de software estandarizado y visualmente relacionado al resto de los productos que se desarrollan.

Las pruebas que se seleccionan se consideran acertadas y adecuadas pues buscan el correcto funcionamiento de la aplicación desde la experiencia del usuario.

## Conclusiones Generales

1. Las soluciones existentes analizadas no responden totalmente al problema de la investigación pues no contienen un enfoque claro dirigido a Líneas de Productos de Software.
2. La utilización de herramientas y tecnologías de plataforma abierta garantiza la soberanía tecnológica impulsada por la universidad y el país.
3. La solución que se propone es multiplataforma lo cual amplía la gama de usuarios y las posibilidades de despliegue en mayor cantidad de entornos de desarrollo.
4. La documentación técnica obtenida durante el proceso de investigación propicia la ampliación y escalabilidad futura de la herramienta.
5. La utilización de patrones de diseño en la implementación de la solución propicia mayor organización, comprensión y menor tiempo de desarrollo.
6. El diseño gráfico propuesto se adecua a la identidad de los productos de la Universidad.
7. Las pruebas que se seleccionan se consideran acertadas y adecuadas pues buscan el correcto funcionamiento de la aplicación desde la experiencia del usuario.

## **RECOMENDACIONES**

Los autores recomiendan:

1. Implementar el proceso de seguimiento y control de los riesgos en la herramienta obtenida.
2. Desarrollar la variante basada en Análisis Dinámico en el cálculo de la exposición a los riesgos.

## REFERENCIAS BIBLIOGRÁFICAS

1. ACUÑA. *Selección de Metodologías de Desarrollo para Aplicaciones Web en la Facultad de Informática de la Universidad de Cienfuegos*. Edición electrónica gratuita. Cienfuegos, Universidad de Cienfuegos, 2009. 120 p.
2. ADÁN. *Pruebas de caja negra*, globeTESTING, 2012. [Disponible en: <http://www.globetesting.com/2012/08/pruebas-de-caja-negra/>]
3. ALEGSA. *¿Qué es un aplicación?*, 2007. [2013]. Disponible en: [www.alegsa.com.ar/Dic/aplicacion.php](http://www.alegsa.com.ar/Dic/aplicacion.php)
4. ALEGSA.COM. *Definición de SGBD*, Diccionario de Informática, 1998. [Disponible en: <http://www.alegsa.com.ar/Dic/sqbd.php>]
5. ALEGSAONLINE.COM. *¿Qué es un lenguaje de programación?*, 2006. [Disponible en: <http://www.alegsaonline.com/art/11.php>]
6. CAMACHO. *Arquitecturas de Software. Guía de estudio.*, 2004. p.
7. CEBRIÁN. *La década prodigiosa de Bill Gates*. El País. España, 2011.
8. CLEMENT and NORTHRHOP. *Software Product Lines: Practices and Patterns*. Pittsburgh, EE.UU., Carnegie Mellon University, 2002. 563 p. 0-201-70332-7
9. CONML. *Preguntas habituales acerca de ConML*, Instituto de Ciencias del Patrimonio, 2013. [Disponible en: <http://www.conml.org/About.aspx>]
10. CRESPO; GÓMEZ, et al. *Methodology for Information Systems Risk Analysis and Management*. Madrid, 2006. 140 p.
11. CHARETTE. *Software Engineering Risk Analysis and Management*. EE.UU., Mcgraw Hill Software Engineering Series, 1989. p.
12. CHRISSIS; KONRAD, et al. *CMMI, Guía para la integración de procesos y la mejora de productos*. Segunda Edición. Madrid, 2009. 630 p. 9788478290963
13. DRAKE. *Proceso de desarrollo de aplicaciones software*, Universidad de Cantabria, 2008. [Disponible en: [http://www.ctr.unican.es/asignaturas/MC\\_OO/Doc/OO\\_08\\_I2\\_Proceso.pdf](http://www.ctr.unican.es/asignaturas/MC_OO/Doc/OO_08_I2_Proceso.pdf)]
14. DTIC. *Introducción al Agile UP*, Universidad Nacional de Costa Rica, 2009. [Disponible en: <http://www.cc.una.ac.cr/AUP/html/overview.html>]
15. ECURED. *XEDRO GESPRO*, 2011. [2013]. Disponible en: <http://www.ecured.cu/index.php/Gespro>
16. EGUILUZ. *DESARROLLO WEB ÁGIL CON SYMFONY2*. primera edición. Symfony.es, 2011. p.
17. ESPAÑOLA. *Mapa*, 2012. [2013]. Disponible en: <http://lema.rae.es/drae/>
18. EUMET. *Por qué utilizar RUP para desarrollar aplicaciones web*, 2011. [2014]. Disponible en: <http://www.eumed.net/libros-gratis/2009c/584/por%20que%20utilizar%20RUP%20para%20desarrollar%20aplicaciones%20web.htm>
19. FERNÁNDEZ. *Propuesta de evaluación de riesgos para los proyectos desarrollados en las Líneas de Productos de Software del Departamento de Geoinformática*. Geoinformática. La Habana, Universidad de las Ciencias Informáticas, 2013a. 79. p.
20. ---. *SISTEMA DE INFORMACIÓN GEOGRÁFICA PARA EL CENTRO NACIONAL DE INVESTIGACIONES SISMOLÓGICAS*. Geoinformática. La Habana, Universidad de las Ciencias Informáticas, 2013b. 105. p.
21. FERNÁNDEZ and HIDALGO. *Sistema automatizado para la gestión de los datos del Balance Nacional de recursos y reservas de petróleo y gas de la Oficina Nacional de Recursos Minerales.*: Geoinformática. La Habana, Universidad de las Ciencias Informáticas, 2008. 119. p.

22. FLORES. *METODOLOGÍAS ÁGILES. PROCESO UNIFICADO ÁGIL (AUP)*. Universidad Unión Bolivariana, 2013. [Disponible en: <http://prezi.com/begizcu7b0ho/copy-of-aup/>]
23. GAMMA; HELM, et al. *Design Patterns—Elements of Reusable Software*. 1ra Edición. Prentice Hall, Adiosn Wesley, 1980. p.
24. GROUP. *What is PHP?*, 2009. [2013]. Disponible en: [www.php.net](http://www.php.net)
25. HERNÁNDEZ. *Multimedia de casos prácticos resueltos utilizando Bases de Datos*, Universidad Miguel Hernández, 2002.
26. INFORMÁTICA. *Introducción al Agile UP*, Universidad Nacional de Costa Rica, 2009. [Disponible en: <http://www.cc.una.ac.cr/AUP/html/overview.html>]
27. JACOBSON; BOOCH, et al. *El proceso Unificado de Desarrollo de Software*. Madrid, PEARSON EDUCACION.SA, 2000. 458 p. 84-7829-036-2
28. JAVASCRIPTYA. *Tutorial de Javascript*, 2012. [2013]. Disponible en: <http://www.javascriptya.com.ar/>
29. LARMAN. *UML y Patrones: Introducción al análisis y programación orientada a objetos*. México DF, Prentice Hall, 1999. 536 p.
30. LLANES. *Riesgo e incertidumbre en la gestión de proyectos informáticos*
31. 2003. 17.
32. MAP. *Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información. Método. (v 1.1)*. PÚBLICAS, M. D. A., Catálogo general de publicaciones oficiales, 2006.
33. MONTILVA. *Desarrollo de Software Basado en Líneas de Productos de Software*. IEEE Computer Society Región 9, Venezuela, 2006. p.
34. MUESTRAS. *Ingeniería del Software de Gestión*, Universidad Complutense de Madrid, 2009. [Disponible en: <http://www.vc.ehu.es/jiwotvim/ISOFT2008-2009/Teoria/Bloquel/Transp-03Proceso-Pressman.pdf>]
35. OVERGAARD and PALMKVIST. *Use Cases: Patterns and Blueprints*. 1ra edición. Addison-Wesley Professional, 2004. 464 p. 0131451340
36. PATÓN. *Metodologías de Desarrollo de Software*, Universidad de Castilla-La Mancha, 2007. [Disponible en: ]
37. PÉREZ; GONZÁLEZ, et al. *Metodología del desarrollo del Software*, Universidad Politécnica de Valencia, 2008. [Disponible en: <http://www.slideshare.net/geurquizo/metodologias-de-desarrollo-del-software>]
38. PESTANO. *Propuesta de modelo de desarrollo para líneas de productos de software en centros de producción*. Dirección Técnica. La Habana, Universidad de las Ciencias Informáticas, 2011. 80. p.
39. PMI. *Guía de los Fundamentos para la dirección de proyectos (Guía del PMBOK)*. Pennsylvania, EE.UU., Project Management Institute, Inc., 2009. 502 p. 978-1-933890-72-2
40. PRESSMAN. *Ingeniería del Software. Un enfoque práctico*. Sexta. Madrid, Mc Graw-Hill/Interamericana de España, S.A, 2005a. p.
41. ---. *Ingeniería del Software. Un enfoque práctico*. 5ta. 2001. p.
42. ---. *Ingeniería del Software: Un enfoque práctico*. Sexta Edición. 2005b. 927 p.
43. PROGRAMACIÓN. *Programación Web*, 2009. [Disponible en: <http://www.lenguajes-de-programacion.com/programacion-web.shtml>]
44. PRUEBASDESOFTWARE. *Las pruebas de software*, 2005. [Disponible en: <http://www.pruebasdesoftware.com/laspruebasdesoftware.htm>]
45. RIQUELME and MORALES. *Sistema Gestor de Base de Datos Relacionales*, 2010. 14.

46. ROSSEL; PEROVICH, *et al.* *Feature model to product architectures: Applying MDE to Software Product Lines*, Universidad de Chile, 2009. [2012]. Disponible en: [http://swp.dcc.uchile.cl/TR/2008/TR\\_DCC-2008-008.pdf](http://swp.dcc.uchile.cl/TR/2008/TR_DCC-2008-008.pdf)
47. SAMETINGER. *Software Engineering with Reusable Components*. New York, EE.UU., 2007. 285 p.
48. SEI. *Software Product Line*, 2011.
49. SOMMERVILLE. *Ingeniería del Software*. Séptima edición. Madrid, Pearson Educación, 2005. 712 p.
50. SYSTEMS. *Herramientas para el desarrollo de Software*, SOLUS S.A., 2007. [Disponible en: [http://www.sparxsystems.com.ar/platforms/software\\_development.html?](http://www.sparxsystems.com.ar/platforms/software_development.html?)
51. TORRES Bases de datos espaciales: Spatial Databases. *Revista Internacional de la Tierra MAPPING Interactivo*, 2009.
52. ZAMITIZ. *El Lenguaje Unificado de Modelado (UML)*, Universidad Autónoma de México, 2011. [Disponible en: <http://profesores.fi-b.unam.mx/carlos/aydoo/uml.html>

## Anexo 1: Descripción textual de los Casos de Uso.

### Descripción textual del Caso de Uso Gestionar LPS.

<b>Caso de Uso:</b>	Gestionar LPS
<b>Actores:</b>	Administrador
<b>Propósito:</b>	Este caso de uso se realiza con el objetivo de gestionar las LPS.
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario desea agregar, modificar o eliminar una LPS.
<b>Precondiciones:</b>	El usuario debe haber accedido al sistema con el rol de Administrador.
<b>Referencias:</b>	RF7, RF8, RF9, RF10
<b>Prioridad:</b>	Crítico.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a la interfaz de Gestión de LPS.	2. Muestra en el área de trabajo una lista de las LPS que existen actualmente registradas en el sistema, así como las opciones correspondientes a la Gestión de LPS: <ul style="list-style-type: none"> <li>▪ Insertar una LPS.</li> <li>▪ Modificar una LPS.</li> <li>▪ Mostrar información de una LPS.</li> <li>▪ Eliminar una LPS</li> </ul>
3. Selecciona una de las opciones correspondientes a la Gestión de LPS.	4. El sistema muestra la interfaz correspondiente a la opción seleccionada. Si selecciona la opción Agregar una LPS ir a la sección "Agregar una LPS". Si selecciona la opción Mostrar información de una LPS ir a la sección "Mostrar información de una LPS". Si selecciona la opción Modificar una LPS ir a la sección "Modificar una LPS". Si selecciona la opción Eliminar una LPS ir a la sección "Eliminar una LPS".
<b>Sección: "Agregar una LPS"</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
5. Selecciona la opción correspondiente a Agregar una LPS.	6. Muestra el formulario requerido para agregar una LPS.
7. Inserta los datos en el formulario y acepta la operación.	8. Adiciona la LPS en la Base de Datos y muestra la lista de LPS actualizada.
<b>Flujo alternativo de los eventos</b>	
8. Si los datos insertados no son completos el sistema muestra un mensaje de rectificación y no se realiza la acción de adición.	
<b>Sección: "Mostrar información de una LPS"</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>

5. Selecciona la opción correspondiente a Mostrar información de una LPS.	6. Muestra una descripción de la LPS seleccionada, y sus datos generales.
7. Acepta la información que brinda el sistema.	8. Se muestra la lista de LPS nuevamente.
<b>Sección: "Modificar una LPS"</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
5. Selecciona la opción correspondiente a Modificar una LPS.	6. Muestra un formulario con los datos de la LPS seleccionada con la posibilidad de modificar cualquier campo.
7. Modifica los datos deseados en el formulario y acepta la operación.	8. Se actualizan los datos en la Base de Datos y se muestra la lista de LPS actualizada.
<b>Flujo alterno de los eventos</b>	
8. Si los datos a modificar no son completos el sistema muestra un mensaje de rectificación y no se realiza la acción de modificación.	
<b>Sección: "Eliminar una LPS"</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
5. Selecciona la opción correspondiente a eliminar una LPS.	6. Elimina la LPS seleccionada de la base de datos y muestra la lista de LPS actualizada.
<b>Postcondiciones:</b>	Una LPS es adicionada, eliminada, modificada o visualizada.

### Descripción textual del Caso de Uso Gestionar Proyecto.

<b>Caso de Uso:</b>	Gestionar Proyecto
<b>Actores:</b>	Administrador
<b>Propósito:</b>	Este caso de uso se realiza con el objetivo de gestionar los proyectos.
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario desea hacer una modificación en la lista de proyectos del sistema.
<b>Precondiciones:</b>	El usuario debe haber accedido al sistema con el rol de administrador.
<b>Referencias:</b>	RF11, RF12, RF13, RF14
<b>Prioridad:</b>	Crítico.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a la interfaz de Gestión de Proyecto.	<p>2. Muestra en el área de trabajo una lista de los proyectos que existen actualmente registrados en el sistema, así como las opciones correspondientes a la Gestión de Proyecto:</p> <ul style="list-style-type: none"> <li>▪ Insertar un Proyecto.</li> <li>▪ Modificar un Proyecto.</li> <li>▪ Mostrar información de un Proyecto.</li> <li>▪ Eliminar un Proyecto.</li> </ul>

3. Selecciona una de las opciones correspondientes a la Gestión de Proyecto.	4. El sistema muestra la interfaz correspondiente a la opción seleccionada. Si selecciona la opción Agregar un Proyecto ir a la sección "Agregar un Proyecto". Si selecciona la opción Mostrar información de un Proyecto ir a la sección "Mostrar información de un Proyecto". Si selecciona la opción Modificar un Proyecto ir a la sección "Modificar un Proyecto". Si selecciona la opción Eliminar un Proyecto ir a la sección "Eliminar un Proyecto".
<b>Sección: "Agregar un Proyecto"</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
5. Selecciona la opción correspondiente a Agregar un Proyecto.	6. Muestra el formulario requerido para agregar un Proyecto.
7. Inserta los datos en el formulario y acepta la operación.	8. Adiciona el Proyecto en la Base de Datos y muestra la lista de Proyectos actualizada.
<b>Flujo alterno de los eventos</b>	
8. Si los datos insertados no son completos el sistema muestra un mensaje de rectificación y no se realiza la acción de adición.	
<b>Sección: "Mostrar información de un Proyecto"</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
5. Selecciona la opción correspondiente a Mostrar información de un Proyecto.	6. Muestra una descripción del Proyecto seleccionado, y sus datos generales.
7. Acepta la información que brinda el sistema.	8. Se muestra la lista de Proyectos nuevamente.
<b>Sección: "Modificar un Proyecto"</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
5. Selecciona la opción correspondiente a Modificar un Proyecto.	6. Muestra un formulario con los datos del Proyecto seleccionado con la posibilidad de modificar cualquier campo.
7. Modifica los datos deseados en el formulario y acepta la operación.	8. Se actualizan los datos en la Base de Datos y se muestra la lista de Proyectos actualizada.
<b>Flujo alterno de los eventos</b>	
8. Si los datos a modificar no son completos el sistema muestra un mensaje de rectificación y no se realiza la acción de modificación.	
<b>Sección: "Eliminar un Proyecto"</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
5. Selecciona la opción correspondiente a eliminar un Proyecto.	6. Elimina el Proyecto seleccionado de la base de datos y muestra la lista de Proyectos actualizada.

<b>Postcondiciones:</b>	Un Proyecto es adicionado, eliminado, modificado o visualizado.
-------------------------	---

### Descripción textual del Caso de Uso Gestionar Usuario.

<b>Caso de Uso:</b>	Gestionar Usuario.
<b>Actores:</b>	Administrador
<b>Propósito:</b>	Este caso de uso se realiza con el objetivo de gestionar los usuarios.
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario desea hacer una modificación en la lista de usuarios del sistema.
<b>Precondiciones:</b>	El usuario debe haber accedido al sistema con el rol de administrador.
<b>Referencias:</b>	RF3, RF4, RF5, RF6
<b>Prioridad:</b>	Crítico.

#### Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. Accede a la interfaz de Gestión de Usuario.	2. Muestra en el área de trabajo una lista de los usuarios que existen actualmente registrados en el sistema, así como las opciones correspondientes a la Gestión de Usuario: <ul style="list-style-type: none"> <li>▪ Insertar un Usuario.</li> <li>▪ Modificar un Usuario.</li> <li>▪ Mostrar información de un Usuario.</li> <li>▪ Eliminar un Usuario.</li> </ul>
3. Selecciona una de las opciones correspondientes a la Gestión de Usuario.	4. El sistema muestra la interfaz correspondiente a la opción seleccionada. Si selecciona la opción Agregar un Usuario ir a la sección "Agregar un Usuario". Si selecciona la opción Mostrar información de un Usuario ir a la sección "Mostrar información de un Usuario". Si selecciona la opción Modificar un Usuario ir a la sección "Modificar un Usuario". Si selecciona la opción Eliminar un Usuario ir a la sección "Eliminar un Usuario".

#### Sección: "Agregar un Usuario"

#### Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
5. Selecciona la opción correspondiente a Agregar un Usuario.	6. Muestra el formulario requerido para agregar un Usuario.
7. Inserta los datos en el formulario y acepta la operación.	8. Adiciona el Usuario en la Base de Datos y muestra la lista de Usuarios actualizada.

#### Flujo alternativo de los eventos

8. Si los datos insertados no son completos el sistema muestra un mensaje de rectificación y no se realiza la
---

acción de adición.	
<b>Sección: “Mostrar información de un Usuario”</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
5. Selecciona la opción correspondiente a Mostrar información de un Usuario.	6. Muestra una descripción del Usuario seleccionado, y sus datos generales.
7. Acepta la información que brinda el sistema.	8. Se muestra la lista de Usuarios nuevamente.
<b>Sección: “Modificar un Usuario”</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
5. Selecciona la opción correspondiente a Modificar un Usuario.	6. Muestra un formulario con los datos del Usuario seleccionado con la posibilidad de modificar cualquier campo.
7. Modifica los datos deseados en el formulario y acepta la operación.	8. Se actualizan los datos en la Base de Datos y se muestra la lista de Usuarios actualizada.
<b>Flujo alternativo de los eventos</b>	
8. Si los datos a modificar no son completos el sistema muestra un mensaje de rectificación y no se realiza la acción de modificación.	
<b>Sección: “Eliminar un Usuario”</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
5. Selecciona la opción correspondiente a eliminar un Usuario.	6. Elimina el Usuario seleccionado de la base de datos y muestra la lista de Usuarios actualizada.
<b>Postcondiciones:</b>	Un Usuario es adicionado, eliminado, modificado o visualizado.

#### Descripción textual del Caso de Uso Gestionar Activo.

<b>Caso de Uso:</b>	Gestionar Activo.
<b>Actores:</b>	Analista de riesgo
<b>Propósito:</b>	Este caso de uso se realiza con el objetivo de gestionar los activos.
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario desea hacer una modificación en la lista de activos del sistema.
<b>Precondiciones:</b>	El usuario debe haber accedido al sistema con el rol de analista de riesgo.
<b>Referencias:</b>	RF15, RF16, RF17, RF18
<b>Prioridad:</b>	Crítico.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a la interfaz de Gestión de Activo.	2. Muestra en el área de trabajo una lista de los activos que existen actualmente registrados en el sistema, así como las opciones correspondientes a la Gestión de Activo: <ul style="list-style-type: none"> <li>▪ Insertar un Activo.</li> <li>▪ Modificar un Activo.</li> </ul>

	<ul style="list-style-type: none"> <li>▪ Mostrar información de un Activo.</li> <li>▪ Eliminar un Activo.</li> </ul>
3. Selecciona una de las opciones correspondientes a la Gestión de Activo.	<p>4. El sistema muestra la interfaz correspondiente a la opción seleccionada.</p> <p>Si selecciona la opción Agregar un Activo ir a la sección "Agregar un Activo".</p> <p>Si selecciona la opción Mostrar información de un Activo ir a la sección "Mostrar información de un Activo".</p> <p>Si selecciona la opción Modificar un Activo ir a la sección "Modificar un Activo".</p> <p>Si selecciona la opción Eliminar un Activo ir a la sección "Eliminar un Activo".</p>
<b>Sección: "Agregar un Activo"</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
5. Selecciona la opción correspondiente a Agregar un Activo.	6. Muestra el formulario requerido para agregar un Activo.
7. Inserta los datos en el formulario y acepta la operación.	8. Adiciona el Activo en la Base de Datos y muestra la lista de Activos actualizada.
<b>Flujo alternativo de los eventos</b>	
8. Si los datos insertados no son completos el sistema muestra un mensaje de rectificación y no se realiza la acción de adición.	
<b>Sección: "Mostrar información de un Activo"</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
5. Selecciona la opción correspondiente a Mostrar información de un Activo.	6. Muestra una descripción del Activo seleccionado, y sus datos generales.
7. Acepta la información que brinda el sistema.	8. Se muestra la lista de Activos nuevamente.
<b>Sección: "Modificar un Activo"</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
5. Selecciona la opción correspondiente a Modificar un Activo.	6. Muestra un formulario con los datos del Activo seleccionado con la posibilidad de modificar cualquier campo.
7. Modifica los datos deseados en el formulario y acepta la operación.	8. Se actualizan los datos en la Base de Datos y se muestra la lista de Activos actualizada.
<b>Flujo alternativo de los eventos</b>	
8. Si los datos a modificar no son completos el sistema muestra un mensaje de rectificación y no se realiza la acción de modificación.	
<b>Sección: "Eliminar un Activo"</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
5. Selecciona la opción correspondiente a eliminar	6. Elimina el Activo seleccionado de la base de datos

un Activo.	y muestra la lista de <u>Activos</u> actualizada.
<b>Postcondiciones:</b>	Un Activo es adicionado, eliminado, modificado o visualizado.

#### Descripción textual del Caso de Uso Autenticar Usuario.

<b>Caso de Uso:</b>	Autenticar Usuario	
<b>Actores:</b>	Administrador, Analista de Riesgos, Usuario Autorizado	
<b>Propósito:</b>	Este caso de uso se realiza con el objetivo de proteger la información del sistema de usuarios no autorizados a acceder a la misma, y establecer los permisos de uso del sistema para cada usuario registrado según el rol que posee.	
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario desea acceder o salir del sistema.	
<b>Precondiciones:</b>	El usuario debe estar registrado en la base de datos del sistema con uno de los tres roles definidos.	
<b>Referencias:</b>	RF1, RF2	
<b>Prioridad:</b>	Crítico.	
<b>Sección: “Acceder al sistema”</b>		
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. Accede a la interfaz de login para ingresar en el sistema.	2. Muestra un formulario con los campos usuario y contraseña.	
3. Rellena los campos con su información personal de acceso.	4. El sistema comprueba que los datos son correctos y permite el acceso al sistema. Si los datos son incorrectos muestra un mensaje de error permitiendo reintentar la operación.	
<b>Sección: “Salir del sistema”</b>		
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. Selecciona la opción correspondiente a salir del sistema.	2. El sistema des-autentica al usuario y muestra la página de login.	
<b>Postcondiciones:</b>	El usuario es autenticado o des-autenticado respectivamente en el sistema.	

#### Descripción textual del Caso de Uso Calcular Exposición a los riesgos.

<b>Caso de Uso:</b>	Calcular Exposición a los riesgos
<b>Actores:</b>	Analista de riesgo
<b>Propósito:</b>	Este caso de uso se realiza con el objetivo de insertar todos los valores necesarios para calcular la exposición a los riesgos.
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario desea calcular la exposición a los riesgos.
<b>Precondiciones:</b>	El usuario debe haber accedido al sistema con el rol de analista de riesgo.
<b>Referencias:</b>	RF24, RF25, RF26, RF27, RF28, RF29, R30, RF31
<b>Prioridad:</b>	Crítico.

<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a una de las actividades de la fase de evaluación mostradas en el menú vertical: <ul style="list-style-type: none"> <li>• Actividad 1</li> <li>• Actividad 2</li> <li>• Actividad 3</li> <li>• Actividad 4</li> <li>• Actividad 5</li> </ul>	2. El sistema muestra la interfaz correspondiente a la actividad seleccionada.
<b>Sección: “Definir la Fd del Riesgo”</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
3. Accede a la interfaz de la Actividad 1.	4. Muestra en el área de trabajo una lista de los riesgos que existen actualmente registrados en su LPS, así como un selector para escoger los riesgos de un proyecto específico.
5. Selecciona un proyecto.	6. Muestra en el área de trabajo una lista de los riesgos del proyecto seleccionado, así como un selector para escoger el valor de Fd que se desea establecer.
7. Rellena los campos y acepta la información.	8. El sistema actualiza la base de datos y refresca la lista de riesgos con el valor agregado.
<b>Sección: “Definir la Po del Riesgo”</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
3. Accede a la interfaz de la actividad 2.	4. Muestra en el área de trabajo una lista de los riesgos que existen actualmente registrados en su LPS, así como un selector para escoger los riesgos de un proyecto específico.
5. Selecciona un proyecto.	6. Muestra en el área de trabajo una lista de los riesgos del proyecto seleccionado, así como un selector para escoger el valor de Po que se desea establecer.
7. Rellena los campos y acepta la información.	8. El sistema actualiza la base de datos y refresca la lista de riesgos con el valor agregado.
<b>Sección: “Definir el impacto de los activos”</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
3. Accede a la interfaz de la actividad 3.	4. Muestra en el área de trabajo una lista de los activos que existen actualmente registrados en su LPS, y la relación con cada uno de ellos, así como un selector para escoger el valor de impacto que

	se desea establecer.
5. Rellena los campos y acepta la información.	6. El sistema actualiza la base de datos y refresca la lista de riesgos con el valor agregado.
<b>Sección: “Definir el impacto de los riesgos”</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
3. Accede a la interfaz de la actividad 4.	4. Muestra en el área de trabajo una lista de los riesgos que existen actualmente registrados en su LPS, y la relación con cada uno de los activos registrados en la misma, así como un selector para escoger el valor de impacto que se desea establecer.
5. Rellena los campos y acepta la información.	6. El sistema actualiza la base de datos y refresca la lista de riesgos con el valor agregado.
<b>Sección: “Definir los pesos de Fd, Po e Ira respectivamente”</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
3. Accede a la interfaz de la actividad 5.	4. Muestra en el área de trabajo una lista de los riesgos que existen actualmente registrados en su LPS, así como un <i>spinner</i> para escoger el valor de peso de la Fd, peso de la Po y peso del Ira respectivamente que se desea establecer.
5. Rellena los campos y acepta la información.	6. El sistema actualiza la base de datos y refresca la lista de riesgos con el valor agregado.
<b>Postcondiciones:</b>	Se calcula la Exposición a cada riesgo.

#### Descripción textual del Caso de Uso Priorizar Riesgos.

<b>Caso de Uso:</b>	Priorizar Riesgos
<b>Actores:</b>	Analista de Riesgos
<b>Propósito:</b>	Este caso de uso se realiza con el objetivo de establecer la prioridad que puede tener un riesgo para un proyecto en una LPS.
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario desea establecer la prioridad de un riesgo.
<b>Precondiciones:</b>	El usuario debe haber accedido al sistema con el rol de Analista de Riesgos.
<b>Referencias:</b>	RF32, RF33
<b>Prioridad:</b>	Crítico.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a la interfaz de Priorizar Riesgos.	2. Muestra en el área de trabajo una lista de los Riesgos que existen actualmente registrados en el sistema, así como las opciones correspondientes a la Priorización de Riesgos:

	<ul style="list-style-type: none"> <li>▪ Modificar una LPS.</li> <li>▪ Mostrar información de una LPS.</li> </ul>
3. Selecciona una de las opciones correspondientes a la Priorización de Riesgos.	4. El sistema muestra la interfaz correspondiente a la opción seleccionada. Si selecciona la opción Obtener de la exposición a cada riesgo ir a la sección "Obtener exposición". Si selecciona la opción Determinar umbrales ir a la sección "Determinar los umbrales de atención".
<b>Sección: "Obtener exposición"</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
9. Selecciona la opción correspondiente a Obtener exposición.	10. El sistema muestra en el área de trabajo una lista de los Riesgos que existen actualmente registrados en el sistema con su exposición correspondiente.
<b>Sección: "Determinar umbrales"</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
3. Selecciona la opción correspondiente a determinar umbrales de atención.	4. El sistema muestra un formulario para insertar los valores de los umbrales de atención.
5. Acepta la información que agregó al 1 sistema.	6. Se muestra la lista de riesgos nuevamente ahora con los riesgos que están por encima del umbral de atención en rojo.
<b>Postcondiciones:</b>	Se definen los riesgos priorizados.

### Descripción textual del Caso de Uso Obtener Reportes.

<b>Caso de Uso:</b>	Obtener Reportes
<b>Actores:</b>	Analista de Riesgos, Usuario Autorizado
<b>Propósito:</b>	Este caso de uso se realiza con el objetivo de obtener reportes de identificación de riesgos en un proyecto de una LPS del sistema, de Evaluación de Riesgos en un proyecto de una LPS del sistema y de umbrales de atención en un proyecto de una LPS del sistema.
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario desea obtener un determinado reporte.
<b>Precondiciones:</b>	El usuario debe haber accedido al sistema con el rol de Analista de Riesgos o Usuario Autorizado.
<b>Referencias:</b>	RF33, RF34, RF35
<b>Prioridad:</b>	Crítico.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Accede al menú vertical en la opción de reportes y selecciona que reporte desea	2. El sistema muestra la interfaz correspondiente a la opción seleccionada.

<p>obtener:</p> <ul style="list-style-type: none"> <li>• Reportes de identificación de riesgos.</li> <li>• Reportes de Evaluación de Riesgos.</li> <li>• Reportes de umbrales de atención.</li> </ul>	<p>Si selecciona la opción Reportes de identificación de riesgos ir a la sección “Reportes de identificación de riesgos”.</p> <p>Si selecciona la opción Reportes de Evaluación de Riesgos ir a la sección “Reportes de Evaluación de Riesgos”.</p> <p>Si selecciona la opción Reportes de umbrales de atención ir a la sección “Reportes de umbrales de atención”.</p>
<b>Sección: “Reportes de identificación de riesgos”</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
3. Selecciona el reporte identificación de riesgos.	4. El sistema muestra en el área de trabajo una tabla con la información asociada al reporte y las opciones de exportar a PDF y Excel.
5. Selecciona una de las opciones: <ul style="list-style-type: none"> <li>• Exportar a PDF.</li> <li>• Exportar a Exel.</li> </ul>	6. El sistema imprime un documento con la información seleccionada.
<b>Sección: “Reportes de Evaluación de Riesgos”</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
3. Selecciona el reporte de Evaluación de Riesgos.	4. El sistema muestra en el área de trabajo una tabla con la información asociada al reporte y las opciones de exportar a PDF y Excel.
5. Selecciona una de las opciones: <ul style="list-style-type: none"> <li>• Exportar a PDF.</li> <li>• Exportar a Excel.</li> </ul>	6. El sistema imprime un documento con la información seleccionada.
<b>Sección: “Determinar los umbrales de atención”</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
3. Selecciona el reporte de umbrales de atención.	4. El sistema muestra en el área de trabajo una tabla con la información asociada al reporte y las opciones de exportar a PDF y Exel.
5. Selecciona una de las opciones: <ul style="list-style-type: none"> <li>• Exportar a PDF.</li> <li>• Exportar a Excel.</li> </ul>	6. El sistema imprime un documento con la información seleccionada.
<b>Postcondiciones:</b>	Se obtiene el reporte deseado.