

**Universidad de las Ciencias Informáticas**

**“Facultad 6”**



**Título: “Sistema informático para la gestión de riesgos en los proyectos productivos.”**

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

**Autoras: Tania Oropesa Pupo  
Yamilé Rengifo Hardy**

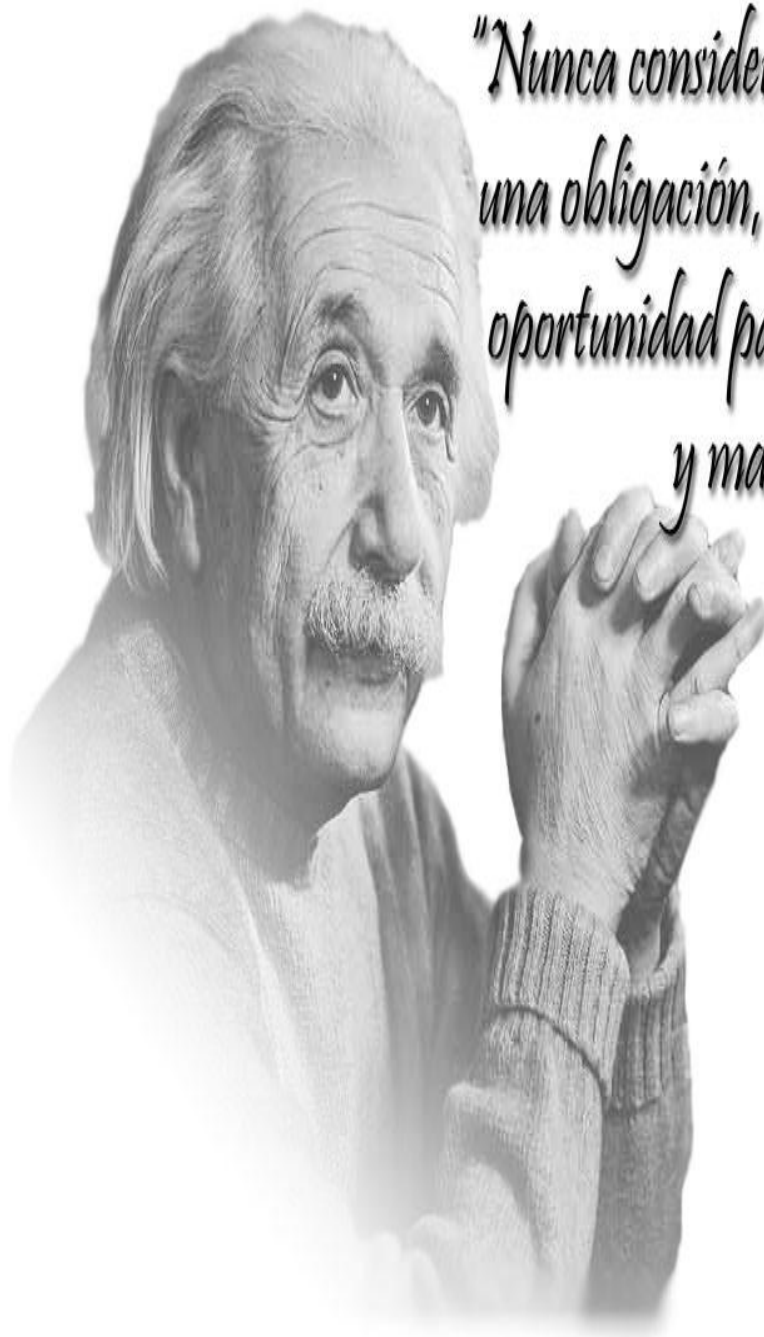
**Tutores: Ing. Lázaro Cánova Amador  
Ing. Carlos Luis Hernández Hernández**

**Co-tutor: Lic. José Alberto Ponce Pérez**

**Ciudad de la Habana, Junio del 2009.**

**“Año del 50 Aniversario del Triunfo de la Revolución.”**

## PENSAMIENTO



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

*Albert Einstein.*

## DECLARACIÓN DE AUDITORÍA

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

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

Yamilé Rengifo Hardy

\_\_\_\_\_

Firma del autor

Tania Oropesa Pupo

\_\_\_\_\_

Firma del autor

Ing. Lázaro Cánova Amador

\_\_\_\_\_

Firma del tutor

Ing. Carlos Luis Hernández Hernández.

\_\_\_\_\_

Firma del tutor

## DATOS DE CONTACTO

### **Tutores:**

Ing. Lázaro Cánova Amador  
Universidad de las Ciencias Informáticas, Habana, Cuba.  
Email: lcanova@uci.cu

Ing. Carlos Luis Hernández Hernández  
Universidad de las Ciencias Informáticas, Habana, Cuba.  
Email: chernandez@uci.cu

### **Co-tutor:**

Lic. Jose Alberto Ponce Pérez  
Universidad de las Ciencias Informáticas, Habana, Cuba.  
Email: japonce@uci.cu

## DEDICATORIA

*A las dos personas que más amo, anhelo y necesito en este mundo, a mi mamita Nancy y a mi hermanita Yanet, por estar siempre a mi lado en las buenas y en las malas.*

*A mis tíos Amauris, Nelvis, Jesús, Martha y Chonchi, los cuales me han brindado su apoyo incondicional y su aliento a seguir luchando y ayudarme a vencer los obstáculos.*

*A Vladimir, por su amor, su paciencia, perseverancia y preocupación en todo.*

*A mis queridos abuelitos, por darme la oportunidad de conocer a esta linda familia que me rodea.*

*A mis queridas amigas de los años, a Yanet, Lissy, Maydalis, Kannia y Lisandra, por confiar en mí.*

*A todas las personitas lindas que conocí en el largo recorrido de estos años.*

*De Yami*

*Especial dedicatoria a mis abuelos por no perder la confianza que depositaron en mi, y porque siempre estuvieron seguros que lograría ser ingeniera.*

*A Taymara, mi hermanita del alma, por ser mi mayor inspiración.*

*De Tany*

## AGRADECIMIENTOS

*Agradezco a todas las personas, que sin su ayuda, no podría estar en estos momentos culminando mis años en la UCI.*

*Agradezco a mis tutores por la preocupación e interés en el desarrollo de nuestra tesis, especialmente a Lázaro, por su apoyo y paciencia.*

*A mi compañera de tesis Tania, por su preocupación y su alegría.*

*A Bárbaro por hacerme reír en los momentos alegres y tristes y por su ayuda.*

*A Eduardito por estar siempre dispuesto en los momentos que más lo necesite*

*De Yami*

*A mi mamá por confiar en mí y darme todo el amor que necesitaba para lograr mi sueño.*

*A mi papá por brindarme seguridad y aliento, y por enseñarme a no rendirme.*

*A mi tío pepe y a mi primo Guillermo, por brindarme todo su apoyo, pues sin su ayuda y confianza no hubiera podido llegar hasta aquí.*

*A mis abuelos por esperar siempre lo mejor de mí.*

*A Yamilé, mi dúo de tesis, por su paciencia en nuestra peleas.*

*A los tutores por ayudarme en la elaboración y construcción de la tesis, en especial a Lázaro por ser tan paciente y dedicado.*

*A Ponce por ayudarnos en la exposición.*

*A Keiler, Yenisel, Leandro y Eduardito por su colaboración.*

*A Liermes por ayudarme siempre que lo necesitaba.*

*A Yumis y Mirielys por ser mis fieles amigas.*

*A Michel por obligarme a ir al laboratorio todas las noches y por su ayuda incondicional.*

*De Tany*

## RESUMEN

El exitoso desarrollo de un software puede verse afectado por la ausencia o incorrecta gestión de los riesgos. Identificar los riesgos, valorar el impacto que pudieran ocasionar de materializarse y la concepción de un plan de contingencia o mitigación, garantizan en gran medida que se minimice la probabilidad de ocurrencia o sus efectos durante el ciclo de desarrollo de un software. Mantener una estrategia proactiva y no reactiva ante los riesgos es uno de los objetivos fundamentales que todo líder de proyecto debería tener en cuenta. El proceso de informatización del país y de la producción de software para la exportación, llevado a cabo por la Universidad de Ciencias Informáticas, no queda exento del impacto indeseable de los riesgos, de ahí la necesidad de realizar una correcta y oportuna gestión de estos en los proyectos productivos. En el presente trabajo de diploma surge teniendo en cuenta investigaciones realizadas en este centro para adaptar el proceso de gestión de riesgos propuesto por la metodología de desarrollo de software Rational Unified Process (RUP), proponiendo la automatización de algunas áreas de este proceso, entre las que se encuentran: identificación, análisis y priorización de los riesgo, así como la planificación, área en la que se obtienen los planes de contingencia.

**PALABRAS CLAVES:** Riesgos, Gestión de Riesgos

## ÍNDICE

<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....</b>	<b>5</b>
Introducción.....	5
1.1    Sistemas informáticos vinculados al campo de acción .....	5
1.2    Principales Definiciones .....	6
1.2.1    ¿Qué es Riesgo? .....	6
1.2.2    Tipos de Riesgos .....	7
1.2.3    ¿Qué es Gestión de Riesgo?.....	9
1.2.4    Estrategias Reactiva y Pro – activa.....	9
1.2.5    Métodos generales de resolución de Riesgos .....	10
1.3    Explicación detallada de cada uno de los Pasos.....	12
1.3.1    Identificación.....	13
1.3.2    Análisis.....	15
1.3.3    Priorización .....	17
1.3.4    Planificación.....	19
1.3.5    Resolución y/o Mitigación.....	21
1.3.6    Monitoreo .....	22
1.4    Metodología de desarrollo.....	23
1.5    Roles y técnicas de los Roles .....	26
1.5.1    Responsabilidades que desempeñan y artefactos que generan .....	26
1.5.2    Roles desempeñados y artefactos que generan por el equipo de trabajo.....	28
1.6    Tecnologías y herramientas en las que se apoya para la solución .....	29
1.6.1    Lenguaje de modelado .....	29
1.6.2    Herramientas de modelado.....	29
1.6.3    Lenguajes de programación .....	31
1.6.4    Entorno de desarrollo .....	34
1.6.5    Framework.....	36
1.6.6    Gestores de base de datos .....	37
Conclusiones .....	39
<b>CAPITULO 2: CARACTERÍSTICAS DEL SISTEMA.....</b>	<b>40</b>
Introducción.....	40
2.1    Modelado del Negocio .....	40
2.1.1    Actores de Negocio.....	40
2.1.2    Trabajador de Negocio .....	40
2.1.3    Diagrama Modelo del Negocio .....	41
2.1.4    Diagrama de actividades.....	41
2.1.5    Modelo de Objeto del Negocio .....	42
2.2    Modelado del Sistema .....	43
2.2.1    Solución informática propuesta .....	43
2.2.2    Definición de los Requerimientos Funcionales .....	43
2.2.3    Definición de los Requerimientos No Funcionales .....	45
2.2.4    Actores del Sistema a Automatizar .....	47
2.2.5    Diagrama de casos de uso del sistema a automatizar.....	47



2.2.6 Descripción de los Casos de Uso del Sistema. Ver Anexo 1 .....	48
Conclusiones .....	48
<b>CAPÍTULO 3: DISEÑO DEL SISTEMA .....</b>	<b>49</b>
Introducción.....	49
2.2 Análisis y Diseño.....	49
2.3 Patrón arquitectónico. Modelo-Vista-Controlador.....	50
2.3.1 Ventajas del MVC .....	51
2.4 Diagrama de Clases de Diseño .....	51
2.4.1 Diagrama de Interacción .....	53
2.4.2 Diagrama de Secuencia .....	53
2.5 Diseño de la Base de Datos .....	55
2.5.1 Modelo lógico de datos .....	56
2.6 Modelo de despliegue .....	56
Conclusiones .....	57
<b>CAPÍTULO 4: IMPLEMENTACIÓN DEL SISTEMA .....</b>	<b>59</b>
Introducción.....	59
4.1 Implementación.....	59
4.2 Diagrama de Componente .....	59
Conclusiones .....	62
<b>CONCLUSIONES .....</b>	<b>63</b>
<b>RECOMENDACIONES .....</b>	<b>64</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>65</b>
<b>BIBLIOGRAFÍA .....</b>	<b>66</b>
<b>Bibliografía.....</b>	<b>66</b>
<b>ANEXOS .....</b>	<b>68</b>
Anexo 1. Descripción textual de los Casos de Uso del Negocio en formato expandido.....	68
Anexo 2. Diagramas de clases del diseño.....	103
Anexo 3 Diagramas de Secuencia .....	108
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>116</b>

## ÍNDICE DE FIGURAS

Figura 1 Procesos para la gestión de riesgos propuestos por RUP .....	12
Figura 2 Proceso de identificación de riesgos .....	13
Figura 3 Proceso de análisis de riesgos .....	15
Figura 4 Proceso de priorización de riesgos.....	17
Figura 5 Proceso de planeación de riesgos.....	19
Figura 6 Proceso de resolución y/o mitigación de riesgos .....	21
Figura 7 Proceso de monitoreo de riesgos .....	22
Figura 8 Diagrama de Modelo de Negocio .....	41
Figura 9 Diagrama de actividades .....	42
Figura 10 Diagrama del Modelo de Objeto del Negocio .....	43
Figura 11 Diagrama de Casos de Uso del Sistema .....	48
Figura 12 Interacción entre los componentes del patrón MVC .....	50
Figura 13 Diagrama de Clases del Diseño CU Identificar riesgos .....	52
Figura 14 Diagrama de Clases del Diseño CU Analizar riesgos.....	52
Figura 15 Diagrama de Clases del Diseño CU Gestionar proyectos .....	52
Figura 16 Diagrama de Clases del Diseño CU Gestionar riesgos .....	53
Figura 17 Diagrama de Secuencia CU Identificar riesgos .....	54
Figura 18 Diagrama de Secuencia CU Analizar riesgos .....	55
Figura 19 Modelo lógico de la base de datos .....	56
Figura 20 Diagrama de Despliegue .....	57
Figura 21 Diagrama de componente CU Autenticar Usuario .....	60
Figura 22 Diagrama de componente CU Gestionar Usuario.....	60
Figura 23 Diagrama de componente Paquete: Reporte.....	61
Figura 24 Diagrama de componente CU Gestión riesgos.....	61
Figura 25 Diagrama de componente Paquete: Nomencladores .....	62

### ÍNDICE DE TABLAS

Tabla 1 Ejemplo de herramientas para la gestión de riesgos .....	6
Tabla 2 Ejemplo de algunos riesgos .....	9
Tabla 3 Probabilidad .....	16
Tabla 4 Consecuencias o impacto .....	16
Tabla 5 Exposición al riesgo o marco de tiempo .....	16
Tabla 6 Equivalencia para la probabilidad.....	18
Tabla 7 Equivalencia para la consecuencia .....	18
Tabla 8 Consecuencias y Probabilidades .....	18
Tabla 9 Relación de actores del negocio. ....	40
Tabla 10 Relación de trabajadores del negocio. ....	41
Tabla 11 Actores del Sistema .....	47

## INTRODUCCIÓN

El continuo desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) han traído consigo una mejora en la calidad de vida del hombre. Es por eso que hoy en día se crean múltiples aplicaciones llevadas a cabo o concebidas como proyectos investigativos en fases iniciales o productivas desde un principio. Estos últimos son los que en su mayoría afrontan mayores peligros debido a los riesgos que poseen. Un riesgo es cualquier acción que pueda influir de manera negativa en la marcha del proyecto en el futuro. Sin embargo los riesgos pueden reducirse o manejarse. Si se es cuidadoso en la relación con el ambiente, y si se está consciente de las debilidades y vulnerabilidades frente a las amenazas existentes, se pueden tomar medidas para asegurarse de que las amenazas no se conviertan en desastres.

Un proyecto productivo nunca está exento de los riesgos, los cuales atentan contra la calidad del producto, la entrega del producto en el tiempo fijado, la obtención de los módulos en el tiempo estimado así como la realización de las tareas propias del ciclo de desarrollo del software y por eso, en este contexto de la gestión de proyecto, muchas empresas reconocieron la necesidad de integrar los riesgos de carácter técnico con los de coste, planificación o calidad. A partir de allí se desarrollaron metodologías integradas de gestión de riesgos. En el contexto de un proyecto, es el arte y ciencia de identificar, analizar y responder a los factores de riesgo a lo largo de la vida del proyecto y en el mejor cumplimiento de sus objetivos.

Cuando se pone mucho en un proyecto de software el sentido común aconseja realizar un análisis de los riesgos. Y sin embargo, la mayoría de los jefes de proyectos lo hacen de manera informal, si es que lo hacen. El tiempo invertido identificando, analizando y gestionando el riesgo merece la pena por muchas razones: menos trastornos durante el proyecto, una mayor habilidad de seguir y controlar y la confianza que da planificar los problemas antes de que ocurran. Para tratar de minimizar los efectos de un problema de seguridad se realiza lo que se denomina un Análisis de Riesgos: es un proceso por el cual se identifican las amenazas y vulnerabilidades de una organización o institución, con el fin de generar controles que minimicen los efectos de los riesgos.

Cuba ha dado un gran impulso a la producción de software nacional y por eso no está ajena a las afectaciones que acarrearán los riesgos en cualquier proyecto que se

acometa. Una de las instituciones líderes de la industria del software en Cuba es la Universidad de las Ciencias Informáticas (UCI), en la cual se llevan a cabo diferentes proyectos productivos especializados por esferas, y en dependencia de su complejidad y perfil pueden propiciar la existencia de ciertos riesgos en su proceso de desarrollo, algunos son comunes a todos estos procesos y otros van más específicos a la naturaleza y concepción del proyecto. Los proyectos que se acometen tienen ciertas particularidades que influyen en las vulnerabilidades propias de estos para que varios riesgos impidan su desarrollo satisfactorio. Tal es el caso que se han detectado problemas que impiden que algunos de los proyectos se realicen en el tiempo requerido y que puedan llegar a lograr el objetivo final. Uno de estos problemas son los riesgos existentes, ya sean informáticos, organizativos o materiales, un ejemplo de estos es la pérdida de información, en la cual entran a jugar los factores de la tecnología, actualización incorrecta o retrasada del antivirus y falta de la tecnología necesaria para tener la información segura guardada en varias computadoras.

Actualmente en los proyectos se conocen de cierta forma, un poco informalmente los riesgos que podrían afectar el desarrollo, pero no se guían por una metodología formal de gestión de riesgos y mucho menos se procede a su análisis o gestión. Por lo que no se tiene medida del nivel de riesgo del proyecto y ocasiona incumplimientos en el término de los objetivos trazados y aumento del costo del mismo.

Además, en la presente institución no existe una vasta cultura referente a este tema, ni un procedimiento para administrar o prever los riesgos durante todo el ciclo de vida del software. Por ende, se ve en la necesidad de realizar un procedimiento para llevar a cabo la gestión de riesgo en cada uno de sus proyectos productivos e ir buscando otras alternativas para garantizar la formación del personal vinculado al desarrollo de la producción de software.

Actualmente en la universidad se lleva a cabo el proceso de planificación, ejecución y control de la gestión de riesgos de forma manual. El gran volumen de información y la variedad de actores que intervienen en la gestión de riesgos hacen que este proceso se torne engorroso, lento y poco confiable; haciendo sumamente difícil el proceso de toma de decisiones con vista a mejorar el desarrollo de los proyectos productivos.

Teniendo en cuenta todo lo antes analizado surge como **problema científico**: ¿Cómo contribuir a la gestión de los riesgos que afectan a los proyectos productivos de una facultad de la Universidad de las Ciencias Informáticas?

Definiéndose como **objeto de estudio**: El proceso de gestión de riesgos en los proyectos productivos, enmarcándose el **campo de acción** en: El proceso de gestión de riesgos propuesto por RUP, aplicados a los proyectos productivos de la Universidad de Ciencias Informáticas. Para dar solución al problema planteado se propone como **objetivo general**: Desarrollar un sistema informático que gestione los riesgos en los proyectos productivos de una facultad de la Universidad de las Ciencias Informáticas.

Para alcanzar el objetivo general se trazaron los siguientes **objetivos específicos**:

- Analizar el sistema informático.
- Diseñar el sistema informático.
- Implementar la propuesta de solución diseñada.

Para dar cumplimiento a los objetivos propuestos se definieron las siguientes **tareas investigativas**:

- El estudio de la situación actual de la gestión de riesgos en proyectos informáticos en el mundo y particularmente en Cuba.
- El estudio del proceso de gestión de riesgos en proyectos informáticos propuesto por RUP.
- El estudio de las modificaciones propuestas a la gestión de riesgos establecida por RUP, adaptada a la producción de software en la UCI, realizada en trabajos de diploma precedentes.
- La selección de la metodología de desarrollo de software a utilizar.
- La selección de las tecnologías y herramientas para llevar a cabo la implementación.
- La modelación del negocio.
- La modelación del sistema.
- El diseño de la solución propuesta.
- La implementación del sistema propuesto.

El presente trabajo de diploma está estructurado en cuatro capítulos que se describen brevemente a continuación:

**Capítulo 1 “Fundamentación Teórica”.** En este capítulo se describe de forma general el grado de automatización de la gestión de riesgo en el mundo y en Cuba, particularizando en la Universidad de las Ciencias Informáticas. Se hace un estudio de

las metodologías, lenguajes y herramientas a utilizar en el desarrollo de la solución propuesta, basada en la automatización de alguna de las áreas de la gestión de riesgos propuesta por la metodología RUP.

**Capítulo 2 “Características del Sistema”.** En este capítulo se realiza la modelación del negocio, se identifican los requisitos funcionales y no funcionales del sistema propuesto y se realiza la modelación del mismo, generándose los artefactos correspondientes a estos flujos de trabajo según la metodología RUP.

**Capítulo 3 “Análisis y Diseño del Sistema”.** Teniendo como base los artefactos generados durante la modelación del negocio y el sistema, en este capítulo se muestra el diseño de la solución propuesta según la metodología, el lenguaje de programación, el framework de desarrollo y la arquitectura seleccionadas. Se presenta además el diagrama de despliegue que muestra la distribución física del sistema para su futura implantación.

**Capítulo 4 “Implementación”.** En este capítulo se presentan los artefactos generados durante el flujo de trabajo de implementación entre los que se encuentran los diagramas de componentes que muestran la relación de dependencia entre los ficheros de la aplicación.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

### Introducción

El presente capítulo ofrece información sobre el estado del arte del proceso de gestión de riesgos en los proyectos de software tanto a nivel mundial como nacional; se realizará un breve análisis acerca del objeto de estudio de la investigación; se explicará brevemente los pasos para la gestión de riesgos que propone RUP; así como las metodologías de desarrollo, tecnologías y herramientas utilizadas para dar solución al trabajo. Incluye además una valoración de las responsabilidades que se desempeñan y los artefactos que realizan los roles genéricos de Analista y Desarrollador definidos por el Proceso Unificado de Desarrollo (RUP).

### 1.1 Sistemas informáticos vinculados al campo de acción

En el presente trabajo, las autoras para dar respuesta a una de las tareas planteadas realizaron una revisión bibliográfica, sobre la existencia de algún sistema informático que permitiera la gestión de riesgos en los proyectos productivos de software. La búsqueda arrojó como resultado de que existen una gran cantidad de herramientas software de gestión de riesgos disponibles en el mercado y que siguen algunas metodologías. Estas herramientas se enfocan sólo en una categoría de riesgos (TRIMS – Technical Risk Identification and Mitigation System), o están orientadas a compañías maduras que poseen una amplia base de datos organizacional que les permite generar información de categorías propias de riesgos (RiskTrak y WelcomRisk), o bien emplean un mecanismo que no se orienta al uso de taxonomías (ARM – Active Risk Manager) (1). En la siguiente tabla se describe las características de algunas herramientas (1). Pero en la búsqueda bibliográfica realizada no se encontró ningún sistema automatizado, referente al desarrollo de proyectos productivos de software, que manejara esta información para una universidad o para diferentes instituciones desarrolladoras de software a nivel nacional e internacional, por lo cual se hace la necesidad de desarrollar una aplicación que resuelva el problema científico planteado.

Producto	Proveedor	Descripción	Plataforma
Active Risk	Strategic	Herramienta integrada de gestión	Web Based



Manager (ARM)	Thought	de riesgos que brinda una solución para la identificación de riesgos mediante la utilización de la información contenida en el WBS de proyecto.	
Technical Risk Identification and Mitigation System (TRIMS)	Best Manufacturing Practices	Herramienta integrada de gestión de riesgos que emplea ingeniería de conocimientos y que se enfoca en la identificación y medición de riesgos técnicos de proyectos.	Win32
RiskTrak	Risk Services & Technology	Herramienta integrada de gestión de riesgos que brinda una solución para la identificación de riesgos mediante el empleo de bases de datos.	Win32
WelcomRisk	Welcom	Herramienta integrada de gestión de riesgos que brinda una solución para la identificación sistemática de riesgos mediante la utilización de bibliotecas configurables de categorías de riesgos.	Win32
Chinchón – Análisis del riesgo	Free	Chinchón es una herramienta para analizar cuantitativamente el riesgo de un sistema (de información). La herramienta sigue el modelo Magerit 1.0	Java

Tabla 1 Ejemplo de herramientas para la gestión de riesgos

## 1.2 Principales Definiciones

### 1.2.1 ¿Qué es Riesgo?

Según estudios realizados, se define el Riesgo como: Un riesgo es cualquier suceso que pueda afectar negativamente a la marcha del proyecto en el futuro. El riesgo se haya asociado de manera inexorable a cualquier actividad que se lleve a cabo y que imponga la decisión entre varias alternativas. El riesgo, por tanto,

acompaña a todo cambio y está presente en cada decisión. El riesgo implica elección e incertidumbre (1).

## 1.2.2 Tipos de Riesgos

En la actualidad, han existido diferentes criterios acerca de la definición adecuada para el riesgo de software, pero se concuerda que él involucra dos características:

1. Incertidumbre: El acontecimiento que caracteriza al riesgo puede o no puede ocurrir; por ejemplo, no hay riesgos de un 100% de probabilidad (ya que un riesgo del 100% es una limitación del proyecto) (2).
2. Pérdida: Si el riesgo se convierte en una realidad, ocurrirán consecuencias no deseadas o pérdidas (2).

Existen diferentes tipos de riesgos de software, entre los principales se encuentran:

1. Riesgo del proyecto: Los riesgos del proyecto amenazan al plan del proyecto de software, lo que significa que si estos se hacen realidad, probablemente la planificación del proyecto se retrase y los costos aumenten. Los riesgos del proyecto identifican:

- Problemas potenciales de presupuesto
- Planificación temporal
- Personas (Asignación y organización)
- Recursos
- Cliente, requerimientos y su impacto en el proyecto software
- La complejidad del proyecto
- Tamaño y grado de incertidumbre estructural

2. Riesgos técnicos: Los riesgos técnicos constituyen una amenaza a la calidad y planificación temporal del software que hay que producir. Si un riesgo llega a convertirse en realidad, la implementación puede llegar a ser difícil o imposible (2).

Los riesgos técnicos identifican problemas potenciales de:

- Diseño
- Implementación
- De interfaz
- Verificación
- Mantenimiento
- Las ambigüedades de especificaciones

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Tecnologías

3. Riesgos del Negocio: Una de las principales amenazas de los riesgos de negocio es que afectan parcial o totalmente la viabilidad del software a construir y a menudo ponen en peligro el proyecto o el producto. Los principales riesgos del negocio son:

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

A continuación se muestra algunos ejemplos de riesgos (1), que anteriormente se explicaron:

Riesgo	Tipo	Descripción
Cambio de equipo de trabajo	Proyecto	Las personas con más experiencia en el equipo dejan su trabajo antes de la terminación.
Cambio de Gestión	Proyecto	Hay un cambio de la dirección del proyecto con otras prioridades diferentes
Indisponibilidad del hardware	Proyecto	El hardware necesario para el proyecto no se entrega a tiempo.
Cambio de los requerimientos	Proyecto y producto	Hay un gran número de cambios en las especificaciones iniciales que no estaba previsto
Retrasos en las especificaciones	Proyecto y producto	Algunas especificaciones de los interfaces principales entre módulos no se terminan de acuerdo a la planificación.
Subestimación del tamaño	Proyecto y producto	El tamaño del sistema está

		subestimado
Insuficiencia de las herramientas	Producto	Las herramientas elegidas para el desarrollo son insuficientes y no tienen las facilidades con las que se contaba en principio.
Cambios de la tecnología	Negocio	La tecnología en la que se ha basado el proyecto es sustituida por una tecnología más moderna.
Competencia de productos	Negocio	Un producto de la competencia es puesto en el mercado antes de terminar el proyecto.

**Tabla 2 Ejemplo de algunos riesgos**

Para evitar que se materialice alguno de los riesgos anteriormente categorizados, es de vital importancia que el equipo de proyecto realice una correcta gestión de riesgos, de modo que pueda minimizar su ocurrencia o eliminar al máximo los efectos producidos por la presencia de alguno de estos.

### 1.2.3 ¿Qué es Gestión de Riesgo?

Los objetivos de la gestión del riesgo son identificar, controlar y eliminar las fuentes de riesgos antes de que empiece a afectar el cumplimiento de los objetivos del proyecto (3).

Muchos son los conceptos asociados a la gestión de riesgos y su importancia. Desde el punto de vista de la producción de software se pudiera definir como el proceso mediante el cual pueden identificarse, previamente, los problemas que pueden afectar la producción de un sistema informático durante su ciclo de desarrollo, y trazar las estrategias necesarias para prevenirlos o minimizar su impacto.

### 1.2.4 Estrategias Reactiva y Pro – activa

Existen dos estrategias fundamentales que ayudan de una forma u otra a tratar de prevenir que éste ocurra. Estas estrategias son Reactiva y Pro – activa.

1. Estrategias reactivas:

- Las estrategias reactivas son aquellas que se dan cuando se deja que los riesgos produzcan sus efectos (en este momento ya no es un riesgo, es una realidad) y entonces se actúa en consecuencia.
- En estas condiciones lo único que cabe es tomar medidas correctoras (apagar incendios), lo que produce muchos tiempos perdidos, retrasos en el proyecto, gabinetes de crisis, etc.
- Las estrategias reactivas no son aconsejables porque ponen en grave peligro el proyecto.

## 2. Estrategias proactivas:

- Las estrategias proactivas pasan por la evaluación previa y sistemática de todos los riesgos inherentes al proyecto, evaluando sus consecuencias.
- Esto produce la creación de un Plan de Gestión de Riesgos, con sus planes de evitación, minimización de consecuencias, planes de contingencia, etc.

Las estrategias de gestión de riesgos, en muchas ocasiones, son valoradas como los profesionales de la informática, donde se cree que el mejor es aquel que más problemas resuelve y no aquel que más problemas evita. Entre las dos estrategias planteadas, la proactiva es la más recomendada, pues una vez identificados los riesgos, se toman todas las medidas para evitar su ocurrencia y los efectos indeseables que estos pudieran acarrear.

### 1.2.5 Métodos generales de resolución de Riesgos

Para contrarrestar los riesgos existen cuatro métodos de resolución de Riesgos, ellos actúan según la actitud que se muestre al frente al Riesgo. Estos cuatro métodos son: eliminación, retención, evitación y transferencia (4).

- 1 Eliminación del Riesgo: Se trata de eliminar los factores que inducen el riesgo y con ello eliminar la posibilidad de exposición.
- 2 Retención del Riesgo: Muchos de los Riesgos que se identifican en el proyecto, no es posible eliminarlos y por tanto hay que desarrollarle otro tipo de tratamiento. La retención, es la asunción por parte de los responsables del proyecto, de informar la existencia de un riesgo. El mero hecho de acometer un proyecto, es un riesgo que no puede ser evitado y sólo se debe retener.
- 3 Evitación del Riesgo: En estos casos hay que identificar los factores que provocan el riesgo y mantenerlos bajo control, para evitar que el mismo provoque sus efectos. Evitar es diferente de eliminar.

En caso del que el riesgo amenazará permanentemente el proyecto, se debe evitar sus efectos.

- 4 Transferencias del Riesgo: Algunos tipos de riesgos (normalmente poco probables pero muy negativos) pueden ser transferidos a terceros, mediante la contratación de seguros o haciendo contratos en los que el cliente o los proveedores asumen este riesgo y liberan al equipo de proyecto de su gestión.

Para la puesta en práctica de estos métodos de resolución de riesgos se necesita tener bien definido un modelo, que guíe a través de un grupo de pasos lógicos la gestión de los riesgos. Muchos son los modelos propuestos, desde la década de los 80 y hasta la actualidad. Según la bibliografía consultada y teniendo como base trabajos de diploma realizados en la Universidad de las Ciencias Informáticas, se asume un modelo que integra los pasos o procesos comunes en la mayoría de modelos propuestos para la gestión de riesgos, y que surge como una adaptación de la propuesta que hace la metodología de desarrollo Rational Unified Process para este tipo de gestión. A continuación se muestran estos en forma de diagrama de actividad.

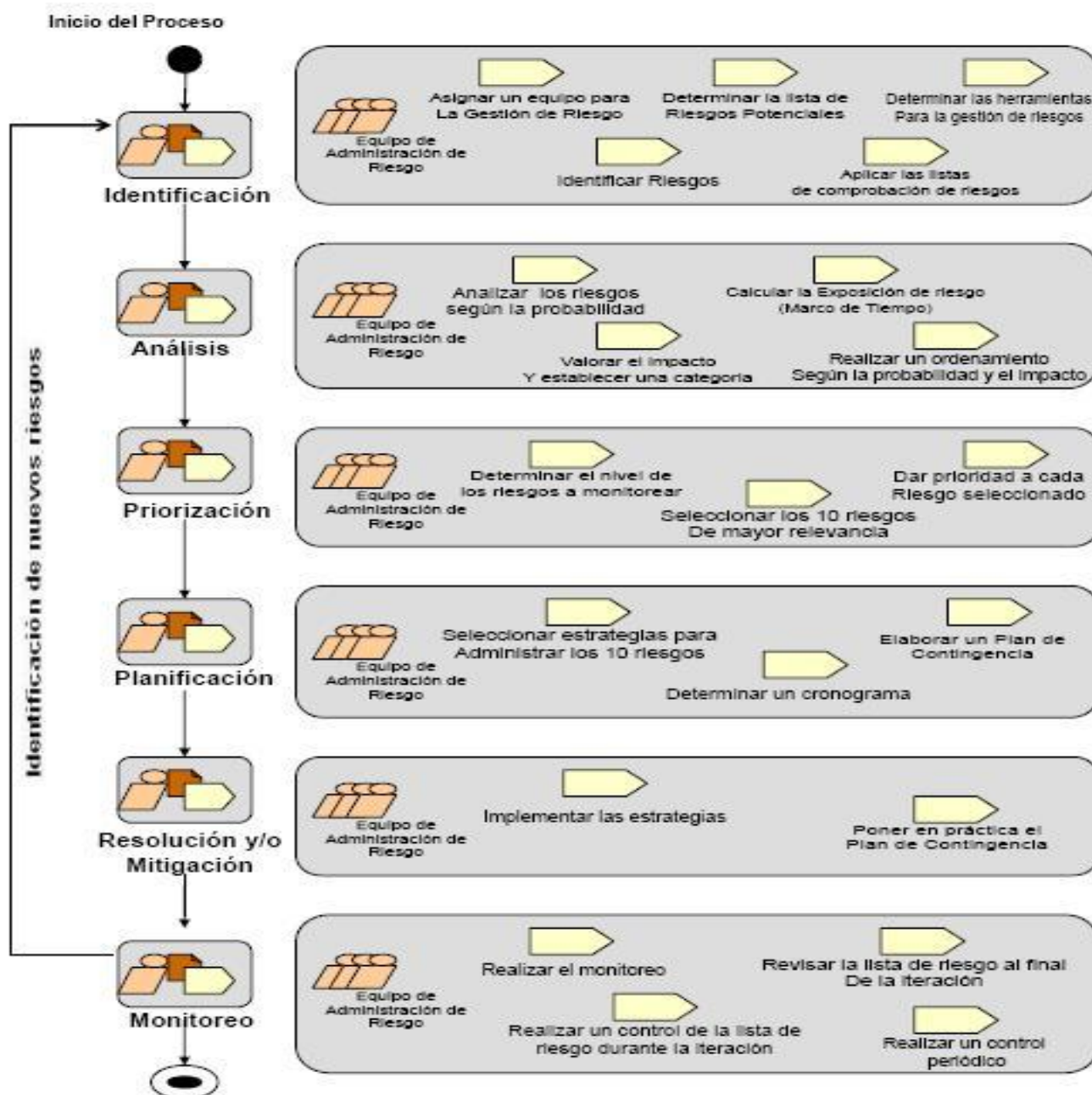


Figura 1 Procesos para la gestión de riesgos propuestos por RUP

## 1.3 Explicación detallada de cada uno de los Pasos

El proceso se basa básicamente en la identificación, análisis, priorización, planificación, resolución y/o mitigación y monitoreo de los riesgos en los proyectos productivos. En este trabajo de diploma se propone, para contribuir a la gestión de riesgos en los proyectos productivos implementar un grupo de actividades de los cuatro primeros pasos de este modelo propuesto.

## 1.3.1 Identificación

La identificación es uno de los flujos más importantes dentro de la Gestión de Riesgo. Consiste en la determinación de los riesgos potenciales, mediante la utilización de métodos o técnicas. Posibilitando la implementación y desarrollo anticipado de los problemas que pueden surgir en el proyecto (4).

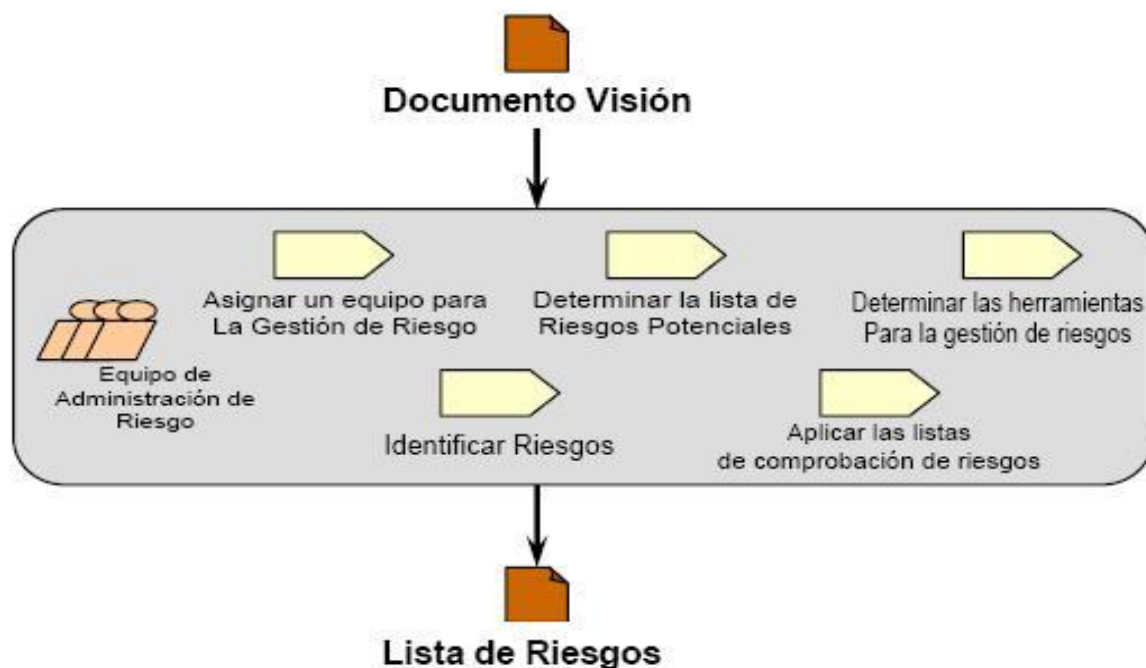


Figura 2 Proceso de identificación de riesgos

En este primer paso se pretende automatizar las actividades Identificar riesgos y determinar la lista de riesgos potenciales, el cual es realizado por un equipo de desarrollo definido.

### ➤ Asignar un equipo para la Gestión de Riesgos

En esta actividad se realiza la creación del equipo para la Gestión de Riesgo. Para conformar este equipo, se debe tener en cuenta el representante por parte de los clientes, el arquitecto de software y el líder de los equipos de prueba, desarrollo, documentación y despliegue, los que se encargarán de transmitir y analizar con todo su equipo los posibles riesgos en su proyecto. Inmediatamente se procede a la selección del líder de este equipo, que será el responsable de reunir al personal para realizar la identificación, el reporte y monitoreo de los riesgos (4).

### ➤ Identificar los riesgos



La identificación del riesgo consiste en un proceso de reflexión a través de diferentes técnicas para determinar los factores que representan un riesgo para el proyecto y constituye, al identificarlos, un criterio pro\_activo.

## ➤ **Determinar la lista de riesgos potenciales**

Se basa en la realización de una lista, la cual contiene los riesgos más importantes del proyecto. Esta lista se debe actualizar en cada fase, para detectar si se han incorporado nuevos riesgos. Las técnicas a aplicar para la realización de los dos primeros pasos son (4):

- 1 Discusiones de grupo o entrevistas
- 2 Experiencia en el personal
- 3 Tormenta de idea (*Brainstorming*)
- 4 Técnicas Delphi

## ➤ **Aplicar las listas de comprobación de riesgos**

Actividad que define un conjunto de cuestiones que son relevantes para cada factor de riesgo, teniendo en cuenta la categoría de los mismos.

- 1 Riesgos específicos del producto: Son aquellos que se asocian a un producto en particular de forma única y específica. Se identifican realizando un examen del plan del proyecto y la declaración del ámbito del software.
- 2 Riesgos genéricos: Son una amenaza potencial para todos los proyectos del software.
  - Tamaño del producto
  - Impacto de negocio
  - Características del cliente
  - Definición del proceso
  - Entorno de desarrollo
  - Tecnología a construir

## ➤ **Determinar las herramientas para la gestión de riesgos**

Para esta actividad se exponen algunas series de herramientas que pueden ser utilizadas para la identificación de riesgos. Estos son:

- Active Risk Manager (ARM)
- Technical Risk and Mitigation

- System (TRIMS)
- Risk Trak
- Welcome Risk

Artefacto de entrada:

1. Documento Visión.

Artefacto de salida:

1. Lista de riesgos.

## 1.3.2 Análisis

En este proceso se examinan los riesgos con un nivel más detallado, para determinar sus extensiones, interrelaciones e importancia.

Además determinan sus atributos:

- Impacto o Consecuencia
- Probabilidad
- Exposición al riesgo (Marco de Tiempo)



Figura 3 Proceso de análisis de riesgos

### ➤ Analizar los riesgos según la probabilidad

En la presente actividad se realiza un análisis detallado y una clasificación de los riesgos según la probabilidad de ocurrencia. La técnica cuantitativa incluye el análisis de la probabilidad, a esta probabilidad se le asigna un parámetro mayor que cero y

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

menor e igual que el cien por ciento. Siendo cero la certeza absoluta de no ocurrencia del suceso y uno cuando se presenta la certeza absoluta de ocurrencia (4).

Atributo	Valor	Descripción
Probabilidad	Muy baja	Menor del 10%
	Baja	Del 10% al 22%
	Moderada	Del 25% al 75%
	Alta	Del 75% al 90%
	Muy alta	Mayor del 90%

Tabla 3 Probabilidad

### ➤ Valorar el Impacto y establecer una categoría

Esta actividad se basa específicamente en valorar el impacto que puede producir un riesgo en el desarrollo de un proyecto. La técnica a utilizar es el Análisis de Consecuencias, incluida en el método al cual se hizo alusión en el paso anterior.

Atributo	Valor	Descripción
Consecuencia o Impacto	Catastrófico	Pérdida del sistema. Costo mayor del 50%
	Crítico	Recuperación de la capacidad operativa. Costo mayor del 10% y menor del 20%
	Tolerable	Capacidad operativa mermada. Costo mayor del 10% y menor del 20%

Tabla 4 Consecuencias o impacto

### ➤ Calcular la Exposición de riesgo (marco de tiempo)

La exposición del riesgo es el periodo de tiempo en que es posible mitigar el riesgo (1). Se calcula mediante la multiplicación de la probabilidad por el impacto. Los valores de la Exposición al riesgo en un proyecto se muestran a modo de resumen en la siguiente:

Atributo	Valor	Descripción
Exposición al riesgo O Marco de tiempo	Improbable	Menor que 30 días
	Corto Plazo	30 días
	Mediano Plazo	De 1 a 4 meses.
	Largo Plazo	Más de 4 meses.

Tabla 5 Exposición al riesgo o marco de tiempo

## ➤ Realizar un ordenamiento según la Probabilidad y el Impacto

En la presente actividad se realiza un ordenamiento de los riesgos analizados, teniendo en cuenta la probabilidad de aparición en el tiempo y el impacto en el proyecto.

Artefacto de entrada:

1. Listas de riesgos

Artefacto de salida:

1. Caracterización de los riesgos

### 1.3.3 Priorización

En este proceso se toma la decisión de cuáles son los 10 riesgos que tiene mayor relevancia en el proyecto, y se realiza una clasificación para darle una prioridad a cada uno de ellos (4).

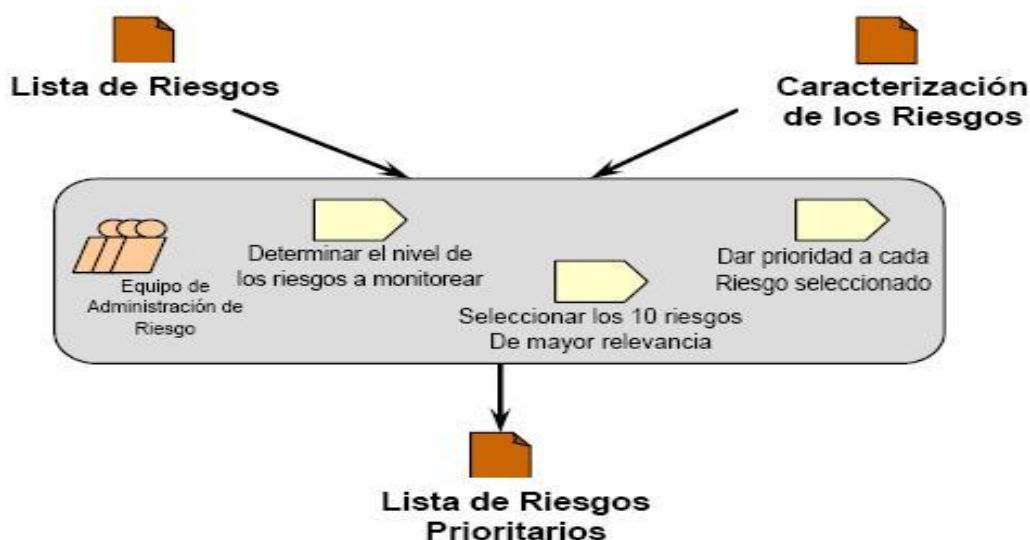


Figura 4 Proceso de priorización de riesgos

## ➤ Determinar el nivel de los riesgos a monitorear

El nivel de cada riesgo analizado se calcula, mediante la relación entre la Probabilidad y la Consecuencia. Teniendo en cuenta que los atributos de las tablas de equivalencia 6 y 7 que se muestran a continuación, no coinciden con los que se hacen alusión en el flujo anterior, en trabajos realizados decidieron realizar un equivalente entre los atributos de ambas tablas.

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Casi Improbable	↔	Muy Baja
Poco Probable	↔	Baja
Moderado	↔	Moderado
Probable	↔	Alta
Casi Cierta	↔	Muy Alta

**Tabla 6 Equivalencia para la probabilidad**

Extrema	↔	Catastrófico
Muy Alto	↔	
Medio	↔	Critico
Bajo	↔	Tolerable
Mínimo	↔	

**Tabla 7 Equivalencia para la consecuencia**

Para una mayor organización se aconseja consultar la tabla que viene a continuación:

		Consecuencias →				
Probabilidad ↓		<b>Extremo</b>	<b>Muy Alto</b>	<b>Medio</b>	<b>Bajo</b>	<b>Mínimo</b>
	<b>Casi Cierta</b>	Alto	Alto	Alto	Importante	Importante
	<b>Probable</b>	Alto	Alto	Importante	Importante	Significativo
	<b>Moderado</b>	Alto	Alto	Importante	Significativo	Bajo
	<b>Poco Probable</b>	Alto	Importante	Significativo	Bajo	Bajo
	<b>Casi Improbable</b>	Importante	Importante	Significativo	Bajo	Bajo

**Tabla 8 Consecuencias y Probabilidades**

- Riesgo Alto: Se requiere una investigación detallada y una planificación a niveles superiores.
- Riesgo Importante: Se requiere una atención del personal superior.
- Riesgo Significativo: Se debe especificar la responsabilidad de gestión.
- Riesgo Bajo: Se maneja mediante procedimientos de rutina.

## ➤ Seleccionar los 10 riesgos de mayor relevancia

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Se selecciona por parte del equipo, los 10 riesgos que tienen mayor relevancia para el proyecto.

## ➤ Dar prioridad a cada riesgo

Se desarrolla una nueva lista con los 10 riesgos más frecuentes que fueron aceptados en el paso anterior, otorgándole a cada uno su prioridad. Se debe tener en cuenta el nivel que le fue otorgado a cada uno (4).

Artefacto de entrada:

1. La lista de riesgos
2. La caracterización de los riesgos

Artefacto de salida:

1. Lista de riesgos prioritarios

### 1.3.4 Planificación

La planificación de gestión de riesgos ayuda a manejar cada elemento de riesgo, incluyendo la coordinación de los planes individuales de elementos de riesgos entre ellos y con respecto al plan general (1). Su objetivo es convertir la información sobre los riesgos en decisiones y acciones para el presente y el futuro.

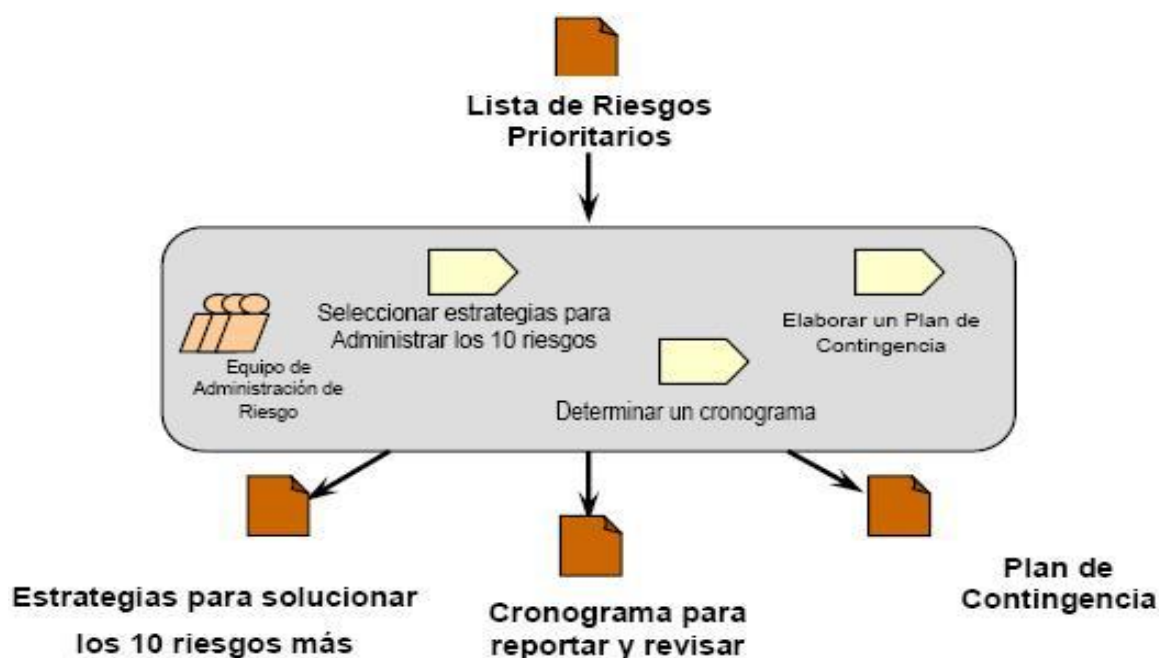


Figura 5 Proceso de planeación de riesgos

## ➤ Seleccionar estrategias para administrar los 10 riesgos

Teniendo en cuentas las siguientes estrategias, se hace una selección de aquella que sea más factible para administrar los 10 riesgos más usuales en el proyecto (4):

- “Evitación”: Se trata de minimizar la probabilidad de que el riesgo se presente.
- Minimización: Reducir el impacto del riesgo en el producto o en el proyecto.
- Planes de Contingencia: Este plan se pone en práctica en caso de que fallen las anteriores estrategias. Se encarga de administrar el riesgo que está vigente en ese momento.

➤ **Determinar un cronograma**

Se realiza un cronograma, para el equipo de desarrollo, con el objetivo de realizar los reportes y las revisiones en tiempo y con la calidad que requerida. De esta manera, se logra acelerar el proceso para eliminarlos.

➤ **Elaborar el Plan de Contingencia**

Incluye, la descripción del riesgo, los indicadores que muestran que el mismo se ha hecho realidad y las acciones a llevar a cabo, para eliminar los riesgos que están afectando el proyecto en ese momento. Las posibles técnicas que se pueden aplicar en este paso son (4):

- 1 Costo – Beneficio: Es el balance entre el costo de la implementación de las medidas preventivas y el riesgo remanente. Además tiene en cuenta la posibilidad de ocurrencia de daños materiales, tanto a la instalación, como a la pérdida de la producción durante los períodos de parada y la reparación de los daños.
- 2 Las acciones para disminuir el riesgo:
  - Eliminación del riesgo: Se elige una alternativa de menor riesgo.
  - Retención del riesgo: Se decide mitigar el riesgo.
  - Evitación del riesgo: Se acepta que el riesgo ocurra.
  - Transferencias del riesgo: Se reduce el riesgo compartiéndolo.

Artefacto de entrada:

1. Lista de riesgos prioritarios.

Artefacto de salida:

1. Estrategias para solucionar los 10 riesgos más frecuentes.
2. Cronograma para reportar y revisar riesgos.
3. Plan de Contingencia.

## 1.3.5 Resolución y/o Mitigación

Este flujo tiene la tarea de implementar las estrategias que se tomaron en el anterior proceso y pone en práctica los planes de contingencia.

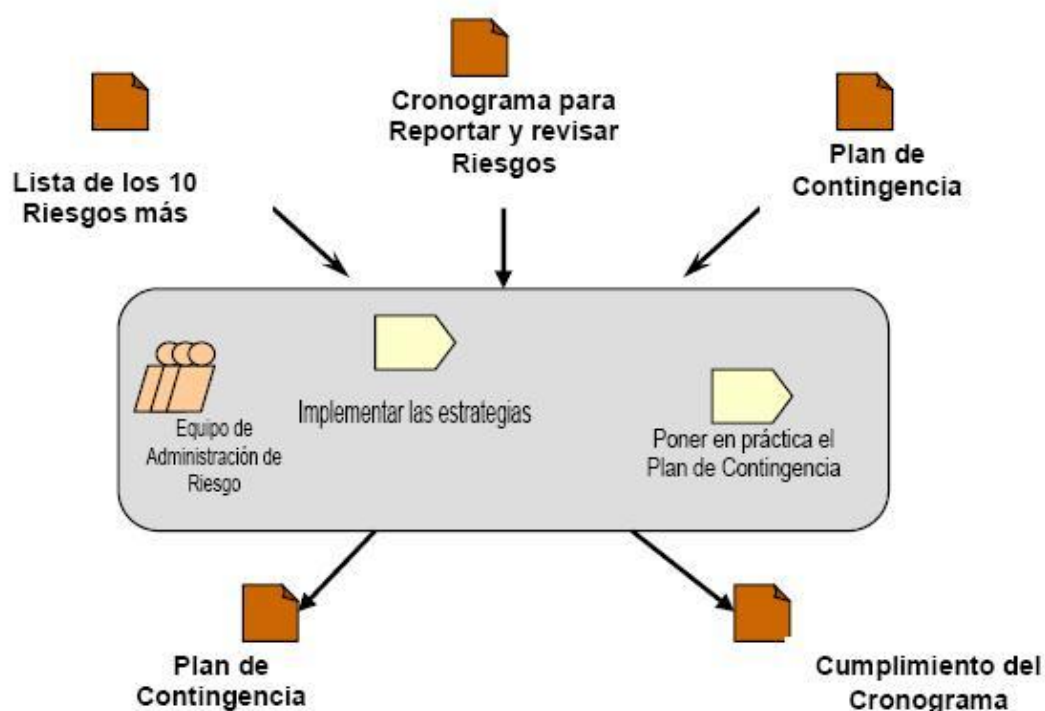


Figura 6 Proceso de resolución y/o mitigación de riesgos

### ➤ Implementar las estrategias

Se ponen en práctica las estrategias que se definieron en el paso de planificación. La mitigación define un conjunto de técnicas, que agilizan este proceso, como son:

- 1 Poda de requisitos: Se realiza una selección de los requisitos más importantes, en caso que se tenga que realizar una primera versión del producto.
- 2 Prototipado: Se brinda una breve panorámica acerca de las acciones que se llevarán a cabo para mitigar los riesgos.
- 3 Desarrollo incremental: Consiste en la utilización de incrementos para aumentar gradualmente el alcance, empezando por las posibilidades más básicas de las aplicaciones y ampliándolas paso a paso.

### ➤ Poner en práctica el Plan de Contingencia



Se refina y pone en práctica el plan de contingencia, elaborado en el flujo de trabajo anterior.

Artefactos de entrada:

1. Lista de los 10 riesgos más frecuentes.
2. Cronograma para reportar y revisar riesgos.
3. Plan de Contingencia.

Artefacto de Salida:

1. Plan de contingencia aplicado.
2. Cumplimiento del cronograma.

## 1.3.6 Monitoreo

La monitorización del riesgo consiste en controlar el progreso del proyecto en lo relativo a resolución de riesgos, tomando las acciones correctoras cuando sea necesario. Es una etapa esencial e integral en el proceso de Gestión de Riesgo, porque es donde se realiza un monitoreo cada cierto tiempo de los riesgos que están presentes y de los que se incluirán a medida que vaya avanzando el proyecto (4).

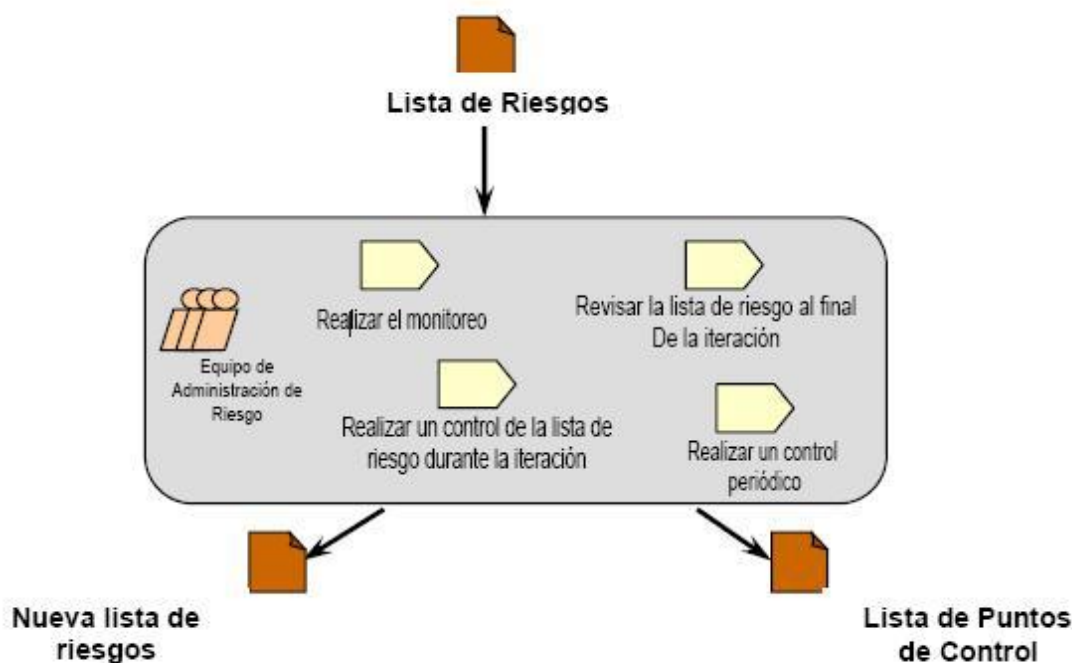


Figura 7 Proceso de monitoreo de riesgos

➤ Realizar un Monitoreo

En esta actividad se realiza el monitoreo de:

1. Los riesgos.
2. La efectividad del plan.
3. Las estrategias.

➤ **Realizar un control de la lista de riesgo durante la iteración**

Este paso comprende, un seguimiento de la lista de riesgos durante toda la iteración que se esté desarrollando en ese momento.

➤ **Revisar la lista de riesgo al final de la iteración**

Se actualiza la lista de riesgo, pero al final de la iteración.

➤ **Realizar un control periódico**

El monitoreo de los riesgos debe desarrollarse periódicamente, para detectar si se han introducido nuevos riesgos. En caso de que la detección sea efectiva, se debe pasar automáticamente al flujo de trabajo de Identificación (4).

Existen algunas técnicas que se pueden aplicar durante todo el proceso como son:

1. Revisión e inspección al proyecto.
2. Reuniones de avance de proyecto.
3. Auditorias.

**Artefacto de Entrada**

1. lista de riesgos

**Artefacto de Salida**

1. Nueva lista de riesgo.
2. Listas de punto de control.

## 1.4 Metodología de desarrollo

Todo desarrollo de software es riesgoso y difícil de controlar por lo que se hace necesario la utilización de una metodología apropiada para el caso en cuestión. Cuando el proyecto que se va a desarrollar es de mayor envergadura, se hace necesaria la utilización de una metodología de desarrollo. Lo cierto es que muchas veces no se encuentra la más adecuada y se opta por hacer o diseñar una

metodología propia, algo que por supuesto no está mal, siempre y cuando cumpla con el objetivo.

## ➤ **Programación Extrema (XP)**

La Programación Extrema es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. Las cuatro variables de esta metodología son:

1. Coste: Máquinas, especialistas y oficinas.
2. Tiempo: Total y de Entregas.
3. Calidad: Externa e Interna.
4. Alcance: Intervención del cliente.

XP surgió como respuesta y posible solución a los problemas derivados del cambio en los requerimientos. Se plantea que es una metodología a emplear en proyectos de riesgo. Con esta metodología se aumenta la productividad.

## ➤ **OpenUp**

OpenUp es un proceso unificado, abierto, mínimo y suficiente lo que brinda la posibilidad de solo incluir el contenido fundamental y necesario. Conserva las características principales de la metodología de desarrollo RUP por lo que se aplica de forma iterativa e incremental, provee los componentes básicos que pueden servir de base a procesos específicos a pesar de no tener lineamientos para todos los elementos que se manejan en un proyecto. Es una forma de desarrollo ágil y ligero donde la mayoría de sus elementos fomentan el intercambio entre los equipos de desarrollo, sincronizando intereses y compartiendo conocimientos, principio fundamental que promueve las buenas prácticas para propiciar un ambiente saludable y de colaboración. Maximiza los beneficios al equilibrar las prioridades, permitiendo a los desarrolladores implementar una solución que cumpla con los requerimientos del proyecto, minimiza el riesgo al centrarse en la arquitectura y proporciona la retroalimentación y mejoramiento continuo al realizar prácticas que permitan incrementos progresivos a las funcionalidades.

## ➤ **RUP**

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

El Proceso Unificado Racional (Rational Unified Process en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

El RUP está basado en 5 principios clave que son:

Adaptar el proceso.

Balancear prioridades.

Demostrar valor iterativamente.

Elevar el nivel de abstracción.

Enfocarse en la calidad.

El ciclo de vida RUP

El RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al culminar cada una de ellos, estos a la vez se dividen en fases que finalizan con un hito donde se debe tomar una decisión importante:

- 1 Inicio: El objetivo en esta etapa es determinar la visión del proyecto.
- 2 Elaboración: En esta etapa el objetivo es determinar la arquitectura óptima.
- 3 Construcción: En esta etapa el objetivo es obtener la capacidad operacional inicial.
- 4 Transición: El objetivo es llegar a obtener el release del proyecto.

Los elementos de RUP son:

1. Actividades: Procesos que se determinan en cada iteración.
2. Trabajadores: Personas o entes involucradas en cada proceso.
3. Artefactos: Puede ser un documento, un modelo, o un elemento de modelo.

## ➤ **Metodología seleccionada (RUP)**

La metodología utilizada para el desarrollo de esta aplicación fue RUP (Rational Unified Process), pues la misma es una metodología escalable al estar formada por fases y estas a su vez por flujos de trabajo, lo cual hace que el proceso de desarrollo del software sea iterativo e incremental, y junto con UML (Lenguaje Unificado de Modelado) constituye la metodología estándar más utilizada en el mundo para el análisis, implementación y documentación de sistemas orientados a objetos. RUP

describe como utilizar de forma efectiva reglas de negocio y procedimientos probados en el desarrollo de software para equipos de desarrollo de software, de forma disciplinada y asignando tareas y responsabilidades con la finalidad de tener una interacción transparente con los clientes para lograr su satisfacción. Aunque el sistema a desarrollar no es de gran envergadura, se decidió además utilizar RUP, porque esta metodología confiere un gran peso a la documentación y esta será de vital importancia para futuros desarrolladores de sistemas semejantes o para el mantenimiento o mejoramiento de esta solución propuesta.

## 1.5 Roles y técnicas de los Roles

En un proceso de desarrollo existen flujos de trabajo, en los cuales participan trabajadores que generan artefactos ocupando un rol. En el Proceso Unificado de Desarrollo (RUP) se definen varias responsabilidades para estos roles y pueden estar representados por una o varias personas. Los roles que se van a asignar para la solución del problema científico son: Analista y Desarrollador.

### 1.5.1 Responsabilidades que desempeñan y artefactos que generan

El analista en sentido general agrupa los roles que están involucrados en los flujos de trabajo de modelación del negocio y levantamiento de requisitos; que pueden estar representados por una o varias personas entre los que se encuentran: Analista de procesos del negocio, diseñador del negocio, analista del sistema y especificador de requerimientos.

- Analista de procesos del negocio.

Es el responsable de definir la arquitectura del negocio; los casos de uso del negocio y actores del negocio, así como sus relaciones. Los artefactos que realiza son: modelo de casos de uso del negocio, modelo de análisis del negocio, reglas del negocio, glosario del negocio, documento de arquitectura del negocio, visión del negocio, metas del negocio, especificación suplementaria del negocio y objetivos de la organización.

- Diseñador del negocio.

Es el encargado de detallar la especificación de la organización y especificar el flujo de trabajo de los casos de usos del negocio en términos de trabajadores del negocio y entidades del negocio. Los artefactos que realiza son: actor del negocio, caso de uso

del negocio, realización de caso de uso del negocio, trabajador del negocio, entidad del negocio, sistema del negocio y eventos del negocio.

➤ Analista del sistema.

Es el responsable de dirigir y coordinar el proceso de captura de requisitos y desarrollo del modelo de casos de uso, definiendo las funcionalidades y límites del sistema. Los artefactos que realiza son: plan de gestión de requerimientos, documento visión, modelo de casos de uso, glosario, solicitudes de los stakeholder, storyboard, especificación suplementaria y atributos de requerimientos.

➤ Especificador de requerimientos.

Es el encargado de especificar los detalles de una o varias partes de la funcionalidad del sistema, describiendo uno o varios aspectos de los requisitos, además de agrupar los casos de usos en paquetes. Los artefactos que realiza son: casos de uso, paquete de casos de uso, requerimientos de software y especificación de requerimientos del software.

El desarrollador en sentido general agrupa los roles que están involucrados fundamentalmente en los flujos de trabajo de análisis y diseño e implementación del sistema, que pueden estar representados por una o varias personas entre los que se encuentran: Arquitecto de software, diseñador, diseñador de interfaz de usuario, diseñador de cápsula, diseñador de base de datos, implementador e integrador.

➤ Arquitecto de software

Es el responsable de la arquitectura del software, que incluye las principales decisiones técnicas que limitan el diseño global y la ejecución del proyecto.

➤ Diseñador

Es el responsable de diseñar una parte del sistema, dentro de las limitaciones de los requerimientos, la arquitectura y los procesos de desarrollo del proyecto. Identifica y define las responsabilidades, operaciones y relaciones entre los elementos del diseño. Además asegura que el diseño es consistente con la arquitectura del software y es detallado a un punto donde la implementación puede continuar. Los artefactos que genera son: realización de los casos de uso, clases del análisis, subsistemas del diseño, paquetes del diseño, clases del diseño y clases de prueba.

## ➤ Diseñador de interfaz de usuario

Es el que coordina y elabora el diseño de la interfaz del usuario abarcando en la misma los requerimientos de forma tal que sean cumplidos en su totalidad. Vale aclarar que no es responsable de implementar dicha interfaz de usuario, solo de diseñar la forma visual de la misma. Los artefactos que genera son: prototipo de interfaz de usuario y mapa de navegación.

## ➤ Implementador

Este rol desarrolla los componentes de software y efectúa las pruebas de desarrollador para la integración en subsistemas más grandes, de acuerdo con los estándares adoptados de proyecto.

## ➤ Integrador

Este rol dirige la planificación y la ejecución de la integración del elemento de implementación para producir compilaciones.

### **1.5.2 Roles desempeñados y artefactos que generan por el equipo de trabajo**

Para el desarrollo del trabajo solo se han tenido en cuenta algunos roles y artefactos que agrupan los roles genéricos de analista y desarrollador; a partir de las características propias del proyecto como la limitación del tiempo de desarrollo y la poca claridad de los procesos del negocio. Estos son:

- Analista de procesos del negocio: Reglas del negocio.
- Diseñador del negocio: Modelo de dominio.
- Especificador de requerimientos: Especificación de requerimientos; Realización de casos de uso del sistema.
- Analista del sistema: Actores del sistema; Casos de uso del sistema: Diagrama de los casos de uso del sistema.
- Diseñador de interfaz de usuario: Prototipos de interfaz no funcionales.
- Diseñador: Clases del diseño; Realización de los casos de uso del diseño.
- Arquitecto de software: Diagrama de despliegue; Diagrama de componente.

## 1.6 Tecnologías y herramientas en las que se apoya para la solución

### 1.6.1 Lenguaje de modelado

#### UML

El Proceso Unificado RUP (Rational Unified Process) utiliza el Lenguaje de Modelado UML (Unified Modeling Language) para preparar todos los esquemas de un sistema software. UML es una parte esencial del Proceso Unificado, está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Mediante UML es posible establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código. UML recomienda 9 diagramas que son modelados durante las diferentes fases de desarrollo del software, es decir, representan las siguientes vistas del sistema:

- Diagrama de Casos de Uso: modela la funcionalidad del sistema agrupándola en descripciones de acciones ejecutadas por un sistema para obtener un resultado.
- Diagrama de Clases: muestra las clases que componen el sistema y cómo se relacionan entre sí.
- Diagrama de Objetos: muestra una serie de objetos y sus relaciones.
- Diagrama de Secuencia: enfatiza la interacción entre los objetos y los mensajes que intercambian entre sí junto con el orden temporal de los mismos.
- Diagrama de Colaboración: igualmente, muestra la interacción entre los objetos resaltando la organización estructural de los objetos en lugar del orden de los mensajes intercambiados.
- Diagrama de Estados: modela el comportamiento de acuerdo con eventos.
- Diagrama de Actividades: simplifica el Diagrama de Estados modelando el comportamiento mediante flujos de actividades.
- Diagrama de Componentes: muestra la organización y las dependencias entre un conjunto de componentes.
- Diagrama de Despliegue: muestra los dispositivos que se encuentran en un sistema y su distribución dentro del mismo.

### 1.6.2 Herramientas de modelado



Con el fin de automatizar los aspectos clave de todo el proceso de desarrollo de un sistema se hace necesaria la utilización de las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador).

## ➤ **Rational Rose**

La herramienta líder en el mundo de modelación visual para el proceso de modelación del negocio, análisis de requerimientos y diseño de arquitectura de componentes. Rational Rose es una herramienta de diseño de software orientado a objeto mediante UML para el modelado visual y la construcción de componentes para las aplicaciones de software. Características del Rational Rose:

- Chequeo de la sintaxis UML.
- Disponibilidad en múltiples plataformas.
- Generación de código a partir de los modelos
- Generación de documentación automáticamente.
- Mantiene la consistencia de los modelos del sistema software.
- Capacidades de ingeniería inversa (crear modelo a partir de código).
- Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.

Rational Rose es la herramienta CASE que comercializan los desarrolladores de UML y que soporta de forma completa la especificación del UML. Propone la utilización de cuatro tipos de vistas para realizar un diseño del sistema: vista de Caso de Uso, vista Lógica, vista de Componentes y la vista de Despliegue; permitiendo crear y refinar de esta forma un modelo completo que represente el dominio del problema y el sistema de software.

## ➤ **Visual Paradigm-UML**

Visual Paradigm para UML es una de las herramientas CASE del mercado, considerada como muy completa y fácil de usar, es multiplataforma y proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. Fue creada para el ciclo vital completo del desarrollo del software que lo automatiza y acelera, permitiendo la captura de requisitos, análisis, diseño e implementación.

Visual Paradigm-UML también proporciona características tales como generación del código, ingeniería reversa y generación de informes. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de clases. Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al instante, creando de manera simple toda la documentación. Apoya los estándares más recientes de las notaciones de Java y de UML. Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros.

Esta herramienta ofrece un entorno de creación de diagramas para UML; presenta un diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad; posee un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación; capacidades de ingeniería directa (versión profesional) e inversa; modelo y código que permanece sincronizado en todo el ciclo de desarrollo; disponibilidad de múltiples versiones para cada necesidad; disponibilidad de integrarse en los principales IDEs como Eclipse, JBuilder, NetBeans IDE, Oracle JDeveloper, BEA Weblogic, Borland®, para soportar las fases de implementación en el desarrollo de un software; disponibilidad en múltiples plataformas, y muy útil para la generación de código fuente en PHP, también con el Paradigm se generan script de las tablas de salidas para las clases persistentes.

Visual Paradigm, es una herramienta CASE para realizar el Análisis y Diseño, utiliza el Lenguaje Unificado de Modelado (UML). Esta herramienta forma parte de los requerimientos de software definidos en este trabajo.

## ➤ **Herramienta seleccionada (Visual Paradigm)**

Por todo lo antes planteado se decide utilizar como herramienta de modelado para el desarrollo de la aplicación el Visual Paradigm. Éste presenta una interfaz de usuario de fácil uso y muy amigable que permite realizar los diagramas y artefactos que se generan durante el desarrollo del software.

### **1.6.3 Lenguajes de programación**

## ➤ **PERL**

Estructuralmente, Perl está basado en un estilo de bloques como los del C o AWK, y fue ampliamente adoptado por su destreza en el procesador de texto y no tener ninguna de las limitaciones de los otros lenguajes de script.

Perl es un lenguaje de propósito general, originalmente desarrollado para la manipulación de texto y que ahora es utilizado para un amplio rango de tareas incluyendo administración de sistemas, desarrollo Web, programación en red, desarrollo de GUI y más. Se previó que fuera práctico (facilidad de uso, eficiente, completo) en lugar de hermoso (pequeño, elegante, mínimo) [otra] (manual Unix).

Sus principales características son:

- La estructura completa de Perl deriva ampliamente del lenguaje C.
- Es fácil de usar.
- Soporta tanto la programación estructurada como la programación orientada a objetos y la programación funcional.
- Tiene incorporado un poderoso sistema de procesamiento de texto y una enorme colección de módulos disponibles.
- Perl es un lenguaje imperativo, con variables, expresiones, asignaciones, bloques de código delimitados por llaves, estructuras de control y subrutinas.
- En Perl 5, se añadieron características para soportar estructuras de datos complejas, funciones de primer orden (p. e. clausuras como valores) y un modelo de programación orientada a objetos. Éstos incluyen referencias, paquetes y una ejecución de métodos basada en clases y la introducción de variables de ámbito léxico, que hizo más fácil escribir código robusto (junto con el programa strict).
- Una característica principal introducida en Perl 5 fue la habilidad de empaquetar código reutilizable como módulos.
- Larry Wall indicó más adelante que "la intención del sistema de módulos de Perl 5 era apoyar el crecimiento de la cultura Perl en vez del núcleo de Perl".
- Todas las versiones de Perl hacen el tipificado automático de datos y la gestión de memoria.
- El intérprete conoce el tipo y requerimientos de almacenamiento de cada objeto en el programa; reserva y libera espacio para ellos según sea necesario.

## ➤ **Python**

Python es un lenguaje de programación creado en el año 1990 por Guido van Rossum, es el sucesor del lenguaje de programación ABC. Los usuarios lo consideran

como un lenguaje más limpio para programar. Permite la creación de todo tipo de programas incluyendo los sitios web.

Su código no necesita ser compilado, por lo que se dice que el código es interpretado.

Es un lenguaje de programación multiparadigma, lo cual fuerza a que los programadores adopten por un estilo de programación particular:

- Programación orientada a objetos.
- Programación estructurada.
- Programación funcional.
- Programación orientada a aspectos.

Este lenguaje es libre y de fuente abierta, de propósito general, cuenta con gran cantidad de funciones y librerías, es sencillo y rápido de programar, es multiplataforma, su licencia es de código abierto (Opensource), además es portable y orientado a objetos. Sin embargo por ser un lenguaje interpretando es lento. Python es comparado habitualmente con Perl.

### ➤ **JAVA**

El lenguaje para la programación en Java, es un lenguaje orientado a objeto, de una plataforma independiente. Esta programación tiene muchas similitudes con el lenguaje C y C++.

El código generado por el compilador Java es independiente de la arquitectura: podría ejecutarse en un entorno UNIX, Mac o Windows. Esto es posible ya que el código generado por el compilador se ejecuta mediante una máquina virtual y por el procesador del ordenador directamente. Esto permite que los Applets puedan ejecutarse en cualquier máquina que se conecte a ella independientemente de qué sistema operativo emplee, siempre y cuando el ordenador en cuestión tenga instalada una máquina virtual de Java.

### ➤ **Lenguaje seleccionado (Java)**

Luego del análisis de estos lenguajes se decidió Java para desarrollar la herramienta por ser un lenguaje cuya portabilidad está verdaderamente probada y no requiere largos períodos de tiempo para el desarrollo de las aplicaciones. Otras características de dicho lenguaje son las siguientes (5):

#### **Orientado a objetos**

Java fue diseñado como un lenguaje orientado a objetos desde el principio. Los objetos agrupan en estructuras encapsuladas tanto sus datos como los métodos (o

funciones) que manipulan esos datos. La tendencia del futuro, a la que Java se suma, apunta hacia la programación orientada a objetos, especialmente en entornos cada vez más complejos y basados en red.

## **Distribuido**

Java proporciona una colección de clases para su uso en aplicaciones de red, que permiten abrir sockets y establecer y aceptar conexiones con servidores o clientes remotos, facilitando así la creación de aplicaciones distribuidas.

## **Robusto**

Java fue diseñado para crear software altamente fiable. Para ello proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. Sus características de memoria liberan a los programadores de una familia entera de errores (la aritmética de punteros), ya que se ha prescindido por completo de los punteros, y la recolección de basura elimina la necesidad de liberación explícita de memoria.

## **Indiferente a la arquitectura**

Java está diseñado para soportar aplicaciones que serán ejecutadas en los más variados entornos de red, desde Unix a Windows Nt, pasando por Mac y estaciones de trabajo, sobre arquitecturas distintas y con sistemas operativos diversos. Para acomodar requisitos de ejecución, el compilador de Java genera bytecodes: un formato intermedio indiferente a la arquitectura diseñada para transportar el código eficientemente a múltiples plataformas hardware y software. El resto de los problemas los soluciona el intérprete de Java.

### **1.6.4 Entorno de desarrollo**

#### ➤ **Eclipse**

Eclipse es una plataforma de desarrollo extensible, basada en Java y de tipo open-source (CPL). Originalmente fue desarrollada por Alphaworks, laboratorio de desarrollo de IBM, pero actualmente está manejado por un consorcio de varias empresas, (Borland, Hitachi, etc.). Lo interesante de Eclipse, es que conforma una suerte de estándar o framework. Muchos vendedores de software basan sus ofertas de IDE en eclipse, al que suman conjuntos de plug-ins.

Este entorno de desarrollo integrado ofrece, el control del editor de código, del compilador y del depurador desde una única interfaz de usuario. Su misión consiste en evitar tareas repetitivas, facilitar la escritura de código correcto, disminuir el tiempo de

depuración e incrementar la productividad del desarrollador. Estas tareas pueden realizarse de muchas maneras distintas: mediante la inclusión de asistentes para las tareas más habituales y mecánicas, de editores que completen automáticamente el código y señalen los errores sintácticos, de gestores de archivos fuente, etc. Eclipse no es un IDE más a añadir a la lista, el objetivo de IBM ha sido crear una plataforma de desarrollo modular que cualquier herramienta de desarrollo pueda usar con cualquier lenguaje de programación.

### ➤ **JBuilder**

JBuilder es un entorno de desarrollo integrado para el lenguaje de programación java desarrollado por Borland. Posee varias ediciones, la Enterprise, para aplicaciones J2EE, Web Services y struts. La Developer, para el desarrollo completo de aplicaciones Java, y la Foundation, con capacidades básicas para iniciarse en Java. No fue seleccionado para el desarrollo de la herramienta JBuilder como IDE de desarrollo por no ser multiplataforma, solo puede ser ejecutado sobre el sistema operativo Windows.

### ➤ **NetBeans**

NetBeans es un entorno de desarrollo integrado OpenSource y de distribución gratuita que proporciona herramientas muy cómodas y de fácil uso para el desarrollo de aplicaciones sobre la plataforma JAVA. Entonces se pueden desarrollar bajo esta IDE soluciones J2SE, J2ME y J2EE. Entre las características que proporciona tenemos (6):

1. Desarrollo de aplicaciones multiplataforma sobre: MacOS, Windows, Linux.
2. Add-ons para desarrollo Móvil, integración con SOA, optimización de aplicaciones y desarrollo con C y C++.
3. Cliente CVS integrado
4. Crecimiento de plataforma por medio de plugins. Entre los plugins que existen se tienen los siguientes:
  - Herramientas java que sirven para la mejora de desarrollo de aplicaciones
  - Herramientas de modelado UML
  - Herramientas XML, etc.
5. Struts, JSF, EJB, WebServices, etc.

### ➤ **Entorno de desarrollo seleccionado (NetBeans)**

Para la solución de este sistema se seleccionó el NetBeans, pues posee numerosas características que hacen que el IDE sea atractivo para cualquier desarrollador, incluyendo la amplia integración de las características específicas de la tecnología Java que no se encuentran disponibles en otros conjuntos de herramientas de aplicaciones multiplataforma.

NetBeans elimina la necesidad de los equipos de desarrollo que tienen que invertir demasiado tiempo manteniendo los modelos actualizados con revisiones exhaustivas que garanticen la actualización de indicadores y códigos de anotación. Juntas, estas características pueden ahorrar al desarrollador muchas horas de trabajo y acelerar la disponibilidad de un nuevo programa. Entre las características que lo evalúan está su facilidad de uso, su cumplimiento de regulaciones, sus perfiles de rendimiento, además de su flexibilidad entre plataformas.

## 1.6.5 Framework

### ➤ Hibernate

Hibernate es considerada como una herramienta para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones. Hibernate es una herramienta ORM (Object Relational Mapping) completa que ha conseguido en un tiempo record una excelente reputación en la comunidad de desarrollo posicionándose claramente como el producto OpenSource líder en este campo gracias a sus prestaciones, buena documentación y estabilidad. Es un motor de persistencia, que es el componente software encargado de traducir entre objetos y registros. Un motor de persistencia de código abierto es Hibernate, que nos permitirá hacer cosas como poder guardar un objeto en la base de datos simplemente con o borrarlo. Hibernate es un entorno de trabajo que tienes como objetivo facilitar la persistencia de objetos Java en bases de datos relacionales y al mismo tiempo la consultas de estas bases de datos para obtener objetos.

Se selecciono este framework debido a que puede configurarse y ejecutarse en la mayoría de aplicaciones java y entornos de desarrollo. Generalmente, Hibernate se utiliza en aplicaciones cliente/servidor de dos y tres capas, desplegándose Hibernate únicamente en el servidor. Hibernate soporta la mayoría de los sistemas de bases de

datos SQL. Hibernate ofrece facilidades para recuperación y actualización de datos, control de transacciones, repositorios de conexiones a bases de datos, consultas programáticas y declarativas, y un control de relaciones de entidades declarativas. Hibernate es una muy buena herramienta en lo que se refiere a mapeo clases en una base de datos relacional (7).

Entre las características se destacan:

- No intrusivo (estilo POJO, es decir, mapear objetos java)
- Muy buena documentación (forums para ayuda, libro)
- Comunidad activa con muchos usuarios
- Transacciones, caché, asociaciones, polimorfismo, herencia, persistencia transitiva.
- Potente lenguaje de consulta (HQL): subqueries, outer, joins, ordering, proyeccion (report query), paginacion.
- Facil testeo.
- No es estandard.

## 1.6.6 Gestores de base de datos

### ➤ MySQL

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario. Fue creado por la empresa de software libre MySQL AB y desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009— lo desarrolla como software libre en un esquema de licenciamiento dual. Por un lado se ofrece bajo la GNU GPL<sup>1</sup>() para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y el copyright del código está en poder del autor individual, MySQL es propietario y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

---

<sup>1</sup> GNU GPL (GNU General Public License), es una licencia creada por la Free Software Foundation a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software.



MySQL está escrito en C y C++ y probado con multitud de compiladores y dispone de APIs para muchas plataformas diferentes. Permite manejar multitud de tipos para columnas y registros de longitud fija o variable así como permite conexiones entre diferentes máquinas con distintos sistemas operativos.

## ➤ **PostgreSQL**

PostgreSQL es un gestor de base de datos de código abierto<sup>2</sup> que posee una gran escalabilidad, es capaz de ajustarse al número de procesadores y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta.

### Principales características:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc.).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.
- Tiene licencia de tipo BSD

## ➤ **Gestor de Base de datos seleccionado (PostgreSQL)**

Se selecciono trabajar con el PostgreSQL por las características que posee, dándole al usuario una facilidad de uso, así como almacenar gran cantidad de datos y distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos.

Teniendo en cuenta las características analizadas de los gestores de base de datos expuestos, se opta por usar PostgreSQL en su versión 8.2, la cual incluye mejoras de rendimiento, características de SQL 2003, incluyendo funciones de agregación

---

<sup>2</sup> Código abierto, es el término con el que se conoce al software distribuido y desarrollado libremente.

estadística, sentencias VALUE con múltiples registros, UPDATE RETURNING y funciones de agregación de múltiples columnas, así como índices invertidos generalizados, lo cual constituye una forma más escalable y programable de indexar datos semi-estructurados y texto.

### Conclusiones

La gestión de riesgos es de vital importancia desde la concepción y planificación de un proyecto informático y hasta su cierre, pues de no tenerse en cuenta podría ser una de las causas potenciales de su fracaso. Esta actividad debe implementarse fundamentalmente de forma proactiva. En este capítulo se propone asumir la adaptación del modelo definido por RUP para la UCI, teniendo en cuenta un grupo de pasos comunes en modelos de gestión de riesgos ya existentes. Dada la necesidad de contribuir a la gestión de riesgos en los proyectos productivos se decide implementar un grupo de actividades presentes en la identificación, análisis, planificación resolución y/o mitigación. Se decidió además, utilizar bajo la metodología de desarrollo RUP, el Visual Paradigm como herramienta CASE, haciendo uso del lenguaje de modelado, UML. Para la implementación se propone utilizar como lenguaje de programación Java, como entorno de desarrollo el NetBeans utilizando el framework hibernate y como gestor de base de datos PostgreSQL.

## CAPITULO 2: CARACTERÍSTICAS DEL SISTEMA

### Introducción

En el presente capítulo, se identificarán el o los proceso de negocio relacionados con la gestión de riegos, que una vez descritos permiten identificar las posibles áreas a automatizar. Para la modelación del sistema propuesto se identificarán los requisitos no funcionales y los requisitos funcionales asociados a estas posibles áreas, agrupándolos por casos de uso que más tarde serán priorizados y descritos para obtener la documentación necesaria para realizar el diseño del sistema.

### 2.1 Modelado del Negocio

Para este trabajo se decidió realizar modelo de negocio y no modelo de dominio, puesto que se ve que el proceso de modelamiento permite obtener una visión de la organización que permita definir los procesos, roles y responsabilidades de la organización en los modelos de casos de uso del negocio y de objetos.

#### 2.1.1 Actores de Negocio

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externo; con los que le negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de los resultados (8).

Actores	Descripción
Directivo de producción	Persona encargada de realizar la gestión de riesgos, que puede ser el líder del proyecto, vicedecano de producción o algún estudiante del proyecto asignado para realizar la gestión.

Tabla 9 Relación de actores del negocio.

#### 2.1.2 Trabajador de Negocio

Un trabajador del negocio representa una abstracción de un ser humano con ciertas capacidades que se requieren en un acaso de uso del negocio. Un trabajador representa el conocimiento y las habilidades que alguien necesita para hacerse cargo del trabajo como trabajador del proyecto (9).

Trabajadores	Descripción
Administrador de riesgos	Persona encargada de registrar los riesgos, además de analizarlos, priorizarlos y planificarlos.

Tabla 10 Relación de trabajadores del negocio.

### 2.1.3 Diagrama Modelo del Negocio

Un diagrama de casos de uso del negocio representa gráficamente a los procesos del negocio y su interacción con los actores del negocio (8).

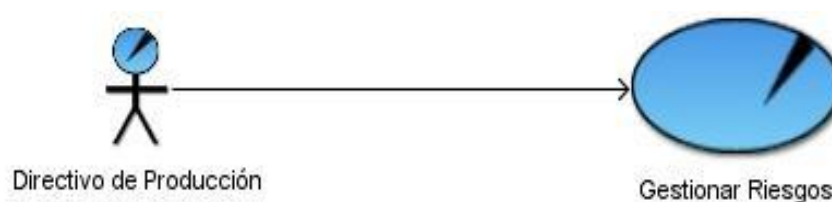


Figura 8 Diagrama de Modelo de Negocio

### 2.1.4 Diagrama de actividades

El diagrama de actividad es un grafo (grafo de actividades) que contiene estados en que puede hallarse una actividad, puede contener bifurcaciones, así como divisiones de control en hilos concurrentes. Describe un proceso que explora el orden de las tareas o actividades que logran los objetivos del negocio. Es similar a un diagrama de estados en el cual todos o la mayoría de los estados son estados de actividad y en la cual todas o la mayoría de las transiciones se disparan al completarse las acciones en los estados fuentes precedentes (8).

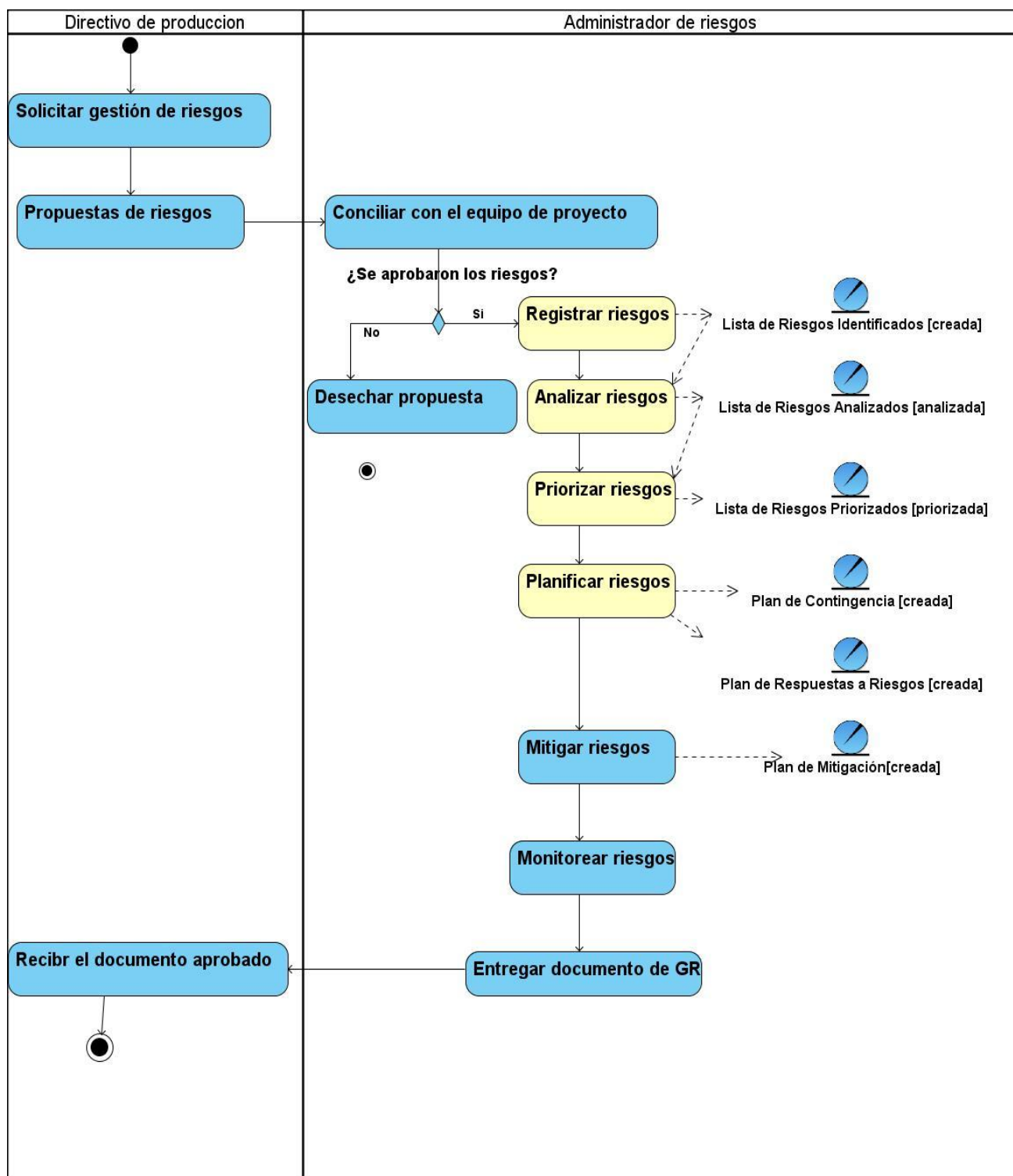


Figura 9 Diagrama de actividades

### 2.1.5 Modelo de Objeto del Negocio

Un modelo de objeto del negocio constituye un modelo interno a un negocio que describe como cada caso de uso del negocio es llevado a cabo por parte de un conjunto de trabajadores que utilizan un conjunto de entidades del negocio.

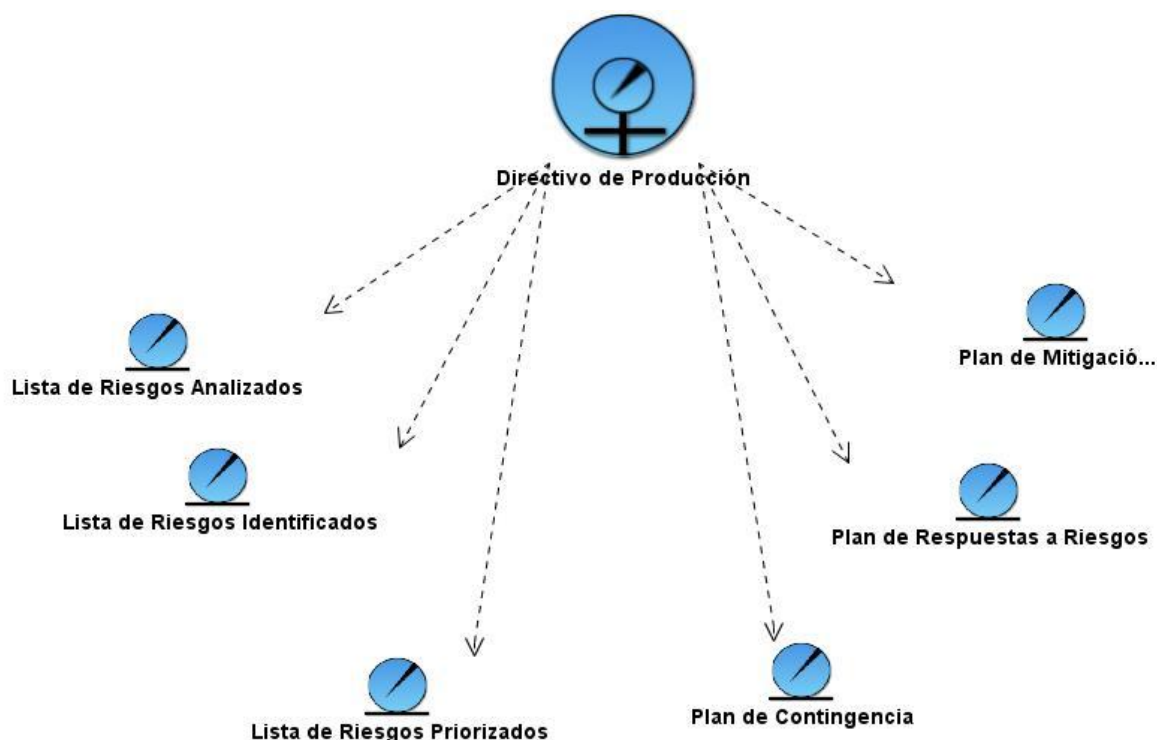


Figura 10 Diagrama del Modelo de Objeto del Negocio

## 2.2 Modelado del Sistema

### 2.2.1 Solución informática propuesta

En el presente trabajo se hace referencia a la tesis: Propuesta para la Gestión de Riesgo en los proyectos productivos de la UCI; donde se hace un estudio en el cual Rational Unified Process (RUP) realiza una propuesta de series de pasos para la gestión de riesgos en los proyectos productivos de software en la UCI. Para cumplir con los objetivos trazados se propone una aplicación que automatice los procesos de identificación, análisis, priorización y planeación que propone RUP, dejando los restantes procesos a posteriores trabajos de diplomas. La solución informática permitirá además generar unas series de reportes que permitirá que los datos digitales sean cargados de un tipo archivo en formato digital a través de una aplicación de escritorio que le permitirá al usuario conocer los resultados.

### 2.2.2 Definición de los Requerimientos Funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir (10). De acuerdo con los objetivos planteados el sistema debe ser capaz de:

R1: Conectar base de datos

R2: Autenticar usuario

R3: Cambiar contraseña

R4: Gestionar usuario

R5: Identificar proyecto

R6: Valorar riesgos propuestos

### **Paquete: Gestionar Riesgos**

R7: Identificar riesgos

R8: Analizar riesgos

R9: Priorizar riesgos relevantes

R10: Proponer riesgos

### **Paquete: Reportes**

R11: Mostrar documento de plan de contingencia

R12: Mostrar documento de gestión de riesgos

### **Paquete: Nomencladores**

R13: Gestionar polo productivo

R13.1 Adicionar polo

R13.2 Modificar polo

R13.3 Eliminar polo

R14: Gestionar proyecto por polo

R14.1 Adicionar proyecto

R14.2 Modificar proyecto

R14.3 Eliminar proyecto

R15: Gestionar clasificación de riesgos

R15.1 Adicionar clasificación

R15.2 Modificar clasificación

R15.3 Eliminar clasificación

R16: Gestionar riesgos

R16.1 Adicionar riesgos

R16.2 Modificar riesgos

R16.3 Eliminar riesgos

R17: Gestionar plan de contingencia

R17.1 Adicionar plan de contingencia

R17.2 Modificar plan de contingencia

R17.3 Eliminar plan de contingencia

R18: Gestionar rol

R18.1 Adicionar rol

R18.2 Modificar rol

R18.3 Eliminar rol

### 2.2.3 Definición de los Requerimientos No Funcionales

Los requerimientos no funcionales especifican propiedades del sistema como restricciones de ambiente y desarrollo, performance, dependencias de plataformas, mantenimiento y confiabilidad. Constituyen otros requisitos que no forman parte de la funcionalidad principal de la aplicación. Son propiedades o cualidades que el producto debe tener. Representan las características del producto.

Dentro de los requerimientos no funcionales para el desarrollo de la herramienta se encuentran:

- **Requisitos de funcionalidad**

El sistema debe someterse a una etapa de adiestramiento en la que los usuarios se familiaricen con la aplicación y sean detectados los posibles errores, o puedan surgir posibles cambios en la interfaz para que los usuarios queden totalmente complacidos.

- **Apariencia o interfaz externa**

El sistema debe contar con una interfaz amigable, fácil de comprender y usar, donde el usuario pueda orientarse fácilmente.

- **Usabilidad**

El sistema le ofrecerá al usuario la posibilidad de realizar reportes para tener conocimiento de la información y analizar los resultados de las mismas. En este sentido se centra el diseño de la aplicación y en específico de las



## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

---

interfaces que harán posible el intercambio de datos de manera que le resulte al usuario de fácil entendimiento.

- **Seguridad**

La aplicación tiene un sistema de seguridad, en la cual solo pueden acceder los usuarios con permisos autorizados.

- **Soporte**

Terminada la aplicación será sometida a pruebas con el objetivo de comprobar su funcionalidad.

- **Requerimientos de Portabilidad**

La herramienta propuesta podrá ser usada bajo cualquier sistema operativo, para su implementación se usaron Herramientas de Programación y Gestión de Bases de Datos que son multiplataforma.

- **Software**

Para PC de Aplicación:

Se debe disponer de sistemas operativos Linux, Windows 2000 o superior para la instalación de la aplicación. Debe tenerse instalado el Java Runtime Environment (JRE) versión 6.0 o superior.

Para PC servidora:

Se debe disponer de una PC con sistema operativo Linux ó Windows 2000 para soportar la base de datos de la aplicación y debe tener instalado el pgAdmin III y "SQL Manager for postgresSQL".

- **Hardware**

Para el desarrollo y puesta en práctica del proyecto se requieren máquinas con los siguientes requisitos.

Para PC Cliente:

- 512 MB de RAM
- Máquina Virtual Java 6.0

Para PC Servidora:

- 1 GB de RAM
- 5 GB de capacidad del disco duro.

### 2.2.4 Actores del Sistema a Automatizar

Un actor es como una entidad externa del sistema que de alguna manera participa en la historia del caso de uso. Por lo general estimula el sistema con eventos de entradas o recibe algo de él. O sea, es un rol de un usuario, que puede intercambiar información o puede ser un recipiente pasivo de información y representa a un ser humano, a un software o a una máquina que interactúa con el sistema. Acorde a la situación en la que se desarrolla la herramienta se encontraron los siguientes actores:

Actores	Descripción
Usuario	Es el encargado de administrar los riesgos de su proyecto: identificarlos, analizarlos, priorizarlos y planificar la contingencia.
Administrador del Sistema	Es el encargado de insertar usuarios, nomencladores, los riesgos y sus atributos para que el usuario pueda interactuar con ellos.

Tabla 11 Actores del Sistema

### 2.2.5 Diagrama de casos de uso del sistema a automatizar

Un diagrama de casos de uso del sistema contiene actores, casos de uso del sistema y las relaciones entres ellos.

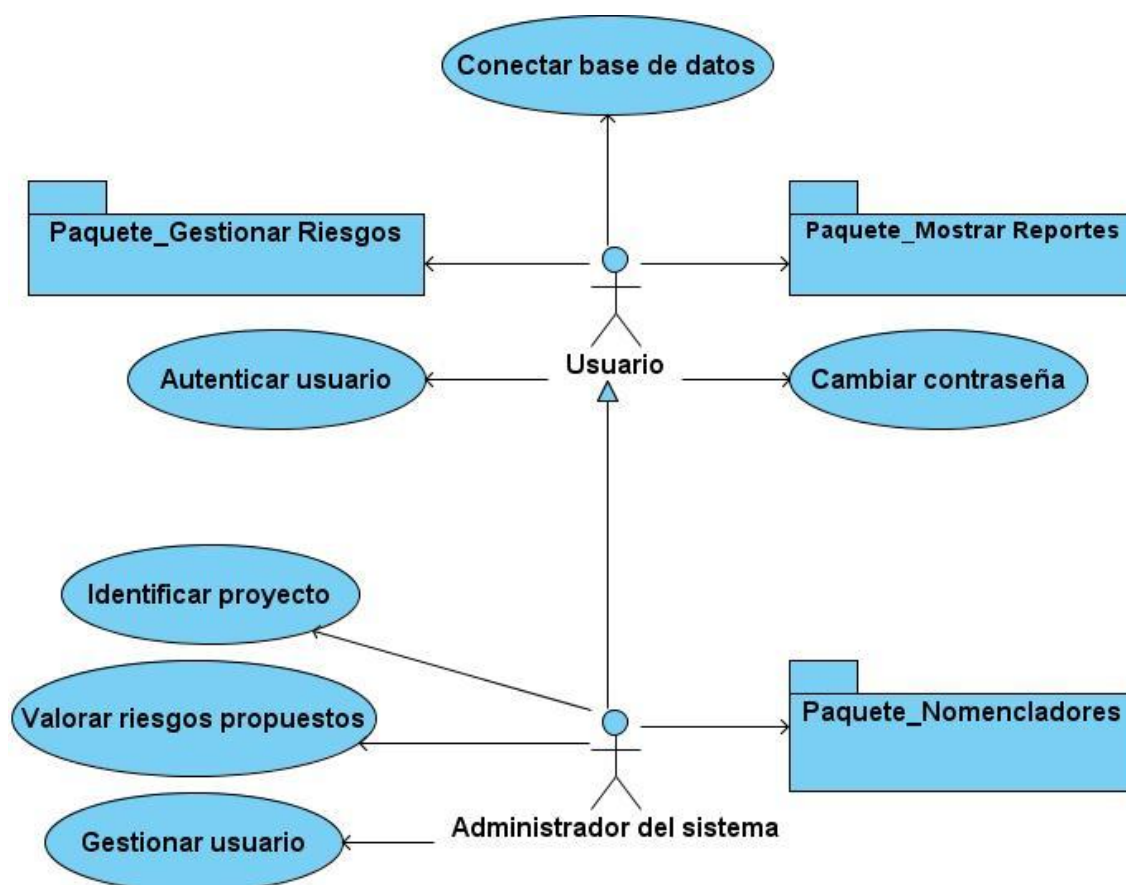


Figura 11 Diagrama de Casos de Uso del Sistema

### 2.2.6 Descripción de los Casos de Uso del Sistema. [Ver Anexo 1](#)

## Conclusiones

Durante la modelación del negocio se identificó dentro del proceso Gestionar Riesgos, cuatro actividades posibles a automatizar: identificar, analizar, priorizar y planificar riesgos. Dentro de estas actividades se identificaron 30 requisitos funcionales que fueron agrupados en 18 casos de uso. Para el mejor entendimiento del diagrama de casos de uso del sistema se utilizó como criterio de empaquetamiento la agrupación de los casos de uso que apoyan una misma funcionalidad dentro del sistema propuesto y se utilizaron entre los patrones de casos de uso: CRUD y múltiples actores - rol común .

## CAPÍTULO 3: DISEÑO DEL SISTEMA

### Introducción

En el este capítulo se presentan los artefactos correspondientes al flujo de Trabajo de Análisis y Diseño según la metodología RUP, teniendo en cuenta el lenguaje de programación, el framework de desarrollo y los patrones de arquitectura y diseño empleados. Mediante los diagramas de clases se muestra una vista estática del sistema, dejando la representación de la interacción entre las clases, para los diagramas de secuencia. Se obtiene además una vista de despliegue, mostrando la distribución física en nodos dispositivos o de procesamiento del sistema diseñado.

### 2.2 Análisis y Diseño

El objetivo de este flujo de trabajo es traducir los requisitos a una especificación que describe cómo implementar el sistema.

Los objetivos del análisis y diseño son:

- Transformar los requisitos al diseño del futuro sistema.
- Desarrollar una arquitectura para el sistema.
- Adaptar el diseño para que sea consistente con el entorno de implementación, diseñando para el rendimiento.

El análisis consiste en obtener una visión del sistema, de modo que sólo se interesa por los requisitos funcionales. Especifica el comportamiento funcional del sistema, independientemente de los aspectos relativos al ambiente en el que va a ser finalmente implementado. El modelo de análisis captura completamente y con exactitud los requerimientos del sistema.

Por otro lado el diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, definiendo cómo el modelo de análisis orientado a la aplicación va a ser realizado en el ambiente de implementación, en definitiva cómo cumple el sistema sus objetivos.

El resultado final más importante de este flujo de trabajo será el modelo de diseño. Consiste en colaboraciones de clases, que pueden ser agregadas en paquetes y subsistemas.

### 2.3 Patrón arquitectónico. Modelo-Vista-Controlador

Un patrón arquitectónico define a una familia de sistemas informáticos en términos de su organización estructural, describe componentes y las relaciones entre ellos con las restricciones de la aplicación, la composición asociada y el diseño para su construcción. Uno de los patrones más usados para desarrollar aplicaciones es el Modelo-Vista-Controlador (MVC), el cual es una aproximación al software que separa la lógica de la aplicación y de la presentación.

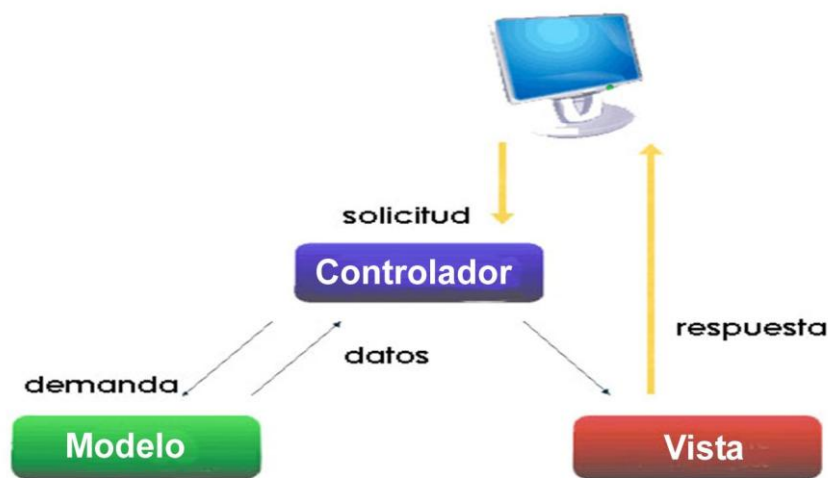


Figura 12 Interacción entre los componentes del patrón MVC

Este patrón de arquitectura de software separa los datos de una aplicación, de la interfaz del usuario y de la lógica de control de la aplicación en tres componentes distintos:

- **Modelo:** Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos. Es el responsable de acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento. Lleva un registro de las vistas y controladores del sistema.
- **Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario. Las vistas son responsables de recibir datos del modelo y las muestras al usuario.
- **Controlador:** Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. Es responsable de

recibir los eventos de entrada. El Controlador sirve como un intermediario entre el Modelo, la Vista y cualquier otro recurso necesario.

### 2.3.1 Ventajas del MVC

- Soporte para múltiples vistas: ya que la vista se separa del modelo y no hay ninguna dependencia directa entre vista y modelo, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos al mismo tiempo.
- Proporciona un mecanismo de configuración a componentes complejos muchos más tratable que el puramente basado en eventos (el modelo puede verse como una representación estructurada del estado de la interacción).
- Mayor soporte a los cambios: los requisitos de interfaz tienden a cambiar más rápidamente que las reglas de negocio. Puesto que el modelo no depende de las vistas, la adición de nuevos tipos de vista al sistema generalmente no afectan al modelo. Por tanto, el ámbito del cambio se limita a la vista.
- Además de tener en cuenta sus ventajas, utilizamos el patrón de arquitectura Modelo Vista Controlador siguiendo el estándar utilizado para el desarrollo del proyecto.

### 2.4 Diagrama de Clases de Diseño

El diagrama de diseño describe la realización física de los casos de uso, centrándose en como los requisitos funcionales y no funcionales junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema. Se representa de una forma sencilla como se lleva a cabo la colaboración y las responsabilidades de las distintas clases que forman el sistema. A continuación se presentan los Diagramas de Clases del Diseño (DCD) de los casos de usos del sistema (CUS) considerados críticos, los restantes se encuentran en los anexos. [Ver Anexo 2.](#)

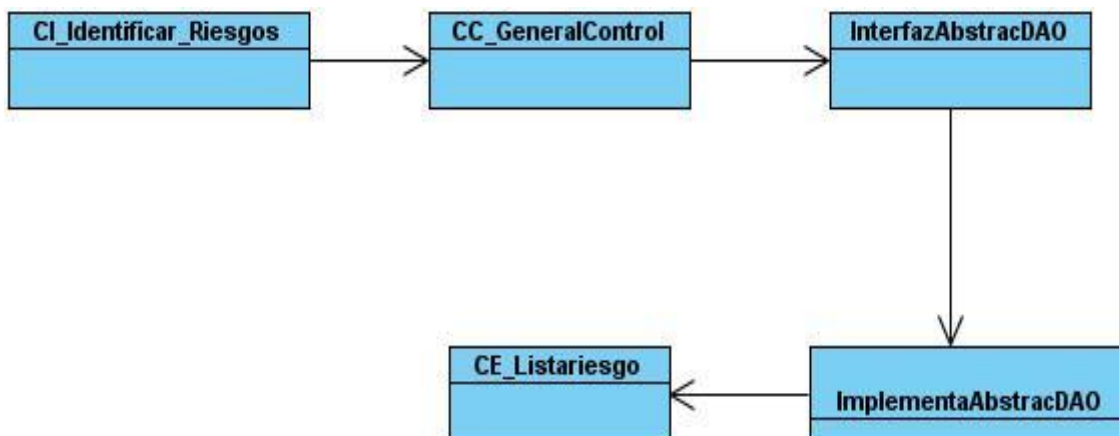


Figura 13 Diagrama de Clases del Diseño CU Identificar riesgos

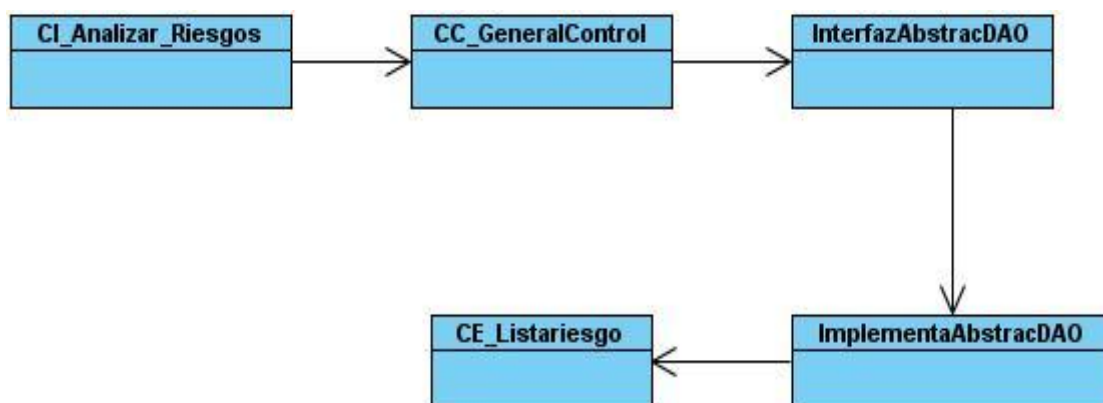


Figura 14 Diagrama de Clases del Diseño CU Analizar riesgos

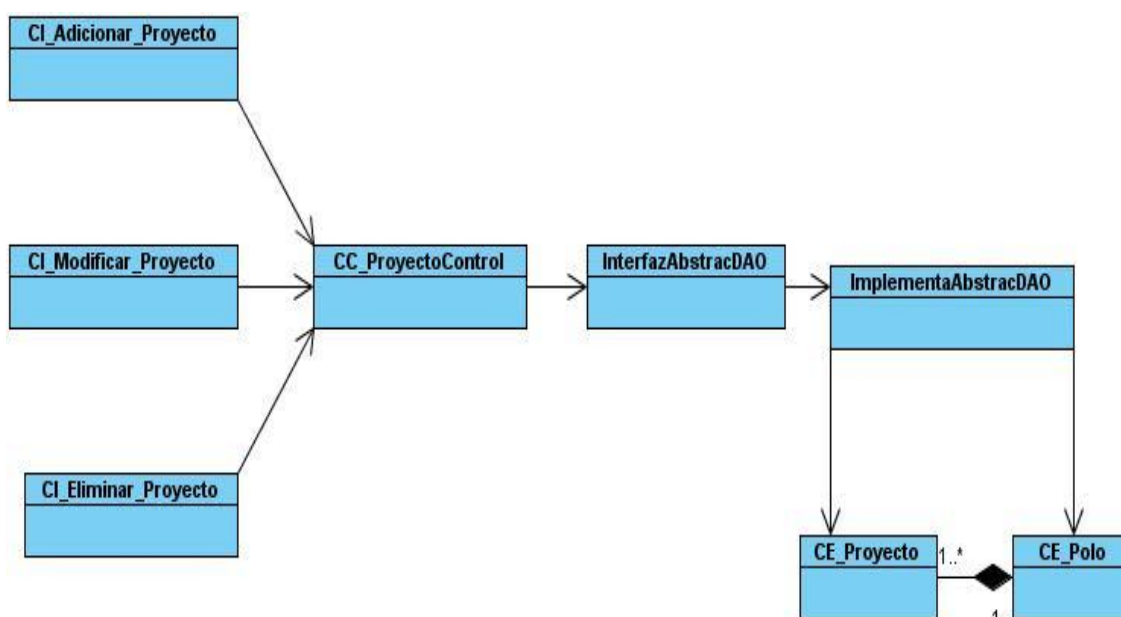


Figura 15 Diagrama de Clases del Diseño CU Gestionar proyectos

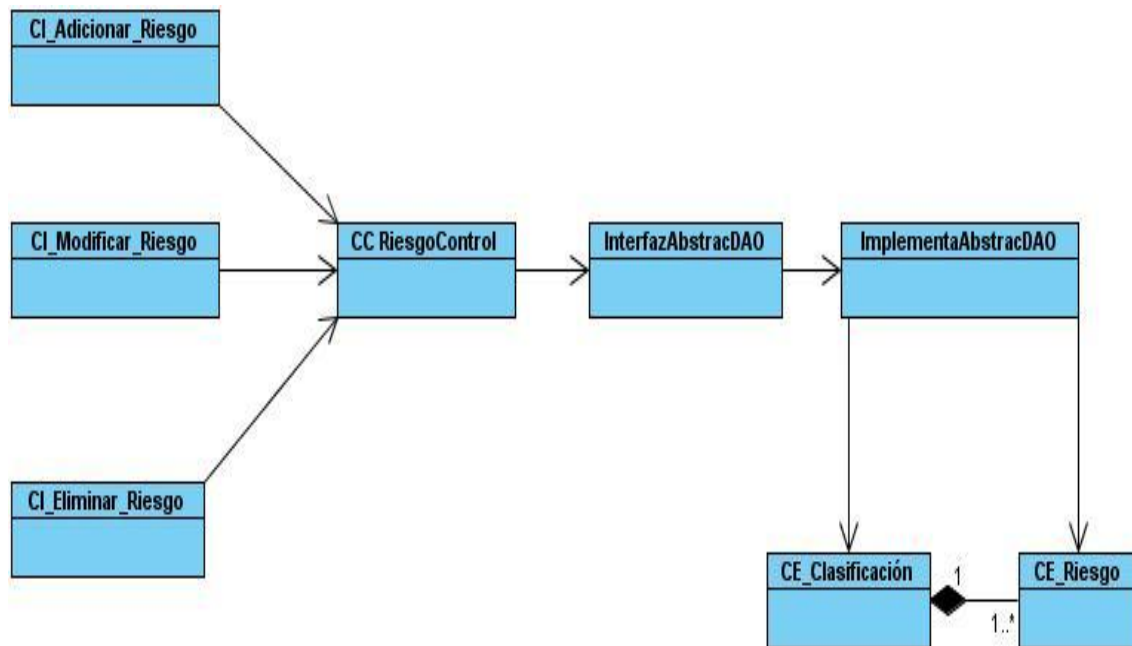


Figura 16 Diagrama de Clases del Diseño CU Gestionar riesgos

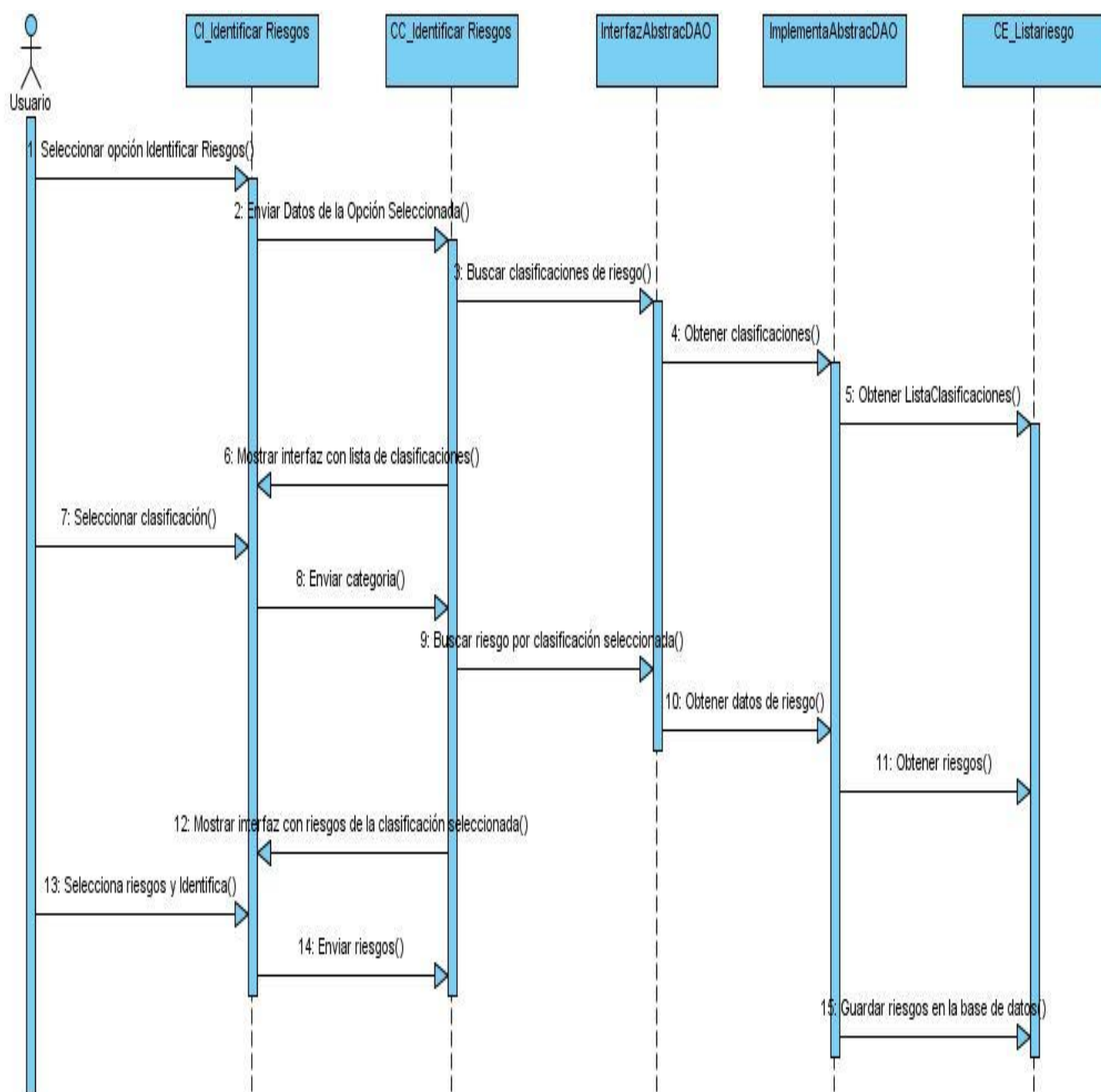
### 2.4.1 Diagrama de Interacción

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema, lo que conlleva modelar instancias concretas o prototípicas de clases interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento. Estos diagramas se expresan de dos formas, una en diagramas de colaboración, que son los que muestran las relaciones entre los objetos y los mensajes que intercambian y, otra en diagramas de secuencia, que muestran las interacciones expresadas en función de secuencias temporales y fue a su vez el escogido para describir cada escenario de los casos de uso de la presente aplicación.

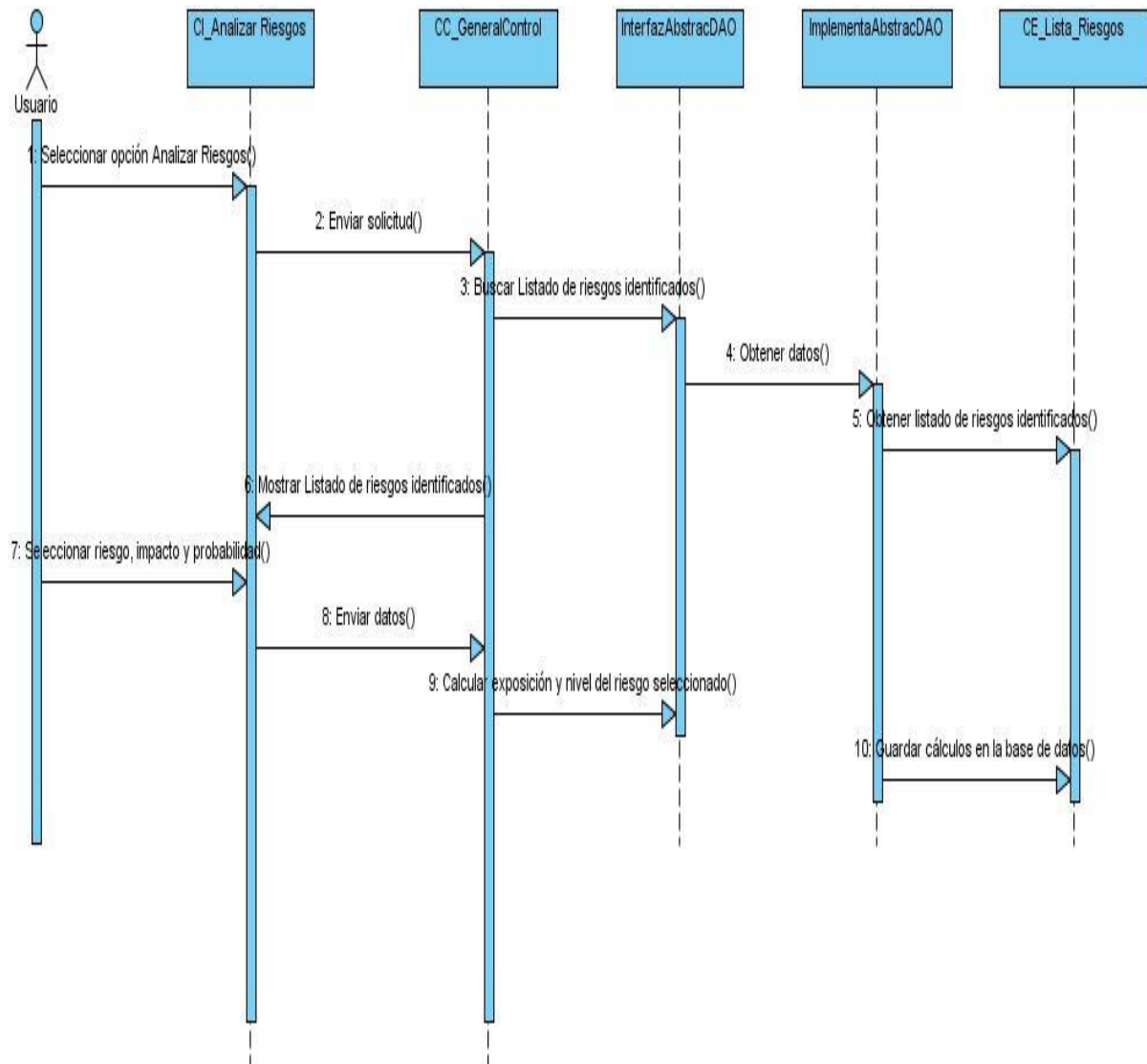
### 2.4.2 Diagrama de Secuencia

Un diagrama de secuencia muestra las interacciones entre objetos ordenadas en secuencia temporal. Muestra los objetos que se encuentran en el escenario y la secuencia de mensajes intercambiados entre los objetos para llevar a cabo la funcionalidad descrita por el escenario. Documentan el diseño desde el punto de vista de los casos de uso, observando qué mensajes se envían a los objetos, componentes o casos de uso y dan una aproximación del tiempo que consume el método invocado. Los diagramas de secuencia de los flujos principales de los CU más significativos del sistema se muestran a continuación. [Ver Anexo 3.](#)





**Figura 17 Diagrama de Secuencia CU Identificar riesgos**



**Figura 18 Diagrama de Secuencia CU Analizar riesgos**

## 2.5 Diseño de la Base de Datos

Las bases de datos necesitan de una definición de su estructura que le permitan almacenar datos, reconocer el contenido, y recuperar la información. Es por ello que uno de los pasos cruciales en la construcción de una aplicación que maneje una base de datos, es sin duda, el diseño de la misma. Esto permite que las tablas sean definidas apropiadamente y se garantice que tenga eficiencia. Al diseñar una base de datos, se refleja la estructura del problema en el mundo real, evita el almacenamiento de información redundante y permite además representar todos los datos esperados, incluso con el paso del tiempo.

Para lograr diseñar la Base de Datos se partió de la definición de las clases persistentes. La persistencia es la capacidad de un objeto de mantener su valor en el

espacio y en el tiempo. Luego del refinamiento y clasificación de estas clases y sus atributos se realizó la construcción de un diagrama de clases persistentes que se muestra a continuación.

### 2.5.1 Modelo lógico de datos

El modelo lógico de datos representa la información que maneja el sistema, siendo una fuente de información para el modelo físico. A continuación se muestra el Modelo lógico de datos diseñado para la base de datos de la presente aplicación, el cual no es más que el diagrama de clases persistentes. El diagrama de clases persistentes es un diagrama de clases en el que, como bien indica su nombre, solo aparecen las clases persistentes, de las que hay que expandir detalles estructurales. Estas clases persistentes tuvieron como origen las clases clasificadas como entidad, ya que estas modelan la información y el comportamiento asociado al fenómeno en cuestión.

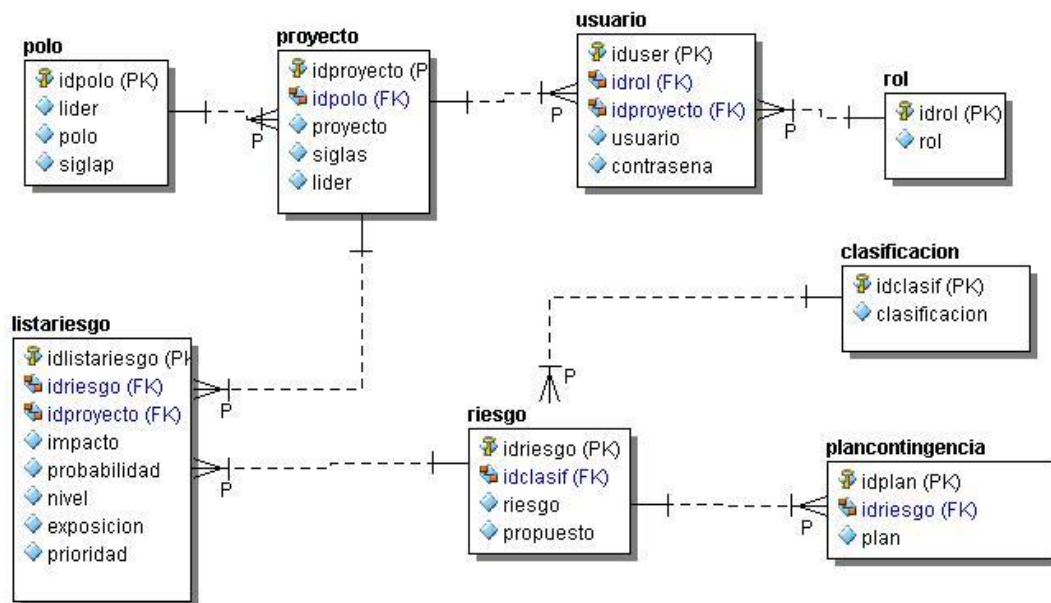


Figura 19 Modelo lógico de la base de datos

### 2.6 Modelo de despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño.

Elementos:

- Procesadores: Nodos que tienen capacidad de procesamiento, computadoras por lo general.
- Dispositivos: Nodos que no tienen capacidad de procesamiento.
- Protocolos: Estándares que deben existir implementados en la red entre máquinas, para efectuar cierta comunicación.

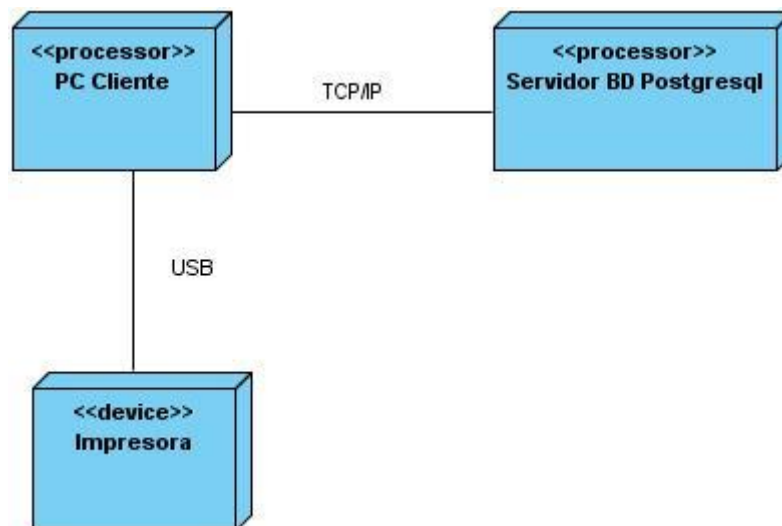


Figura 20 Diagrama de Despliegue

## Conclusiones

En este capítulo se obvió la realización del “Análisis”, teniendo en cuenta que es una aproximación al diseño, no obligatoria para la realización de este y que las secuencias de acciones a modelar son muy sencillas, aconsejándose la realización directamente del diseño, la cual permitió obtener una visión avanzada del sistema basada en la comprensión de los requisitos funcionales y no funcionales. Se modelaron además los diagramas de interacción correspondientes a los escenarios más significativos, de modo que quedó descrito más claramente el flujo de los mensajes entre los objetos del sistema y la dinámica de la aplicación en el diagrama de secuencia. Se tuvo en cuenta que la implementación se hará utilizando Java como lenguaje de programación,

Hibernate como framework de desarrollo, bajo el patrón de arquitectura Modelo Vista Controlador. Finalmente se realizó el diseño de la base de datos, basando su desarrollo en la construcción de un modelo de datos, que tuvo su origen en el diagrama de clases persistentes.

## CAPÍTULO 4: IMPLEMENTACIÓN DEL SISTEMA

### Introducción

En el presente capítulo se pasa a la implementación del software en el lenguaje de programación escogido anteriormente, en la cual se describen cómo los elementos del modelo de diseño se implementan en términos de componentes, los cuales se mostraran a través de los diagramas de componentes.

### 4.1 Implementación

En este flujo de trabajo se comienza con el resultado del diseño en el cual se propone crear un plano del modelo de implementación. El flujo de trabajo de implementación describe cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue (11).

### 4.2 Diagrama de Componente

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean estos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. A continuación se muestran los diagramas de componentes.

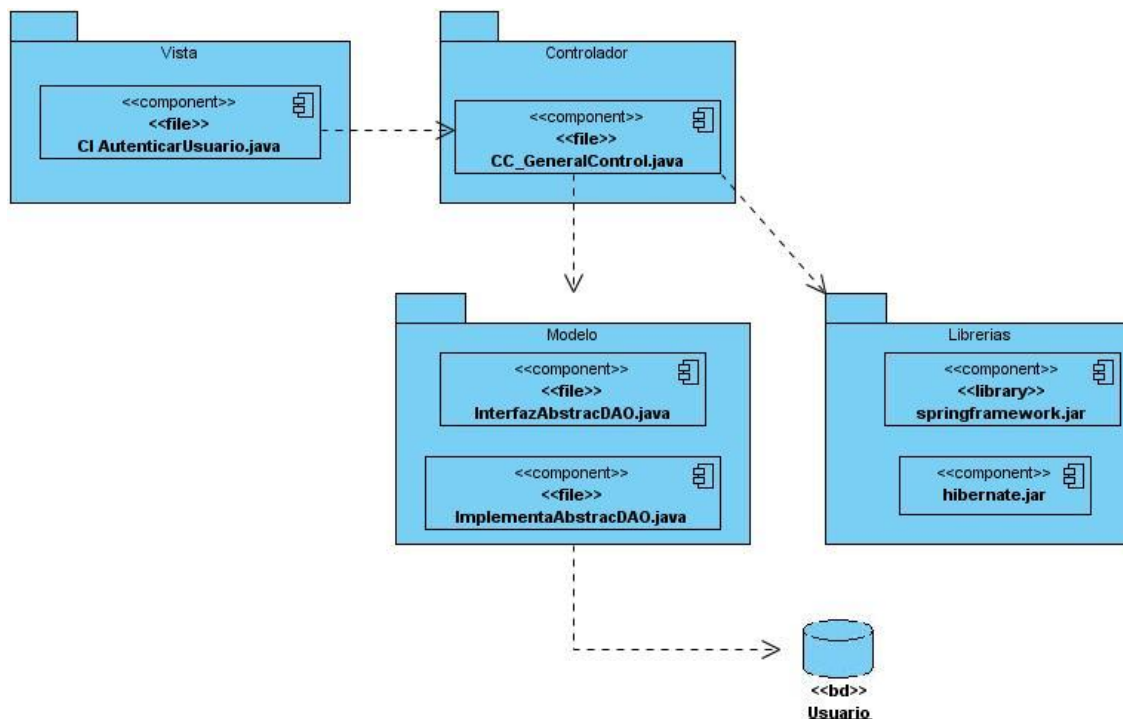


Figura 21 Diagrama de componente CU Autenticar Usuario

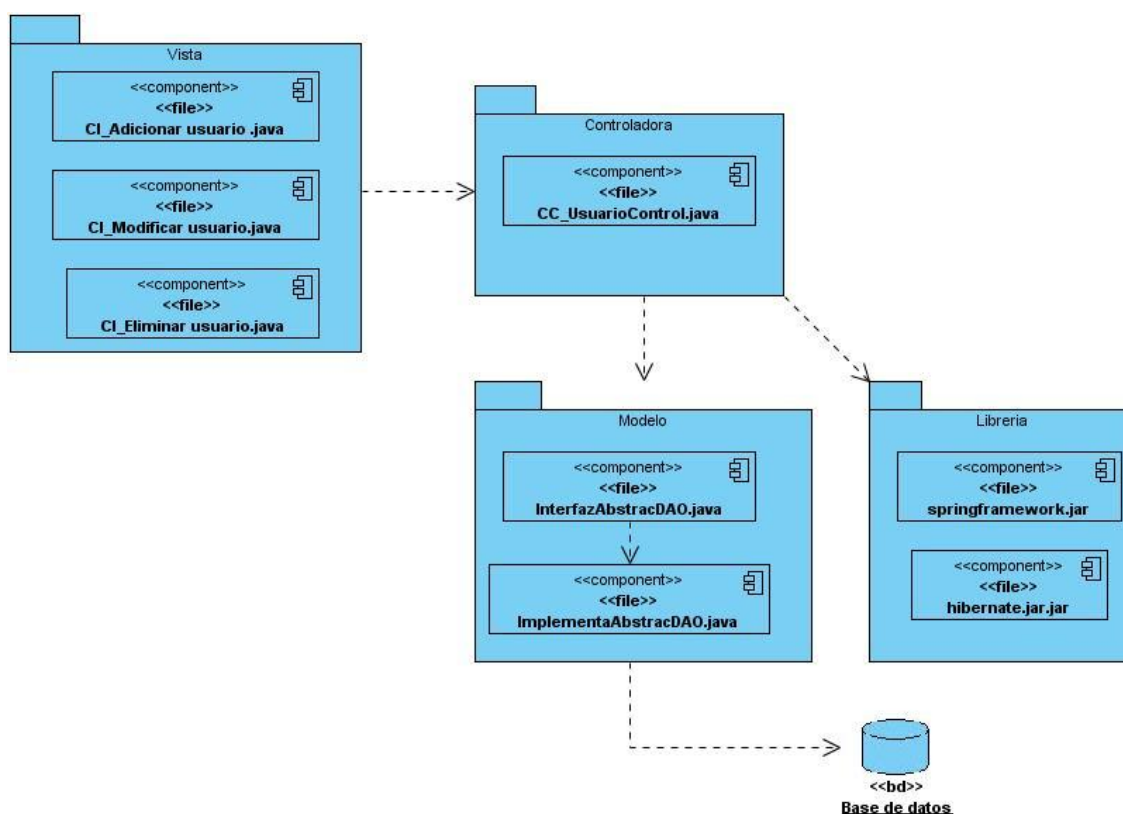


Figura 22 Disgrama de componente CU Gestionar Usuario

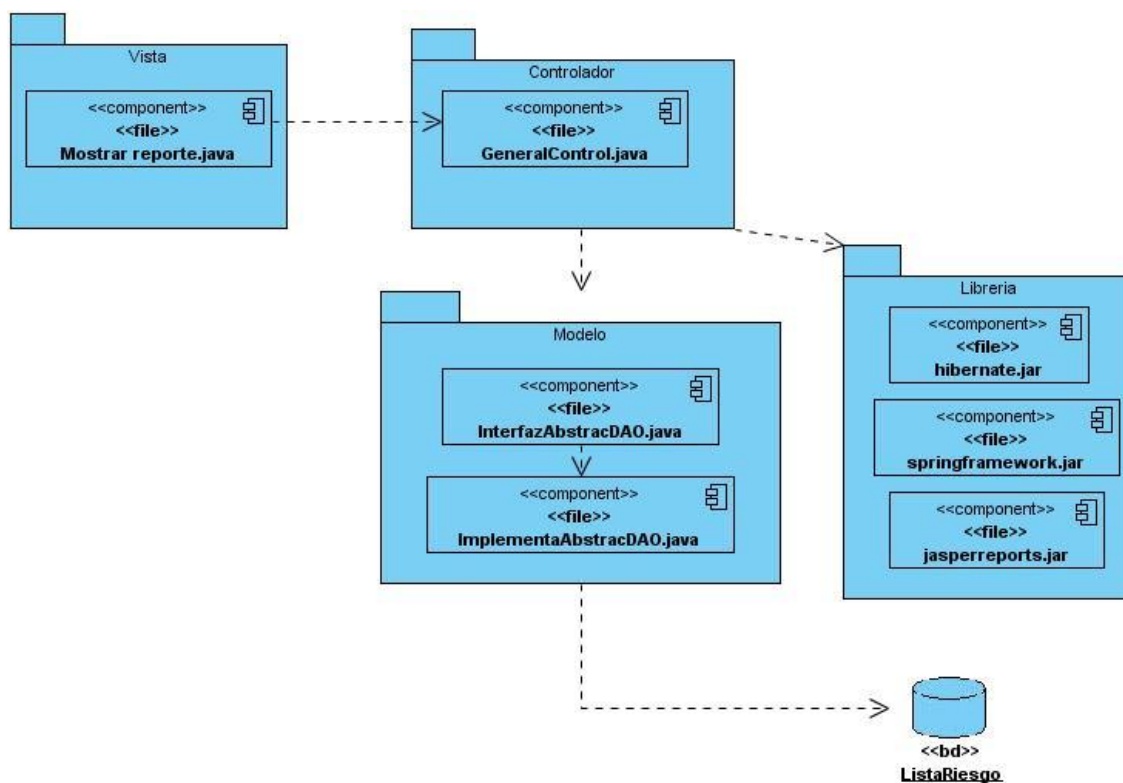


Figura 23 Diagrama de componente Paquete: Reporte

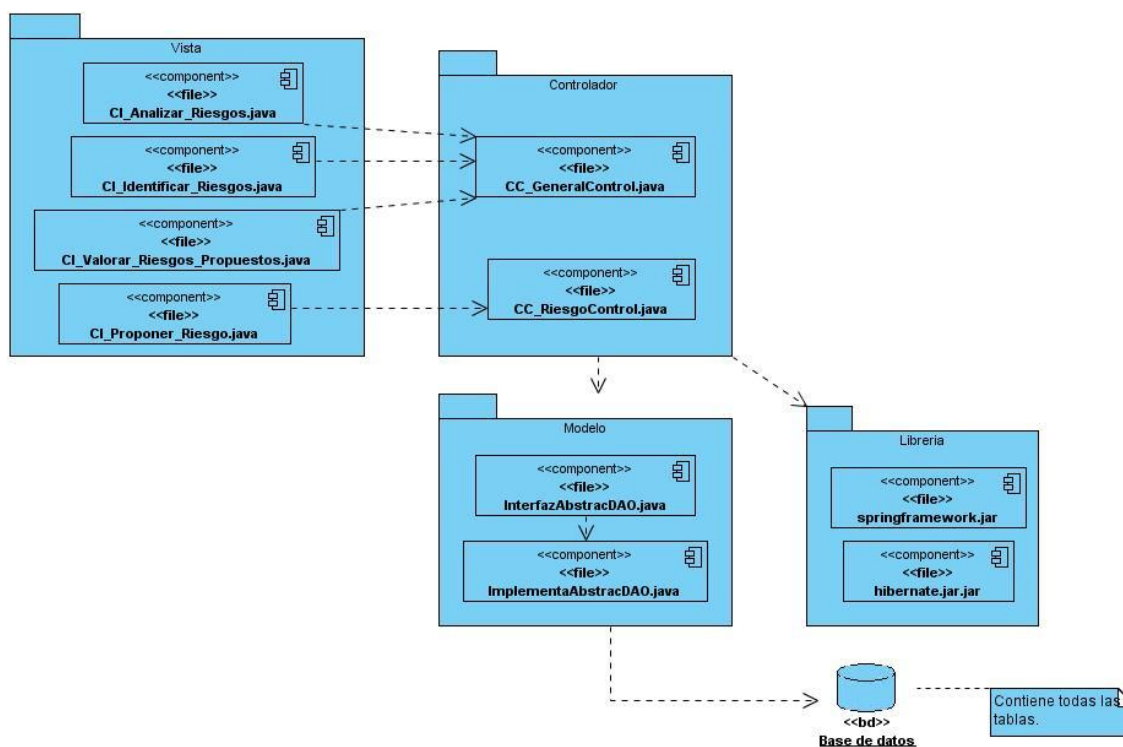


Figura 24 Diagrama de componente CU Gestión riesgos



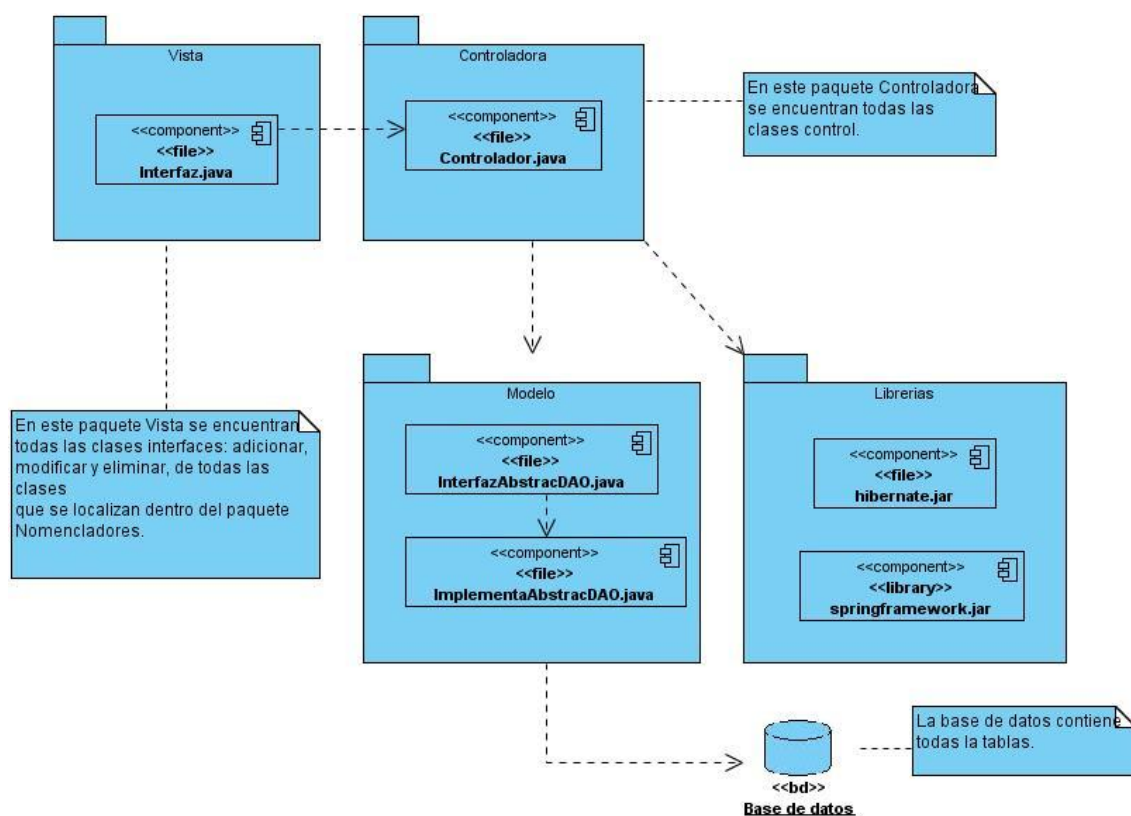


Figura 25 Diagrama de componente Paquete: Nomencladores

### Conclusiones

Durante el desarrollo de este capítulo se utilizó el diagrama de componentes para estructurar el modelo de implementación en términos de subsistemas de implementación, así como mostrar las relaciones entre sus elementos.

### CONCLUSIONES

- Se modeló el negocio y el sistema propuesto, generándose los artefactos necesarios para diseñar la solución propuesta.
- Se diseñó la aplicación a implementar teniendo en cuenta lenguaje y framework a utilizar durante la implementación.
- Se obtuvo una aplicación informática que contribuirá a la gestión de los riesgos en los proyectos productivos.
- Con la aplicación se contará con una base de datos de riesgos, lo que constituye el punto de partida para realizar futuros análisis estadísticos de la incidencia de los mismos.

### RECOMENDACIONES

- Automatizar los procesos de resolución y/o mitigación y monitoreo.
- Modificar la base de datos para que constituya un archivo histórico del comportamiento de los riesgos en los proyectos para futuros estudios.
- Aplicar técnicas de inteligencia artificial y minería de datos para realizar estudios de tendencia y reportes estadísticos del comportamiento de los riesgos en los proyectos productivos.

## REFERENCIAS BIBLIOGRÁFICAS

### Trabajos citados

1. 1 07. Gestion de Riesgos. [En línea] [Citado el: 23 de marzo de 2009.] [www.di.uniovi.es/~aquilino/Asignaturas/ProyectosInformatica/Documentos/07-GestionRiesgos.pdf](http://www.di.uniovi.es/~aquilino/Asignaturas/ProyectosInformatica/Documentos/07-GestionRiesgos.pdf).
2. RIESGOS DEL SOFTWARE. [En línea] [Citado el: 23 de marzo de 2009.] [www.sitios.uach.cl/caminosfor/CristianSalazar/SIE/RS.html](http://www.sitios.uach.cl/caminosfor/CristianSalazar/SIE/RS.html) - .
3. **Cabrera, Armando, Castillo, Luis Enriquez y Cuenca, Luis Eduardo.** Gestión del Riesgo. [En línea] mayo de 2008. [Citado el: 24 de marzo de 2009.] <http://www.slideshare.net/lecastillox/gestion-del-riesgo>.
4. **López Cabrera, Yanisleidy; Alvarez Lamas, Tailys.** Propuesta para la Gestión de Riesgo en los proyectos productivos de la UCI. [En línea] 2007. [Citado el: 10 de marzo de 2009.] [http://bibliodoc.uci.cu/TD/TD\\_0466\\_07.pdf](http://bibliodoc.uci.cu/TD/TD_0466_07.pdf).
5. Características del lenguaje Java. [En línea] [Citado el: 25 de marzo de 2009.] [www.iec.csic.es/CRIPTONOMICON/java/quesjava.html](http://www.iec.csic.es/CRIPTONOMICON/java/quesjava.html).
6. **Valdez Altamirano, Alfonso.** Comparativo de Entornos de Desarrollo Integrados. [En línea] [Citado el: 25 de marzo de 2009.] <http://bicuos.com/wp-content/uploads/2009/05/comparativoides.pdf> - .
7. 1.0 DEFINICION DE HIBERNATE. [En línea] [Citado el: 3 de abril de 2009.] [www.unife.edu.pe/ing/desarrollo.doc](http://www.unife.edu.pe/ing/desarrollo.doc).
8. Fase de Inicio. Modelo del Negocio. [En línea] 2007-2008. [Citado el: 5 de abril de 2009.] <http://eva.uci.cu/mod/resource/view.php?id=11553>.
9. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El proceso unificado de desarrollo de software*. la Habana : Félix Varela, 2004. Volumen I.
10. Fase de Inicio. Flujo de trabajo de requerimientos. . [En línea] 2008-2009. [Citado el: 9 de junio de 2009.] <http://eva.uci.cu/mod/resource/view.php?id=12103>.
11. Flujo de Implementación. [En línea] [Citado el: 20 de marzo de 2009.] <http://eva.uci.cu/mod/resource/view.php?id=14094>.

## BIBLIOGRAFÍA

### Bibliografía

1 07. Gestion de Riesgos. [En línea] [Citado el: 23 de marzo de 2009.] [www.di.uniovi.es/~aquilino/Asignaturas/ProyectosInformatica/Documentos/07-GestionRiesgos.pdf](http://www.di.uniovi.es/~aquilino/Asignaturas/ProyectosInformatica/Documentos/07-GestionRiesgos.pdf).

RIESGOS DEL SOFTWARE. [En línea] [Citado el: 23 de marzo de 2009.] [www.sitios.uach.cl/caminosfor/CristianSalazar/SIE/RS.html](http://www.sitios.uach.cl/caminosfor/CristianSalazar/SIE/RS.html) - .

**Cabrera, Armando, Castillo, Luis Enriquez y Cuenca, Luis Eduardo.** Gestión del Riesgo. [En línea] mayo de 2008. [Citado el: 24 de marzo de 2009.] <http://www.slideshare.net/lecastillox/gestion-del-riesgo>.

**López Cabrera, Yanisleidy; Alvarez Lamas, Tailys.** Propuesta para la Gestión de Riesgo en los proyectos productivos de la UCI. [En línea] 2007. [Citado el: 10 de marzo de 2009.] [http://bibliodoc.uci.cu/TD/TD\\_0466\\_07.pdf](http://bibliodoc.uci.cu/TD/TD_0466_07.pdf).

Características del lenguaje Java. [En línea] [Citado el: 25 de marzo de 2009.] [www.iec.csic.es/CRIPTONOMICON/java/quesjava.html](http://www.iec.csic.es/CRIPTONOMICON/java/quesjava.html).

**Valdez Altamirano, Alfonso.** Comparativo de Entornos de Desarrollo Integrados. [En línea] [Citado el: 25 de marzo de 2009.] <http://bicuos.com/wp-content/uploads/2009/05/comparativoides.pdf> - .

1.0 DEFINICION DE HIBERNATE. [En línea] [Citado el: 3 de abril de 2009.] [www.unife.edu.pe/ing/desarrollo.doc](http://www.unife.edu.pe/ing/desarrollo.doc).

Fase de Inicio. Modelo del Negocio. [En línea] 2007-2008. [Citado el: 5 de abril de 2009.] <http://eva.uci.cu/mod/resource/view.php?id=11553>.

**Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El proceso unificado de desarrollo de software*. la Habana : Félix Varela, 2004. Volumen I.

Fase de Inicio. Flujo de trabajo de requerimientos. . [En línea] 2008-2009. [Citado el: 9 de junio de 2009.] <http://eva.uci.cu/mod/resource/view.php?id=12103>.

Flujo de Implementación. [En línea] [Citado el: 20 de marzo de 2009.] <http://eva.uci.cu/mod/resource/view.php?id=14094>.

**Hammer, Theodore; Rosenberg, Linda;.** Continuous Risk Management. [En línea] [Citado el: 15 de abril de 2009.]

**Murcia, Universidad de.** Gestión de riesgos en ingeniería del software . [En línea] 30 de diciembre de 2006. [Citado el: 22 de marzo de 2009.] <http://www.um.es/docencia/barzana/IAGP/lagp5.html>..

**Rubio, Gabriel Buades.** Ingeniería de Software III [Portal de la Universidad de Islas Baleares]. [En línea] [Citado el: 22 de marzo de 2009.] <http://dmi.uib.es/~bbuades/riesgos/index.htm..>

**CUEVA LOVELLE, JUAN MANUEL y JUAN FUENTE, AQUILINO ADOLFO.** Gestión de Riesgo [PDF digital]. [En línea] [Citado el: 23 de marzo de 2009.] <http://www.di.uniovi.es/~aquilino/Asignaturas/ProyectosInformatica/Documentos/Proyectos.v2006.C7.V2.pdf..>

## ANEXOS

Anexo 1. Descripción textual de los Casos de Uso del Negocio en formato expandido.

### Caso de Uso 1: Conectar base de datos

Caso de Uso:	Conectar base de datos
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el usuario abre el sistema y le aparece la interfaz para conectarse a la base de datos, introduciendo todos los datos necesarios.
Precondiciones:	
Referencias	R1
Poscondiciones	
Prioridad	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario inicia el sistema.	2. El sistema le muestra la interfaz para conectarse a la base de datos, mostrándoles los campos a llenar(nombBD ,puerto , host, usuario y contraseña)
3. El usuario introduce los datos (nombBD, puerto, host, usuario y contraseña) y selecciona el botón Conectar.	4. El sistema verifica completitud y correctitud de los datos y le muestra la interfaz para que el usuario se autentique.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4.1 En caso de que hayan errores, el sistema muestra un mensaje informando al usuario que debe llenar todos los campos correctamente.
4.2 El usuario llena todos los campos	4.3 El sistema muestra interfaz de

correctamente y selecciona el botón Conectar.	autenticación.
---	----------------

**Caso de uso 2: Autenticar usuario**

<b>Caso de Uso:</b>	Autenticar usuario
<b>Actores:</b>	Usuario
<b>Resumen:</b>	El caso de uso inicia cuando el usuario necesita autenticarse para poder acceder al sistema, entra su usuario y contraseña y accede al sistema. El sistema verifica que todos los datos necesarios hayan sido insertados y de forma correcta, dando acceso al sistema según nivel de acceso, finalizando el caso de uso.
<b>Precondiciones:</b>	El Usuario debe haberse conectado a la base de datos.
<b>Referencias</b>	R2
<b>Prioridad</b>	Crítico
<b>Poscondiciones</b>	El usuario entra al sistema

**Flujo Normal de Eventos**

Acción del Actor	Respuesta del Sistema
	1. El sistema muestra una interfaz para ingresar usuario y contraseña.
2. El usuario introduce usuario y contraseña.	3. El sistema verifica completitud y correctitud de datos.
	4. El sistema verifica el tipo de usuario.
	5.- Si corresponde a un "Administrador del Sistema" el sistema muestra la interfaz según el privilegio del usuario autenticado, finalizando el caso de uso.

**Flujos Alternos**

Acción del Actor	Respuesta del Sistema
	3.1 En caso de que los datos no estén correctos, el sistema muestra un



	mensaje informando que el usuario no puede acceder al sistema.
3.2 El usuario introduce los datos correctamente.	3.3 El sistema muestra la interfaz según el privilegio del usuario, finalizando el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	5.1 Si corresponde a un “Usuario” el sistema muestra la interfaz según el privilegio del usuario autenticado, finalizando el caso de uso.
<b>Prototipo de Interfaz</b>	<b>Tabla 2.1:</b> Descripción del Caso de Uso Autenticar Usuario.

### Caso de uso 3: Cambiar contraseña

<b>Caso de Uso:</b>	<b>Cambiar contraseña</b>
<b>Actores:</b>	<b>Usuario</b>
<b>Resumen:</b>	<b>El caso de uso se inicia cuando el usuario selecciona la opción Cambiar contraseña y el sistema muestra una interfaz para cambiar la contraseña.</b>
<b>Precondiciones:</b>	<b>El usuario debe estar autenticado.</b>
<b>Referencias</b>	<b>R3</b>
<b>Poscondiciones</b>	
<b>Prioridad</b>	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción Cambiar Contraseña	2. El sistema muestra la interfaz para cambiar contraseña con las opciones para introducir, el usuario, la contraseña anterior y la nueva.
3. El usuario introduce los datos y selecciona el botón Aceptar.	4. El sistema actualiza los cambios y muestra un mensaje informando el cambio realizado.

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4.1 En caso de que las contraseñas nuevas sean diferentes, el sistema muestra un mensaje informando que tiene que introducir las contraseñas correctamente.
	3.2 El sistema muestra la interfaz para que el usuario introduzca las contraseñas correctamente.
3.3 El usuario introduce los nuevos datos y selecciona el botón Aceptar.	

#### Caso de uso 4: Gestionar usuario

<b>Caso de Uso:</b>	<b>Gestionar Usuario</b>
<b>Actores:</b>	<b>Administrador del sistema</b>
<b>Resumen:</b>	El caso de uso se inicia cuando el Administrador indica adicionar, modificar o eliminar un usuario del sistema. El sistema muestra las interfaces correspondientes a cada una de estas opciones. El administrador realiza las acciones necesarias para que se adicione, modifique o elimine un usuario, finalizando el caso de uso.
<b>Precondiciones:</b>	<ul style="list-style-type: none"> <li>- El usuario autenticado debe ser administrador del sistema.</li> <li>- Deben existir usuarios insertados siempre que se desee modificar o eliminar usuarios del sistema.</li> </ul>
<b>Referencias</b>	R4
<b>Prioridad</b>	<b>Crítico</b>

#### Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el Administrador del sistema selecciona la opción Gestionar Usuario y las opciones para gestionar usuarios:	2. El sistema realiza una de las siguientes acciones: a) Si el administrador del sistema desea insertar un nuevo Usuario ver sesión "Adicionar Usuario"

<ul style="list-style-type: none"> <li>• Adicionar usuario.</li> <li>• Eliminar usuarios.</li> <li>• Modificar usuarios.</li> </ul>	<p>b) Si el administrador del sistema desea modificar un Usuario existente ver sección “Modificar Usuario”</p> <p>c) Si el administrador del sistema desea eliminar un Usuario existente ver sección “Eliminar Usuario”</p>
<b>Sección “Adicionar Usuario”</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador del sistema selecciona la opción Adicionar usuario.	2. El sistema muestra la interfaz para insertar nuevo usuario con todos los campos (usuario, contraseña, repetir contraseña, proyecto y rol).
3. El administrador del sistema llena los campos: (usuario, contraseña, repetir contraseña, proyecto y rol) y oprime el botón Aceptar.	4. El sistema verifica correctitud y completitud de los datos.
	5. El sistema adiciona el usuario y muestra un mensaje confirmando la inserción, finalizando el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	4.1 En caso de que algún campo esté vacío, el sistema muestra un mensaje de error advirtiéndole de que debe llenar todos los campos y regresa a la acción 3.

3.1 El administrador del sistema oprime el botón Cancelar.	3.2 El sistema muestra la interfaz inicial finalizando el caso de uso.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Eliminar Usuario”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador del sistema selecciona la opción Eliminar Usuario.	2. El sistema muestra la interfaz con la lista de usuarios existentes.
3. El administrador del sistema selecciona el usuario a eliminar y oprime el botón Aceptar.	4. El sistema elimina el usuario de la base de datos y muestra un mensaje informando que el usuario ha sido eliminado, finalizando el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	2.1 En caso de que no exista usuario en la base de datos, el sistema muestra un mensaje informando que no existe usuario a eliminar, finalizando el caso de uso.
3.1 El administrador del sistema oprime el botón Cancelar.	3.2 El sistema muestra la interfaz inicial finalizando el caso de uso.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Modificar Usuario”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador del sistema selecciona la opción Modificar Usuario.	2. El sistema muestra la interfaz con la lista de usuarios existentes en la base de datos y con los datos de los usuarios que pueden ser modificados (usuario, contraseña, repetir contraseña, proyecto y rol).
3. El administrador del sistema selecciona el usuario a modificar y los datos que desea modificar (usuario, contraseña, repetir contraseña, proyecto y rol) y oprime el botón Aceptar.	4. El sistema verifica correctitud y completitud de los datos.

		5. El sistema actualiza los cambios en la base de datos mostrando un mensaje de que todos los cambios se efectuaron satisfactoriamente, finalizando el caso de uso.
<b>Flujos Alternos</b>		
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>
		2.1 En caso de que no exista el usuario en la base de datos, el sistema muestra un mensaje informando que no existe el usuario a modificar.
3.1. El administrador del sistema oprime el botón Cancelar.		3.2 El sistema muestra la interfaz inicial finalizando el caso de uso.
		4.1 En caso de que los datos insertados estén incorrectos, el sistema muestra un mensaje de error y regresa a la acción 3.
<b>Poscondiciones</b>	Se actualizan los datos del usuario adicionado, modificado o eliminado.	
<b>Prototipo de Interfaz</b>		

#### Caso de uso 5: Identificar proyecto

<b>Caso de Uso:</b>	<b>Identificar Proyecto</b>
<b>Actores:</b>	<b>Administrador del sistema</b>
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario decide identificar proyecto. El sistema muestra en la interfaz la lista de proyectos por polo.
<b>Precondiciones:</b>	El usuario debe estar autenticado con los privilegios requeridos.
<b>Referencias</b>	R5
<b>Poscondiciones</b>	Cuando el usuario realiza una operación, los datos se registran en el proyecto identificado por el usuario.

<b>Prioridad</b>	<b>Secundario</b>
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El Administrador del sistema decide identificar proyecto.	2. El sistema muestra la interfaz con el listado de polos y el de proyectos por polo.
3. El Administrador del sistema selecciona el polo.	4. El sistema muestra el listado de proyectos por el polo seleccionado.
5. El Administrador del sistema selecciona el proyecto y oprime el botón Identificar.	6. El sistema muestra las opciones para que el administrador del sistema realice las acciones sobre el proyecto identificado.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	2.1 En caso de que no existan polos en la base de datos, el sistema muestra un mensaje de dicha inexistencia.
	4.1 En caso de que no existan proyectos por el polo seleccionado el sistema muestra un mensaje informando la inexistencia de los mismos y regresa a la acción 3.
5.1 El Administrador del sistema oprime el botón Cancelar.	5.2 El sistema regresa a la interfaz inicial, y finaliza el caso de uso.

#### Caso de uso 6: Valorar riesgos propuestos

<b>Caso de Uso:</b>	Valorar Riesgos propuestos.
<b>Actores:</b>	Administrador
<b>Resumen:</b>	El caso de uso se inicia cuando el administrador del sistema selecciona la opción valorar riesgos propuestos, el sistema le muestra un listado con dichos riesgos y el administrador valora si los inserta en la lista de riesgos o si los elimina.
<b>Precondiciones:</b>	El administrador debe estar autenticado.
<b>Referencias</b>	R6

Poscondiciones	Se actualizan los datos de los riesgos en la base de datos.	
Prioridad	Secundario	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
1. El administrador del sistema selecciona la opción de Valorar Riesgos Propuestos.		2. El sistema muestra la interfaz con los riesgos propuestos y las opciones de Aceptar, modificar o eliminar.
3. El Administrador del sistema selecciona una de las opciones para valorar los riesgos propuestos: <ul style="list-style-type: none"><li>• Aceptar.</li><li>• Eliminar.</li><li>• Modificar.</li></ul>		4 El sistema realiza una de las siguientes acciones: a)Si el administrador del sistema desea aceptar el riesgo propuesto ver sesión “Aceptar”  b)Si el administrador del sistema desea modificar un riesgo propuesto ver sección “Modificar”  c) Si el administrador del sistema desea eliminar riesgo propuesto “Eliminar ”
Flujos Alternos		
Acción del Actor		Respuesta del Sistema
		2.1 En caso de que no existan riesgos propuestos, el sistema muestra un mensaje informando la inexistencia de los mismos y se finaliza el caso de uso.
Sección “Aceptar”		
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
1. El administrador del sistema selecciona el riesgo y oprime el botón Aceptar.		2 El sistema guarda los cambios en la base de datos.

Flujo Normal de Eventos	
Sección “Eliminar”	
Acción del Actor	Respuesta del Sistema
1. El administrador del sistema selecciona el riesgo a eliminar y selecciona la opción Eliminar.	2. El sistema Guarda los cambios en la base de datos.
Flujo Normal de Eventos	
Sección “Modifica”	
Acción del Actor	Respuesta del Sistema
1. El administrador del sistema selecciona el riesgo a modificar y selecciona la opción Modificar	2. El sistema muestra la interfaz para que el usuario introduzca los datos del riesgo a modificar. (Clasificación existente o Nueva Clasificación y riesgo).
3. El administrador del sistema selecciona los datos que desea modificar (Clasificación existente o Nueva Clasificación y riesgo). Y oprime el botón aceptar.	4. El sistema verifica completitud y correctitud de los datos y los guarda.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4.1 En caso de que los datos insertados estén incorrectos, el sistema muestra un mensaje de error y regresa a la acción 3.
<b>Poscondiciones</b>	Se actualizan los datos del usuario adicionado, modificado o eliminado.
<b>Prototipo de Interfaz</b>	

### Caso de uso 7: Identificar riesgos

<b>Caso de Uso:</b>	Identificar Riesgos
<b>Actores:</b>	Usuario



Resumen:	El caso de uso inicia cuando el usuario indica identificar los riesgos. El sistema le muestra la interfaz para que seleccione los riesgos presentes en su proyecto y posteriormente genera un listado de riesgos identificados, finalizando el caso de uso.
Precondiciones:	El usuario debe estar autenticado.
Referencias	R7
Prioridad	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción Identificar Riesgos.	2. El sistema muestra en la interfaz las clasificaciones de los riesgos.
3. El usuario selecciona la clasificación de riesgo.	4. El sistema muestra una interfaz con los riesgos de la clasificación seleccionada por el usuario.
5. El usuario marca los riesgos y selecciona el botón Identificar.	6. El sistema guarda los riesgos en la base de datos.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4.1 En caso de que no existan riesgos por la clasificación seleccionada, el sistema muestra un mensaje informando la inexistencia de los mismos y regresa a la acción 3.
5.1 El usuario cancela la operación y regresa a la acción 3.	
Poscondiciones	Se crea una lista de riesgos identificados y se guardan en la base de datos.

#### Caso de uso 8: Analizar riesgos

<b>Caso de Uso:</b>	Analizar Riesgos
<b>Actores:</b>	Usuario
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario selecciona la opción Analizar Riesgos, donde le inserta la probabilidad y el

	impacto. El sistema calcula el nivel y la exposición genera una lista de riesgos analizados, finalizando el caso de uso.
<b>Precondiciones:</b>	El usuario debe estar autenticado, y debe existir una lista de riesgos ya identificados.
<b>Referencias</b>	R8
<b>Prioridad</b>	Secundario
<b>Poscondiciones</b>	Se crea una lista de riesgos analizados con los valores adicionados de probabilidad e impacto.

#### Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción Analizar Riesgos.	2. El sistema muestra una interfaz con la lista de riesgos identificados.
3. El usuario asigna para cada riesgo identificado un valor de probabilidad e impacto y oprime el botón Aceptar.	4. El sistema realiza los cálculos de exposición y nivel y muestra un mensaje con los valores calculados.
	5. El sistema guarda los cálculos en la base de datos y finaliza el caso de uso.

#### Flujos Alternos

Acción del Actor	Respuesta del Sistema
	2.1 En caso de que no existan riesgos identificados en el proyecto, el sistema muestra un mensaje informando la inexistencia de riesgos identificados finalizando el caso de uso.
3.1 El usuario oprime el botón Cancelar.	3.2 El sistema regresa a la interfaz inicial finalizando el caso de uso.

#### Caso de uso 9: Priorizar Riesgos Relevantes

<b>Caso de Uso:</b>	Priorizar Riesgos Relevantes
<b>Actores:</b>	Usuario
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario selecciona la opción Priorizar Riesgos de mayor relevancia, el sistema le muestra un listado con los riesgos ordenados de acuerdo al nivel y el usuario le inserta la prioridad.

Precondiciones:	El usuario debe haberse conectado a la base de datos y estar autenticado en la misma. Debe existir una lista de riesgos identificados, analizados y deben estar ordenados de acuerdo al nivel.	
Referencias	R9	
Poscondiciones	Se crea una nueva lista de riesgos ordenados por prioridad.	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
1. El usuario selecciona la opción Priorizar riesgos relevantes.		2. El sistema muestra la interfaz que le permite al usuario priorizar los riesgos.
3. El usuario introduce la cantidad de riesgos que desea priorizar.		
4. El usuario oprime el botón Mostrar Listado.		5. El sistema busca la lista de riesgos analizados a los que le hayan calculado el nivel.
		6. El sistema ordena la lista teniendo en cuenta el nivel.
7. El usuario introduce la prioridad y selecciona la opción Guardar.		
		8. El sistema guarda los cambios.
Flujos Alternos		
Acción del Actor		Respuesta del Sistema
		5.1 En caso de que no existan riesgos analizados ó a los que le hayan calculado el nivel, el sistema muestra un mensaje, informando dicho problema.
		5.2 El sistema muestra la interfaz para identificar un nuevo proyecto.
8.1 El usuario oprime el botón Cancelar.		8.2 El sistema muestra la interfaz inicial.

**Caso de uso 10: Proponer riesgos**

Caso de Uso:	Proponer Riesgos
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el usuario selecciona la opción Proponer Riesgos, el sistema muestra la interfaz para que adicione el nuevo riesgo y la nueva clasificación para este.
Precondiciones:	El usuario debe estar autenticado.
Referencias	R10
Poscondiciones	
Prioridad	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción Proponer riesgos.	2. El sistema muestra la interfaz para que el usuario introduzca los datos. (Clasificación existente o Nueva Clasificación y riesgo).
3. El usuario selecciona la opción Clasificación existente e introduce el nuevo riesgo y oprime el botón Aceptar.	4. El sistema verifica completitud y correctitud de los datos y los guarda.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3.1 El usuario selecciona la opción Nueva Clasificación e introduce el nuevo riesgo y oprime el botón Aceptar.	
3.2 El Usuario oprime el botón cancelar.	3.3 El sistema muestra la interfaz inicial y finaliza el caso de uso.
	4.1 En caso de que los datos insertados no estén correctos, el sistema muestra un mensaje de erros y regresa a la acción 3.

**Caso de uso 11: Mostrar documento de plan de contingencia**

Caso de Uso:	Mostrar documento de plan de contingencia.
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el usuario selecciona la opción Mostrar documento de plan de contingencia y el sistema muestra un archivo con el listado con los mismos.
Precondiciones:	El usuario debe estar autenticado.
Referencias	R11
Poscondiciones	Se muestra la lista de plan de contingencia.
Prioridad	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción Mostrar documento de plan de contingencia.	2. El sistema busca la lista de planes de contingencia identificadas y las muestra.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	2.1 En caso de que no existan planes de contingencia identificados, el sistema muestra un mensaje informando que no hay planes identificados en el proyecto.

**Caso de uso 12: Mostrar documento de gestión de riesgos**

<b>Caso de Uso:</b>	Mostrar documento de gestión de riesgos	
<b>Actores:</b>	Usuario	
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario selecciona la opción Mostrar documento de gestión de riesgos y el sistema muestra un listado con los mismos.	
<b>Precondiciones:</b>	El usuario debe estar autenticado.	
<b>Referencias</b>	R12	
<b>Poscondiciones</b>	Se muestra la lista de los riesgos.	
<b>Prioridad</b>	Secundario	

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción Mostrar documento de gestión de riesgos.	2. El sistema busca los riesgos y los muestra.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	2.1 En caso de que no existan riesgos, el sistema muestra un mensaje informando que no hay riesgos en el proyecto.
	2.2 El sistema muestra la interfaz inicial.

### Caso de uso 13: Gestionar polo productivo

<b>Caso de Uso:</b>	Gestionar Polo Productivo
<b>Actores:</b>	Administrador del sistema.
<b>Resumen:</b>	El caso de uso se inicia cuando el Administrador del sistema indica adicionar, modificar o eliminar un Polo Productivo. El sistema muestra las interfaces correspondientes a cada una de estas opciones. El administrador realiza las acciones necesarias para que se adicione, modifique o elimine un Polo productivo, finalizando el caso de uso.
<b>Precondiciones:</b>	El administrador debe estar autenticado y tener los permisos necesarios para realizar dicha acción.
<b>Referencias</b>	R13
<b>Prioridad</b>	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el Administrador del sistema selecciona la opción Nomencladores, la opción Gestionar Polo y selecciona una de las	2. El sistema muestra la interfaz según la opción seleccionada.  a) Si el administrador del sistema desea

<p>opciones:</p> <ul style="list-style-type: none"> <li>• Adicionar Polo.</li> <li>• Eliminar Polo.</li> <li>• Modificar Polo.</li> </ul>	<p>insertar un nuevo Polo Productivo ver sesión “Adicionar Polo”</p> <p>b) Si el administrador del sistema desea modificar un Polo Productivo existente ver sección “Modificar Polo”</p> <p>c) Si el administrador del sistema desea eliminar un Polo Productivo existente ver sección “Eliminar Polo”</p>
<b>Sección “Adicionar Polo Productivo”</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador del sistema selecciona la opción Adicionar Polo.	2. El sistema muestra la interfaz Adicionar Polo con todos los campos: nombre del polo, siglas y líder.
3. El administrador del sistema llena los campos: polo, siglas y líder y oprime el botón Aceptar.	4. El sistema verifica la completitud y correctitud de los datos.
	5. El sistema adiciona el polo y muestra un mensaje confirmando la inserción, finalizando el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	4.1 En caso de que algún campo esté vacío, el sistema muestra un mensaje de error advirtiéndole de que debe llenar todos los campos y regresa a la acción 3.
3.1 El administrador del sistema oprime el botón Cancelar.	3.2 El sistema muestra la interfaz inicial finalizando el caso de uso.

Flujo Normal de Eventos	
Sección “Eliminar Polo”	
Acción del Actor	Respuesta del Sistema
1. El administrador del sistema selecciona la opción Eliminar Polo.	2. El sistema muestra la interfaz con la lista de polos.
3. El administrador del sistema selecciona de la lista el polo a eliminar y presiona el botón Aceptar.	4. El sistema elimina el polo de la base de datos y muestra un mensaje informando que el polo ha sido eliminado, finalizando el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	2.1 En caso de que no exista polo en la base de datos, el sistema muestra un mensaje informando que no existe usuario a eliminar, finalizando el caso de uso.
3.1 El administrador del sistema oprime el botón Cancelar.	3.2 El sistema muestra la interfaz inicial finalizando el caso de uso.
Flujo Normal de Eventos	
Sección “Modificar Polo”	
Acción del Actor	Respuesta del Sistema
1. El administrador del sistema selecciona la opción Modificar Polo.	2. El sistema muestra la interfaz con la lista de polos existentes en la base de datos y con los datos de los proyectos que pueden ser modificados (polo, siglas y líder).
3. El administrador del sistema selecciona el polo y los datos que desea modificar oprime el botón Aceptar.	4. El sistema actualiza los cambios en la base de datos mostrando un mensaje de que todos los cambios se efectuaron satisfactoriamente, finalizando el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	2.1 En caso de que no exista el polo en la base de datos, el sistema muestra



	un mensaje informando que no existe el polo a modificar.
3.1. El administrador del sistema oprime el botón Cancelar.	3.2 El sistema muestra la interfaz inicial finalizando el caso de uso.
	4.1 En caso de que los datos insertados estén incorrectos, el sistema muestra un mensaje de error y regresa a la acción 3.
<b>Poscondiciones</b>	Se debe actualizar los cambios de los Polo Productivo adicionados, eliminados o modificados.

#### Caso de uso 14: Gestionar proyecto por polo

<b>Caso de Uso:</b>	Gestionar Proyecto por Polo
<b>Actores:</b>	Administrador del sistema.
<b>Resumen:</b>	El caso de uso se inicia cuando el Administrador del sistema indica adicionar, modificar o eliminar un proyecto. El sistema muestra las interfaces correspondientes a cada una de estas opciones. El administrador realiza las acciones necesarias para que se adicione, modifique o elimine un proyecto, finalizando el caso de uso.
<b>Precondiciones:</b>	El administrador debe estar autenticado y tener los permisos necesarios para realizar dicha acción.
<b>Referencias</b>	R14
<b>Prioridad</b>	Crítico

#### Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
2. El caso de uso inicia cuando el Administrador del sistema selecciona la opción Nomencladores, la opción Gestionar Proyecto y selecciona una de las opciones: <ul style="list-style-type: none"> <li>• Adicionar Proyecto.</li> <li>• Eliminar Proyecto.</li> </ul>	2. El sistema muestra la interfaz según la opción seleccionada. <p>a) Si el administrador del sistema desea insertar un nuevo Proyecto ver sesión "Adicionar Proyecto"</p> <p>b) Si el administrador del sistema desea</p>

<ul style="list-style-type: none"> <li>• Modificar Proyecto.</li> </ul>	<p>modificar un Proyecto existente ver sección “Modificar Proyecto”</p> <p>c) Si el administrador del sistema desea eliminar un Proyecto existente ver sección “Eliminar Proyecto”</p>
<b>Sección “Adicionar Proyecto por Polo”</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador del sistema selecciona la opción Adicionar Proyecto.	2. El sistema muestra la interfaz con los polos y con todos los campos a llenar: polo, proyecto, siglas y líder.
3. El administrador del sistema selecciona el polo e introduce los datos: polo, proyecto, siglas y líder y selecciona el botón Aceptar.	4. El sistema verifica completitud y correctitud de los datos.
	5. El sistema adiciona el proyecto y muestra un mensaje confirmando la inserción, finalizando el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	2.1 Si no existe polo, el sistema muestra un mensaje de dicha inexistencia, finalizando el caso de uso.
3.1 Si el administrador del sistema presiona el botón Cancelar.	3.2 El sistema muestra la interfaz inicial, finalizando el caso de uso.
	4.1 Si existen datos incorrectos o incompletos el sistema muestra un mensaje de error. Ir a la acción 3.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Eliminar Proyecto por Polo”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador del sistema selecciona la	2. El sistema muestra la interfaz con la

opción Eliminar Proyecto.	lista de polos.
3. El administrador del sistema selecciona el polo.	4. El sistema muestra el listado de proyectos por polo.
5. El administrador del sistema selecciona el proyecto por polo a eliminar y presiona el botón Aceptar.	6. El sistema muestra un mensaje de aviso, preguntando si está seguro que desea eliminar el proyecto encontrado.
7. El administrador del sistema presiona el botón Aceptar.	8. El sistema elimina el proyecto de la base de datos, finalizando el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	2.1 Si no existe proyecto, el sistema muestra un mensaje de dicha inasistencia, finalizando el caso de uso.
	4.1 En caso que no existan proyectos por el polo seleccionado, el sistema muestra un mensaje informando que no hay proyectos a eliminar y regresa a la acción 3.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
5.1 Si el administrador del sistema presiona el botón Cancelar.	5.2 El sistema muestra la interfaz inicial, finalizando el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
7.1 Si el administrador del sistema no desea eliminar, presiona el botón Cancelar.	7.2 El sistema muestra la interfaz inicial, y regresa a la acción 3.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Modificar Proyecto por Polo”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador del sistema selecciona la opción Modificar Proyectos.	2. El sistema muestra la interfaz con la lista de polos y de proyectos por polo.
3. El administrador del sistema selecciona el polo.	4. El sistema muestra el listado de proyectos por polo y los campos a

	modificar (proyecto, sigla, líder y polo).
5. El administrador del sistema selecciona el proyecto por polo a modificar.	7. El sistema verifica correctitud y completitud de los datos modificados.
	8. El sistema actualiza los cambios en la base de datos mostrando un mensaje de que todos los cambios se efectuaron satisfactoriamente, finalizando el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	2.1 Si no existe polo, el sistema muestra un mensaje de dicha inasistencia, finalizando el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	4.1 En caso que no existan proyectos por el polo seleccionada, el sistema muestra un mensaje informando que no hay proyectos a eliminar y regresa a la acción 3.
6.1 Si el administrador del sistema presiona el botón Cancelar y finaliza el caso de uso.	
	7.1 En caso de que los datos estén incorrectos, el sistema muestra un mensaje que debe corregir los datos insertados y regresa a la acción 6.
<b>Poscondiciones</b>	Se debe actualizar los cambios de los proyectos adicionados, eliminados o modificados.

### Caso de uso 15: Gestionar clasificación de riesgo

<b>Caso de Uso:</b>	Gestionar Clasificación de Riesgo
<b>Actores:</b>	Administrador del sistema
<b>Resumen:</b>	El caso de uso se inicia cuando el Administrador indica adicionar, modificar o eliminar una clasificación de riesgo. El sistema

	muestra las interfaces correspondientes a cada una de estas opciones. El administrador realiza las acciones necesarias para que se adicione, modifique o elimine una clasificación, finalizando el caso de uso.
<b>Precondiciones:</b>	- Deben existir clasificaciones insertados siempre que se desee modificar o eliminar clasificaciones del sistema.
<b>Referencias</b>	R15
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El caso de uso inicia cuando el Administrador del sistema selecciona la opción Nomencladores, la opción Gestionar Clasificación y selecciona una de las opciones: <ul style="list-style-type: none"> <li>• Adicionar Clasificación de Riesgo.</li> <li>• Eliminar Clasificación de Riesgo.</li> <li>• Modificar Clasificación de Riesgo.</li> </ul>	2. El sistema muestra la interfaz según la opción seleccionada.  a) Si el administrador del sistema desea insertar una nueva clasificación ver sección "Adicionar Clasificación de Riesgo"  b) Si el administrador del sistema desea modificar una clasificación existente ver sección "Modificar Clasificación de Riesgo"  c) Si el administrador del sistema desea eliminar una clasificación existente ver sección "Eliminar Clasificación de Riesgo"
<b>Sección "Adicionar Clasificación de Riesgo"</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador del sistema selecciona la opción Adicionar Clasificación de Riesgo.	2. El sistema muestra la interfaz para insertar la nueva clasificación con el campo: clasificación.

3. El administrador del sistema llena el campo de clasificación y oprime el botón Aceptar.	4. El sistema verifica completitud y correctitud de datos.
	5. El sistema adiciona la clasificación y muestra un mensaje confirmando la inserción, finalizando el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	4.1 En caso de que el campo este incorrecto, el sistema muestra un mensaje de error advirtiéndolo que los datos insertados no son correctos y regresa a la opción 3.
3.1 El administrador del sistema oprime el botón Cancelar.	3.2 El sistema muestra la interfaz inicial finalizando el caso de uso.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Eliminar Clasificación de Riesgo”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador del sistema selecciona la opción Eliminar Clasificación de Riesgo.	2. El sistema muestra la interfaz con la lista de clasificaciones existente en la base de datos.
3. El administrador del sistema selecciona la clasificación deseada y oprime el botón Aceptar.	4. El sistema elimina la clasificación de la base de datos, finalizando el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	2.1 En caso de que no existan clasificaciones en la base de datos, el sistema muestra un mensaje informando que no existen

	clasificaciones a eliminar, finalizando el caso de uso.
3.1 El administrador del sistema oprime el botón Cancelar.	3.2 El sistema muestra la interfaz inicial finalizando el caso de uso.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Modificar Clasificación de Riesgo”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador del sistema selecciona la opción Modificar Clasificación de Riesgo.	2. El sistema muestra la interfaz con la lista de clasificaciones y el dato que desee modificar.
3. El administrador del sistema selecciona la clasificación y adiciona el dato a modificar y oprime el botón Aceptar.	4. El sistema verifica correctitud y completitud de los datos.
	5. El sistema actualiza los cambios en la base de datos mostrando un mensaje de que todos los cambios se efectuaron satisfactoriamente, finalizando el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	2.1 En caso de que no exista clasificación en la base de datos, el sistema muestra un mensaje informando que no existe clasificación a modificar.
3.1. El administrador del sistema oprime el botón Cancelar.	3.2 El sistema muestra la interfaz inicial finalizando el caso de uso.
	4.1 En caso de que los datos insertados estén incorrectos, el sistema muestra un mensaje de error y regresa a la acción 3.
<b>Poscondiciones</b>	Se actualizan los datos de la clasificación adicionada, modificada o eliminada.
<b>Prototipo de Interfaz</b>	

**Caso de uso 16: Gestionar riesgo según clasificación**

<b>Caso de Uso:</b>	Gestionar Riesgo según Clasificación
<b>Actores:</b>	Administrador del sistema
<b>Resumen:</b>	El caso de uso se inicia cuando el Administrador indica adicionar, modificar o eliminar un riesgo. El sistema muestra las interfaces correspondientes a cada una de estas opciones. El administrador realiza las acciones necesarias para que se adicione, modifique o elimine un riesgo, finalizando el caso de uso.
<b>Precondiciones:</b>	- Deben existir riesgos por clasificación insertados siempre que se desee modificar o eliminar riesgos del sistema.
<b>Referencias</b>	R16
<b>Prioridad</b>	Crítico

**Flujo Normal de Eventos**

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>1. El caso de uso inicia cuando el Administrador del sistema selecciona la opción Nomencladores, la opción Gestionar Riesgo y selecciona una de las opciones:</p> <ul style="list-style-type: none"> <li>• Adicionar Riesgo.</li> <li>• Eliminar Riesgo.</li> <li>• Modificar Riesgo.</li> </ul>	<p>3. El sistema muestra la interfaz según la opción seleccionada.</p> <p>a) Si el administrador del sistema desea insertar un nuevo riesgo ver sesión “Adicionar riesgo”</p> <p>b) Si el administrador del sistema desea modificar un riesgo existente ver sección “Modificar riesgo”</p> <p>c) Si el administrador del sistema desea eliminar un riesgo existente ver sección “Eliminar riesgo”</p>

**Sección “Adicionar Riesgo”****Flujo Normal de Eventos**



Acción del Actor	Respuesta del Sistema
1. El administrador del sistema selecciona la opción Adicionar riesgo.	2. El sistema muestra la interfaz con las clasificaciones existentes, y el campo a llenar del nombre del riesgo.
3. El administrador del sistema selecciona la clasificación e introduce el nombre del riesgo y presiona el botón Aceptar.	4. El sistema verifica completitud y correctitud de datos.
	5. El sistema adiciona el riesgo y muestra un mensaje confirmando la inserción, finalizando el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	2.1 Si no existe clasificación, el sistema muestra un mensaje de dicha inexistencia, finalizando el caso de uso.
3.1 Si el administrador del sistema presiona el botón Cancelar.	3.2 El sistema muestra la interfaz inicial, finalizando el caso de uso.
	4.1 Si existen datos incorrectos o incompletos el sistema muestra un mensaje de error. Ir a la acción 3.
Flujo Normal de Eventos	
Sección “Eliminar Riesgo”	
Acción del Actor	Respuesta del Sistema
1. El administrador del sistema selecciona la opción Eliminar Riesgo.	2. El sistema muestra la interfaz con la lista de clasificación.
3. El administrador del sistema selecciona la clasificación.	4. El sistema muestra el listado de riesgos por clasificación.
5. El administrador del sistema selecciona el riesgo por clasificación a eliminar y presiona el botón Aceptar.	6. El sistema muestra un mensaje de aviso, preguntando si está seguro que desea eliminar el riesgo encontrado.

7. El administrador del sistema presiona el botón Aceptar.	8. El sistema elimina el riesgo de la base de datos, finalizando el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	2.1 Si no existe clasificación, el sistema muestra un mensaje de dicha inasistencia, finalizando el caso de uso.
	4.1 En caso que no existan riesgos por la clasificación seleccionada, el sistema muestra un mensaje informando que no hay riesgos a eliminar y regresa a la acción 3.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
5.1 Si el administrador del sistema presiona el botón Cancelar.	5.2 El sistema muestra la interfaz inicial, finalizando el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
7.1 Si el administrador del sistema no desea eliminar, presiona el botón Cancelar.	7.2 El sistema muestra la interfaz inicial, y regresa a la acción 3.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Modificar Riesgo”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador del sistema selecciona la opción Modificar Riesgo.	2. El sistema muestra la interfaz con la lista de clasificación y de riesgos por clasificación.
3. El administrador del sistema selecciona la clasificación.	4. El sistema muestra el listado de riesgos por clasificación y los campos a modificar (nombre y clasificación).
5. El administrador del sistema selecciona el riesgo por clasificación a modificar.	
6. El administrador del sistema selecciona	7. El sistema verifica correctitud y

los campos (nombre y/o clasificación) a modificar y presiona el botón Aceptar.	completitud de los datos modificados.
	8. El sistema actualiza los cambios en la base de datos mostrando un mensaje de que todos los cambios se efectuaron satisfactoriamente, finalizando el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	2.1 Si no existe clasificación, el sistema muestra un mensaje de dicha inasistencia, finalizando el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	4.1 En caso que no existan riesgos por la clasificación seleccionada, el sistema muestra un mensaje informando que no hay riesgos a eliminar y regresa a la acción 3.
6.1 Si el administrador del sistema presiona el botón Cancelar y finaliza el caso de uso.	
	7.1 En caso de que los datos estén incorrectos, el sistema muestra un mensaje que debe corregir los datos insertados y regresa a la acción 6.
<b>Poscondiciones</b>	Se actualizan los datos del riesgo adicionado, modificado o eliminado.
<b>Prototipo de Interfaz</b>	

### Caso de uso 17: Gestionar plan de contingencia

<b>Caso de Uso:</b>	Gestionar Plan de Contingencia
<b>Actores:</b>	Administrador del sistema

<b>Resumen:</b>	El caso de uso se inicia cuando el Administrador indica adicionar, modificar o eliminar un plan de contingencia. El sistema muestra las interfaces correspondientes a cada una de estas opciones. El administrador realiza las acciones necesarias para que se adicione, modifique o elimine un plan de contingencia, finalizando el caso de uso.
<b>Precondiciones:</b>	- Deben existir planes de contingencia insertadas siempre que se desee modificar o eliminar tipo de plan de contingencia del sistema.
<b>Referencias</b>	R17
<b>Prioridad</b>	Crítico

#### Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el Administrador del sistema selecciona la opción Nomencladores, la opción Gestionar Plan de Contingencia y selecciona una de las opciones: <ul style="list-style-type: none"> <li>• Adicionar plan de contingencia.</li> <li>• Eliminar plan de contingencia.</li> <li>• Modificar plan de contingencia.</li> </ul>	2. El sistema muestra la interfaz según la opción seleccionada.  a) Si el administrador del sistema desea insertar un nuevo plan de contingencia ver sesión “Adicionar plan de contingencia”  b) Si el administrador del sistema desea modificar un plan de contingencia existente ver sección “Modificar plan de contingencia”  c) Si el administrador del sistema desea eliminar un plan de contingencia existente ver sección “Eliminar plan de contingencia”

#### Sección “Adicionar plan de contingencia”

#### Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El administrador del sistema selecciona la	2. El sistema muestra la interfaz

opción Adicionar plan de contingencia.	con la lista de clasificación, lista de riesgos, y el campo a llenar del nombre del plan.
3. El administrador del sistema selecciona la clasificación.	4 El sistema muestra la lista de riesgos para la clasificación seleccionada.
5. El administrador del sistema selecciona el y riesgo e introduce el nombre del plan y presiona el botón Aceptar.	6. El sistema verifica completitud y correctitud de datos.
	7. El sistema adiciona el plan y muestra un mensaje confirmando la inserción, finalizando el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	2.1 Si no existe clasificación de riesgo, el sistema muestra un mensaje de dicha inexistencia, finalizando el caso de uso.
	4.1 Si no existe riesgo, el sistema muestra un mensaje de dicha inexistencia, finalizando el caso de uso.
5.1 Si el administrador del sistema presiona el botón Cancelar.	5.2 El sistema muestra la interfaz inicial, finalizando el caso de uso.
	6.1 Si existen datos incorrectos o incompletos el sistema muestra un mensaje de error. Ir a la acción 3.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Eliminar plan de contingencia”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador del sistema selecciona la opción Eliminar plan de contingencia.	2. El sistema muestra la interfaz con la lista de clasificación, lista de riesgos, y la lista de plan.

3. El administrador del sistema selecciona la clasificación.	4 El sistema muestra la lista de riesgos para la clasificación seleccionada.
5. El administrador del sistema selecciona el y riesgo.	6. El sistema muestra la lista de plan que pertenece al riesgo seleccionado.
7. El administrador del sistema selecciona el plan a eliminar y presiona el botón Aceptar.	8. El sistema elimina el plan y muestra un mensaje confirmando la inserción, finalizando el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	2.1 Si no existe clasificación de riesgo, el sistema muestra un mensaje de dicha inexistencia, finalizando el caso de uso.
	4.1 Si no existe riesgo, el sistema muestra un mensaje de dicha inexistencia, finalizando el caso de uso.
	6.1 Si no existe plan, el sistema muestra un mensaje de dicha inexistencia, finalizando el caso de uso.
7.1 Si el administrador del sistema presiona el botón Cancelar.	7.2 El sistema muestra la interfaz inicial, finalizando el caso de uso.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Modificar Plan de Contingencia”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador del sistema selecciona la opción Modificar plan de contingencia.	2. El sistema muestra la interfaz con la lista de clasificaciones, de riesgos y de plan.
3 El administrador del sistema selecciona la clasificación.	4 El sistema muestra la lista de riesgos que pertenecen a la clasificación seleccionada.
5. El administrador del sistema selecciona el riesgo.	6. El sistema muestra el listado de plan de contingencia por riesgo y los campos a modificar (nombre y riesgo).

7. El administrador del sistema selecciona el plan de contingencia por riesgo a modificar.	
8. El administrador del sistema selecciona los campos (nombre y/o riesgo) a modificar y presiona el botón Aceptar.	9. El sistema verifica correctitud y completitud de los datos modificados.
	10. El sistema actualiza los cambios en la base de datos mostrando un mensaje de que todos los cambios se efectuaron satisfactoriamente, finalizando el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	2.1 Si no existe clasificación, el sistema muestra un mensaje de dicha inasistencia, finalizando el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	4.1 En caso que no existan riesgos por la clasificación seleccionada, el sistema muestra un mensaje informando la inexistencia y regresa a la acción 3.
	6.1 En caso que no existan plan por el riesgo seleccionado, el sistema muestra un mensaje informando la inexistencia y regresa a la acción 3.
8.1 Si el administrador del sistema presiona el botón Cancelar y finaliza el caso de uso.	9.1 En caso de que los datos estén incorrectos, el sistema muestra un mensaje que debe corregir los datos insertados y regresa a la acción 6.
<b>Poscondiciones</b>	Se actualizan los datos del plan de contingencia adicionado, modificado o eliminado.
<b>Prototipo de Interfaz</b>	

**Caso de uso 18: Gestionar rol**

<b>Caso de Uso:</b>	Gestionar Rol
<b>Actores:</b>	Administrador del sistema
<b>Resumen:</b>	El caso de uso se inicia cuando el Administrador indica adicionar, modificar o eliminar un rol. El sistema muestra las interfaces correspondientes a cada una de estas opciones. El administrador realiza las acciones necesarias para que se registre, modifique o elimine un rol, finalizando el caso de uso.
<b>Precondiciones:</b>	- Deben existir roles insertados siempre que se desee modificar o eliminar roles del sistema.
<b>Referencias</b>	R18
<b>Prioridad</b>	Crítico

**Flujo Normal de Eventos**

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El caso de uso inicia cuando el Administrador del sistema selecciona la opción Nomencladores, la opción Gestionar rol y selecciona una de las opciones: <ul style="list-style-type: none"> <li>• Adicionar rol.</li> <li>• Eliminar rol.</li> <li>• Modificar rol.</li> </ul>	2. El sistema muestra una interfaz con las opciones a seleccionar: <p>a) Si el administrador del sistema desea insertar un nuevo rol ver sección "Adicionar rol"</p> <p>b) Si el administrador del sistema desea modificar un rol existente ver sección "Modificar rol"</p> <p>c) Si el administrador del sistema desea eliminar un rol existente ver sección "Eliminar rol"</p>

**Sección "Adicionar Rol"****Flujo Normal de Eventos**

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
-------------------------	------------------------------



1. El administrador del sistema selecciona la opción Adicionar rol.	2. El sistema muestra la interfaz para insertar rol con el campo: tipo.
3. El administrador del sistema llena el campo rol y oprime el botón Aceptar.	4. El sistema verifica la completitud y correctitud del dato.
	5. El sistema adiciona el rol y muestra un mensaje confirmando la inserción, finalizando el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
3.1 El administrador del sistema oprime el botón Cancelar.	3.2 El sistema muestra la interfaz inicial finalizando el caso de uso.
	4.1 En caso de que los datos estén incorrectos, el sistema muestra un mensaje de error advirtiendo el error y regresa a la acción 3.
<b>Flujo Normal de Eventos</b>	
<b>Sección "Eliminar Rol"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador del sistema selecciona la opción Eliminar Rol.	2. El sistema muestra la interfaz con la lista de rol.
3. El administrador del sistema selecciona el rol que desea eliminar y oprime el botón Aceptar.	4. El sistema elimina el rol de la base de datos, finalizando el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	2.1 En caso de que no exista rol en la base de datos, el sistema muestra un mensaje informando que no existe usuario a eliminar, finalizando el caso de uso.
3.1 El administrador del sistema oprime el	3.2 El sistema muestra la interfaz inicial

botón Cancelar.	finalizando el caso de uso.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Modificar Rol”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador del sistema selecciona la opción Modificar Rol.	2. El sistema muestra la interfaz con la lista de roles y el dato que desee modificar: tipo de rol.
3. El administrador del sistema selecciona el rol y adiciona el nuevo dato y oprime el botón Aceptar.	4. El sistema verifica correctitud y completitud de los datos.
	5. El sistema actualiza los cambios en la base de datos mostrando un mensaje de que todos los cambios se efectuaron satisfactoriamente, finalizando el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	2.1 En caso de que no exista el rol en la base de datos, el sistema muestra un mensaje informando que no existe el usuario a modificar.
3.1. El administrador del sistema oprime el botón Cancelar.	3.2 El sistema muestra la interfaz inicial finalizando el caso de uso.
	4.1 En caso de que los datos insertados estén incorrectos, el sistema muestra un mensaje de error y regresa a la acción 3.
<b>Poscondiciones</b>	Se actualizan los datos del rol adicionado, modificado o eliminado.
<b>Prototipo de Interfaz</b>	

## Anexo 2. Diagramas de clases del diseño

Diagrama de clases del diseño DCD\_Analizar Riesgos

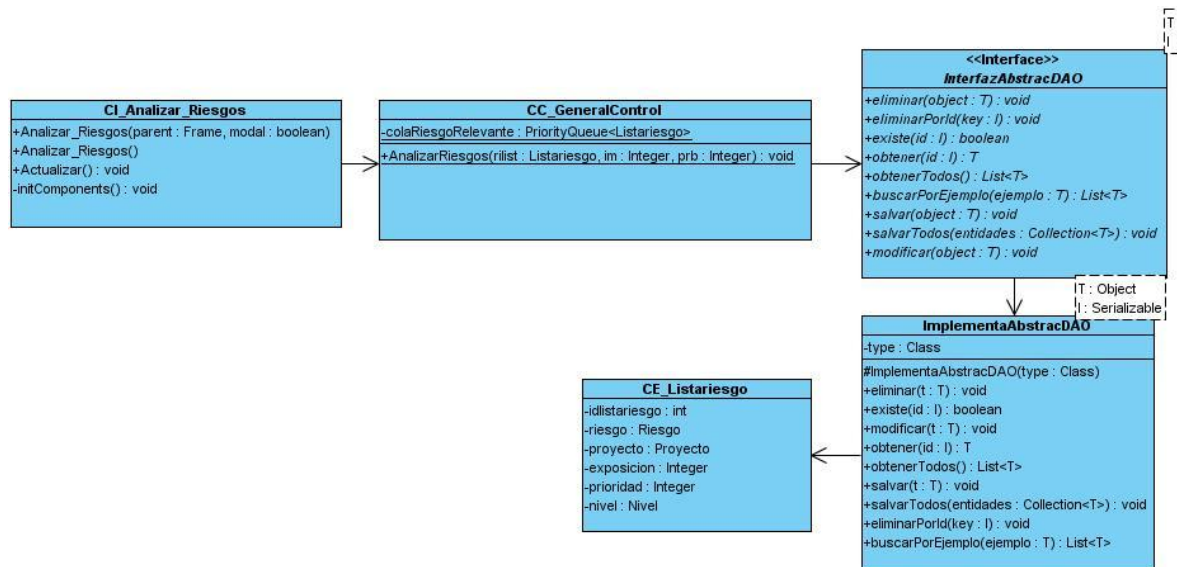


Diagrama de clases del diseño DCD\_Identificar Riesgos



Diagrama de clases del diseño DCD\_Autenticar usuarios

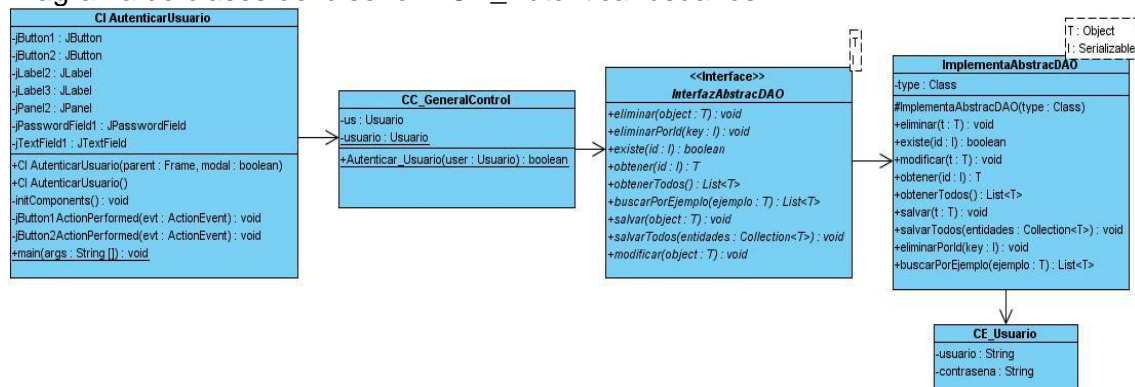


Diagrama de clases del diseño DCD\_Gestionar Clasificación por Riesgos

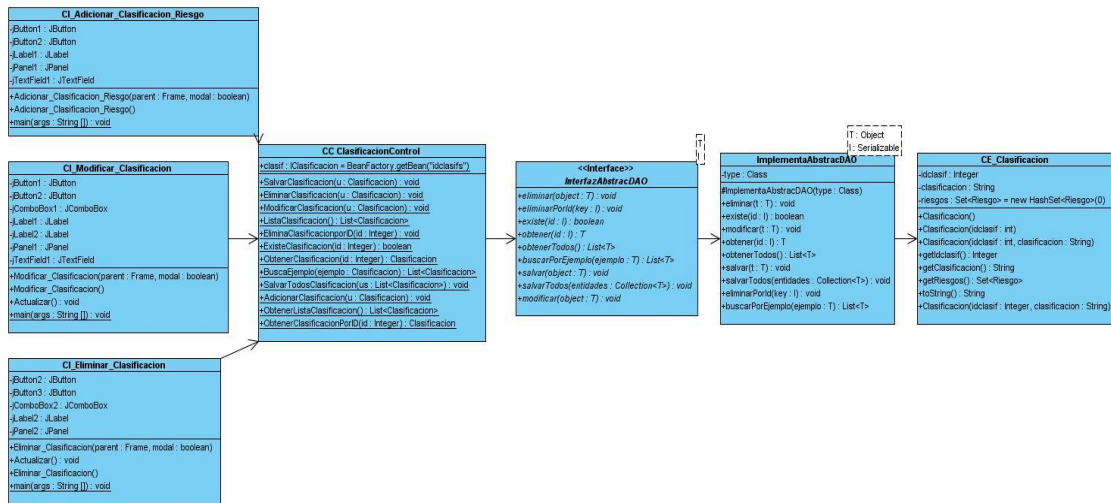


Diagrama de clases del diseño DCD\_Gestionar Plan de Contingencia

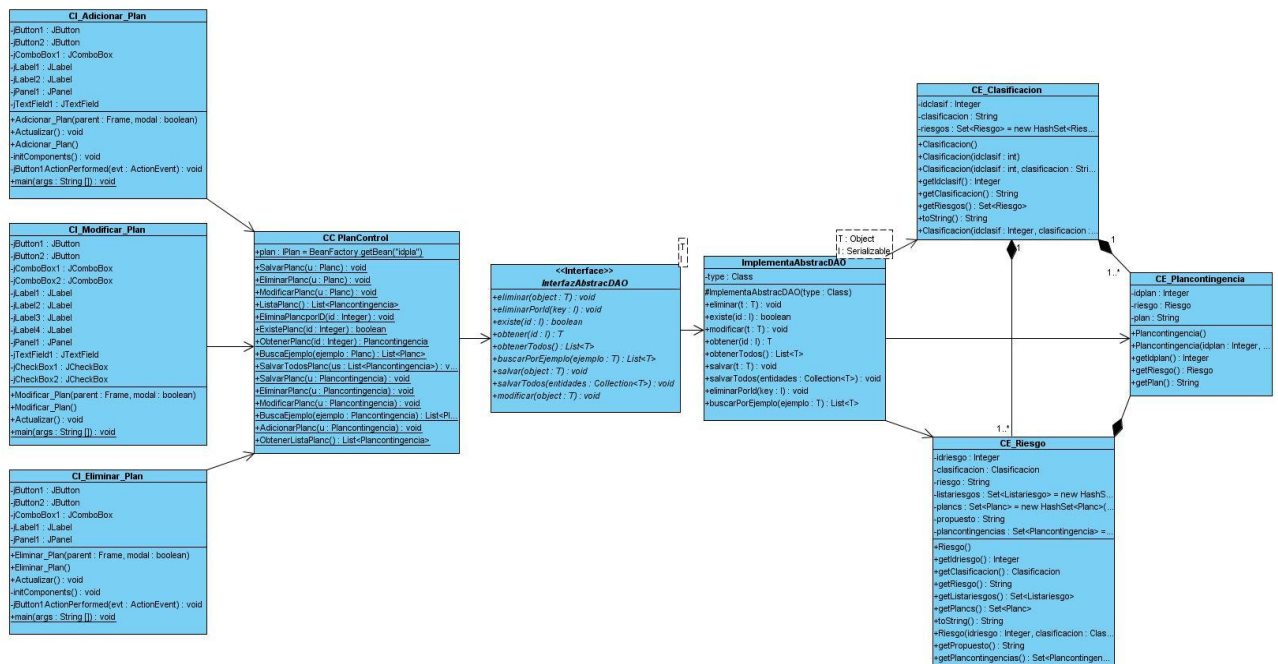


Diagrama de clases del diseño DCD\_Gestionar Polo Productivo

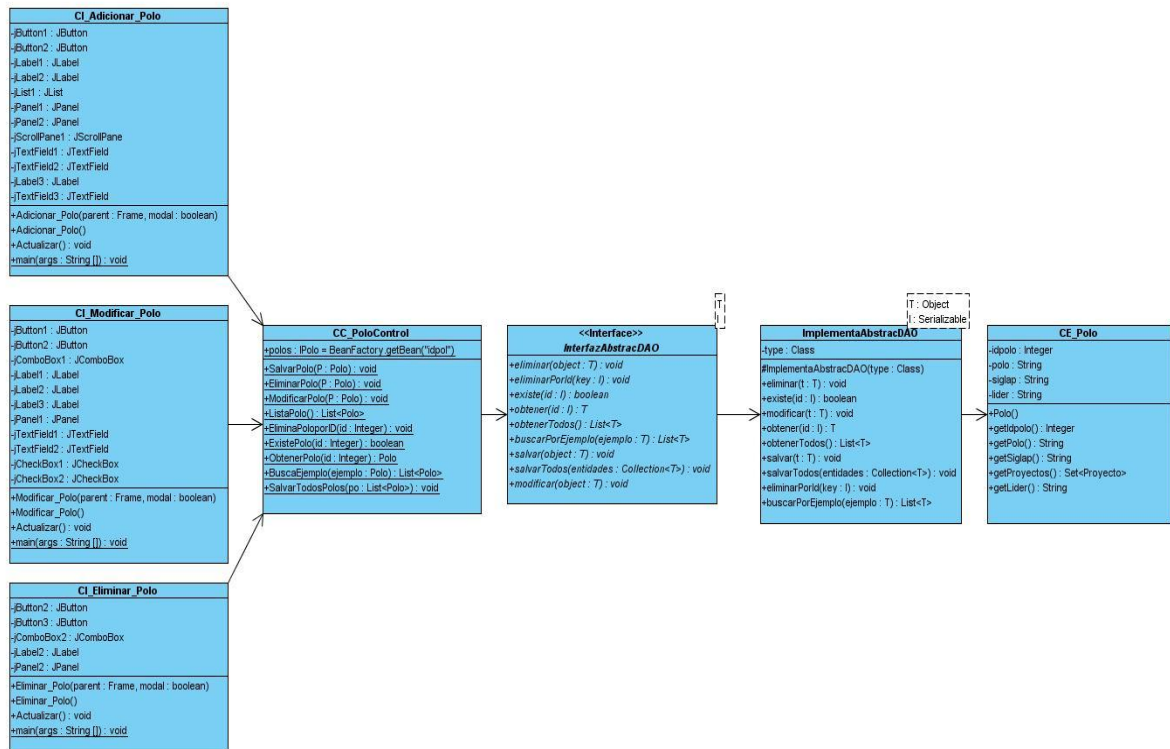


Diagrama de clases del diseño DCD\_Gestionar Polo por Proyecto

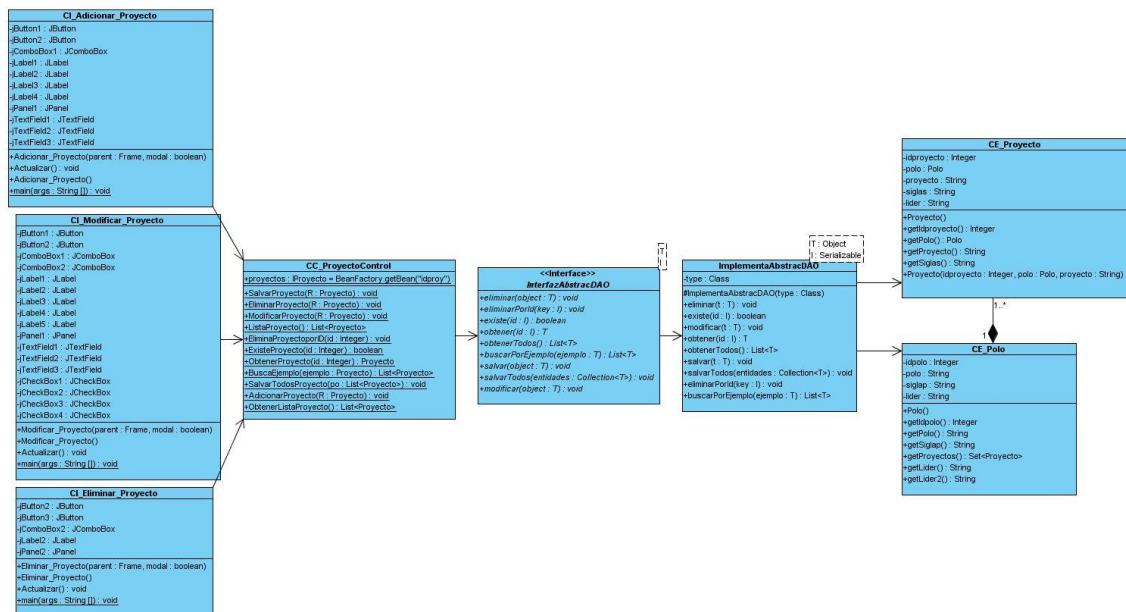


Diagrama de clases del diseño DCD\_Gestionar rol



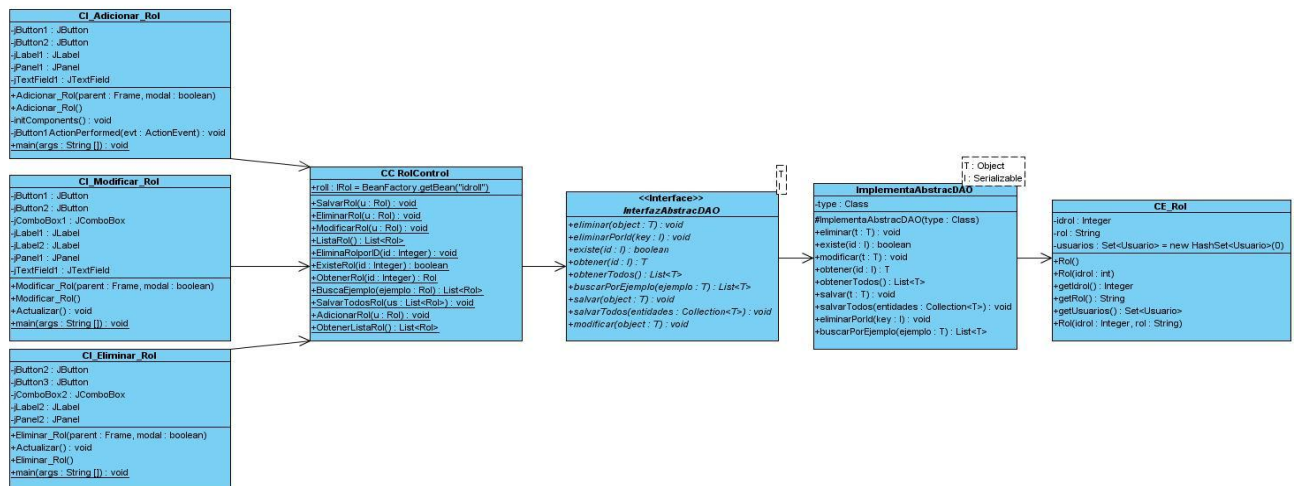


Diagrama de clases del diseño DCD\_Mostrar Doc Gestión Riesgos

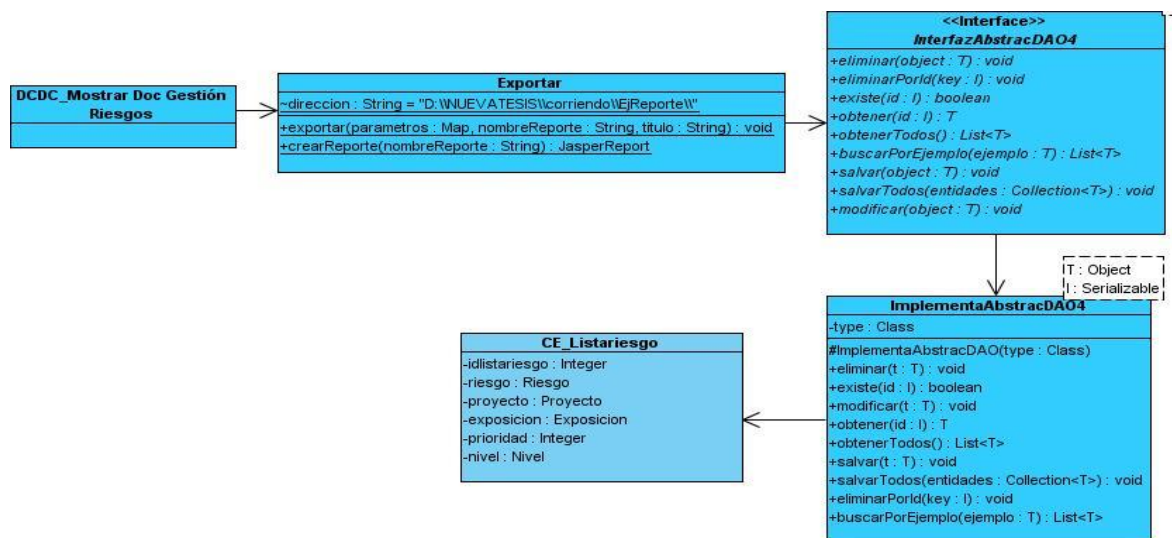


Diagrama de clases del diseño DCD\_Mostrar Doc Plan de Contingencia

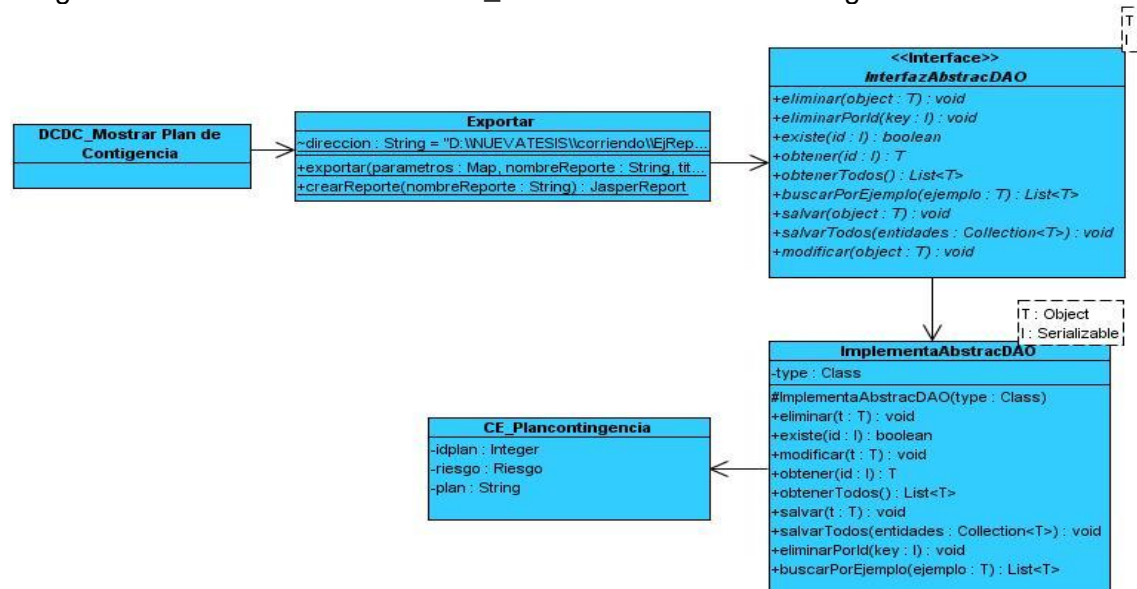
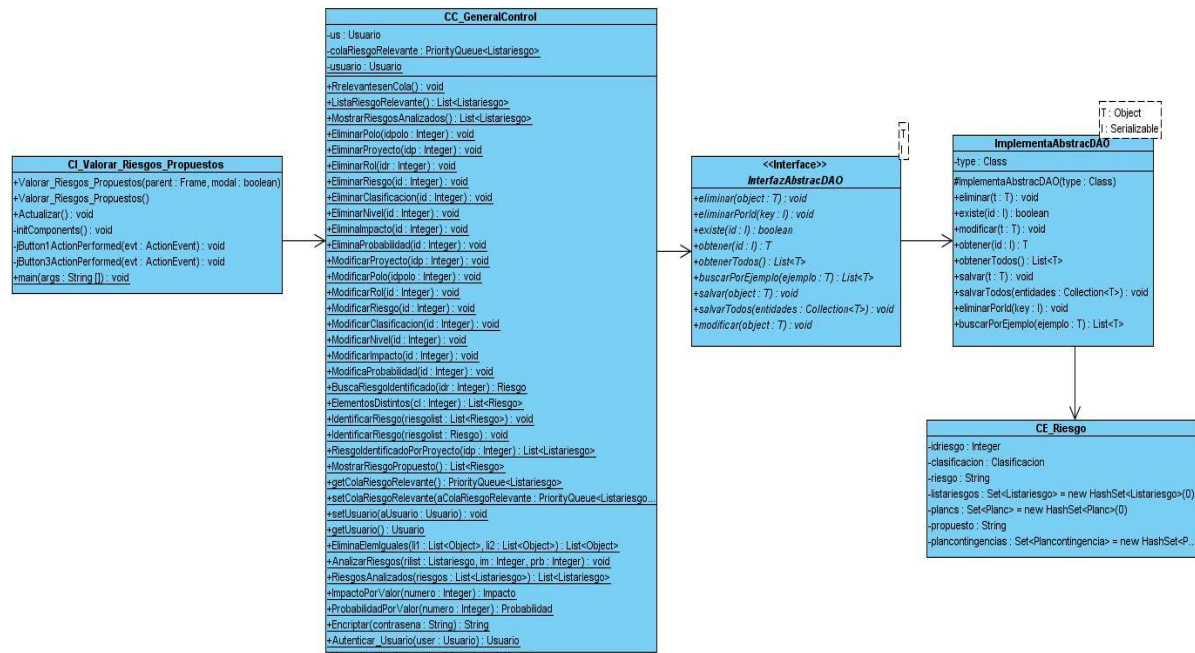
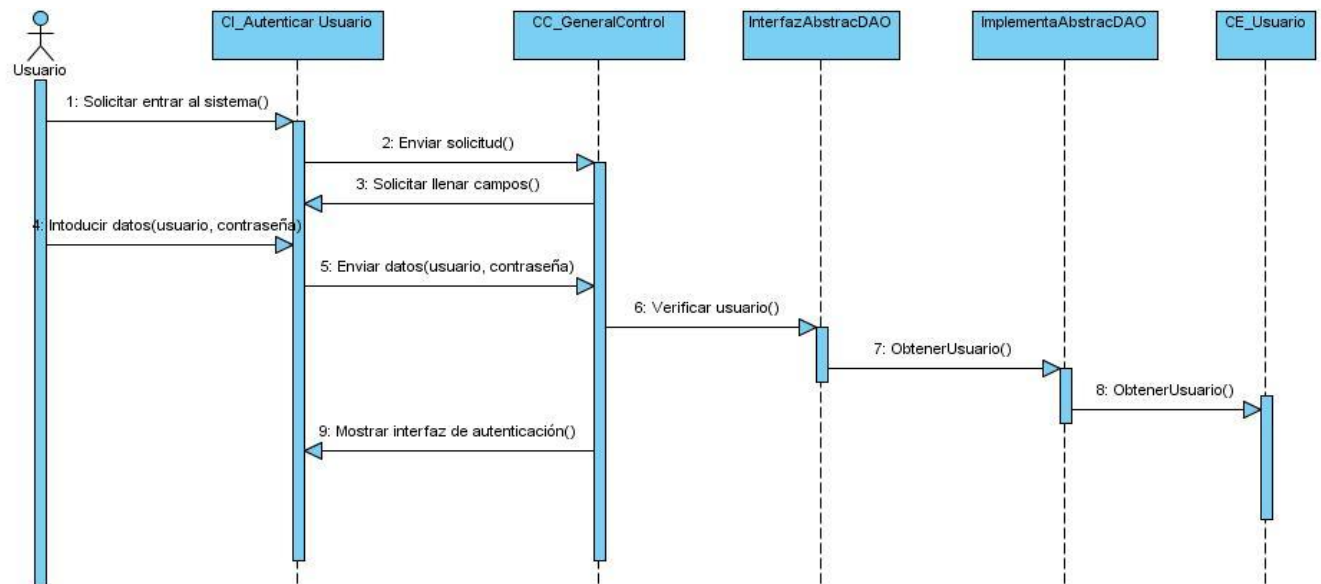


Diagrama de clases del diseño DCD\_Valorar Riesgos Propuestos

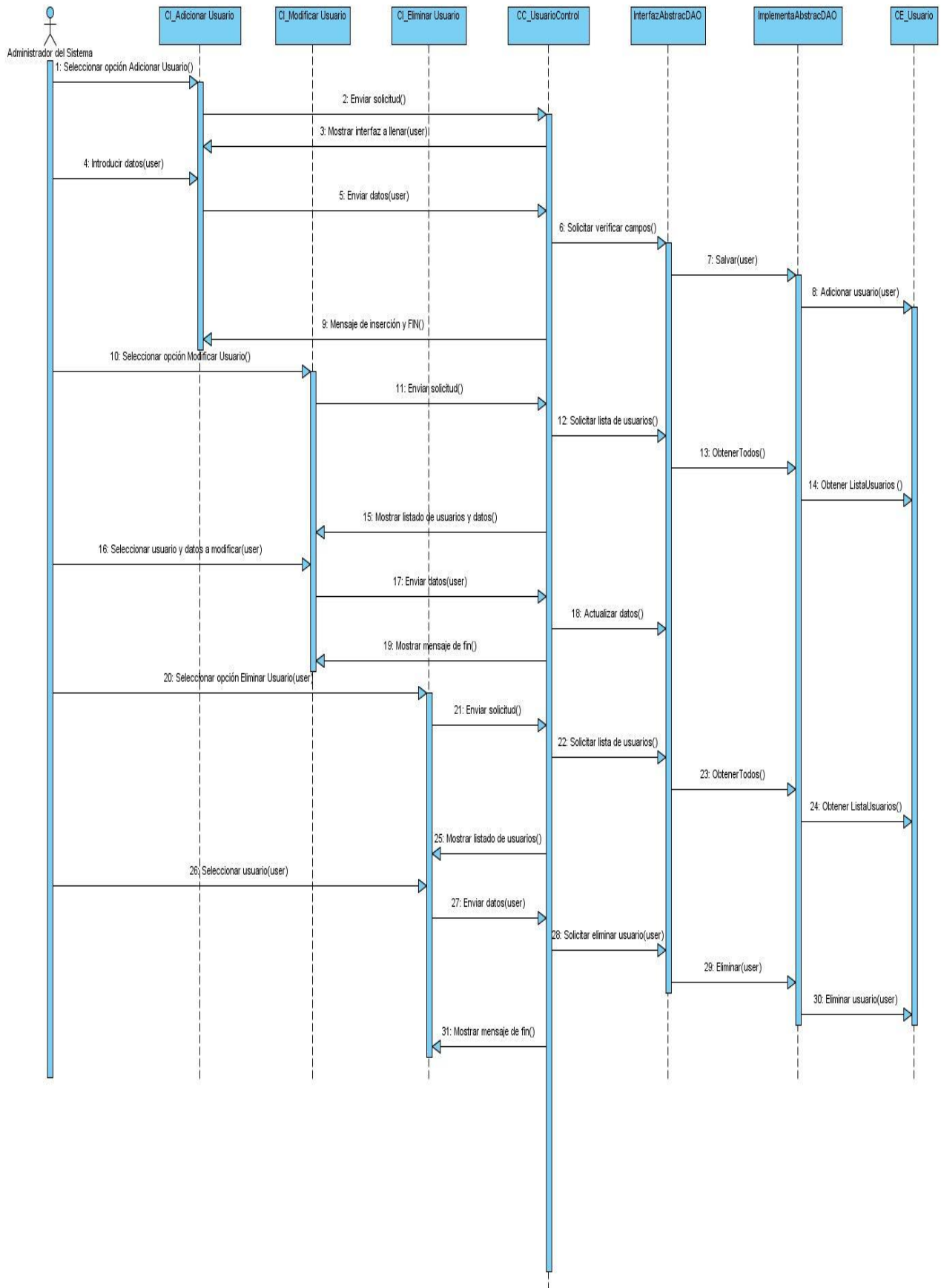


## Anexo 3 Diagramas de Secuencia

Diagrama de Secuencia CU Autenticar usuario



## Diagrama de Secuencia CU Gestionar Usuario





## Diagrama de Secuencia CU Gestionar Clasificación

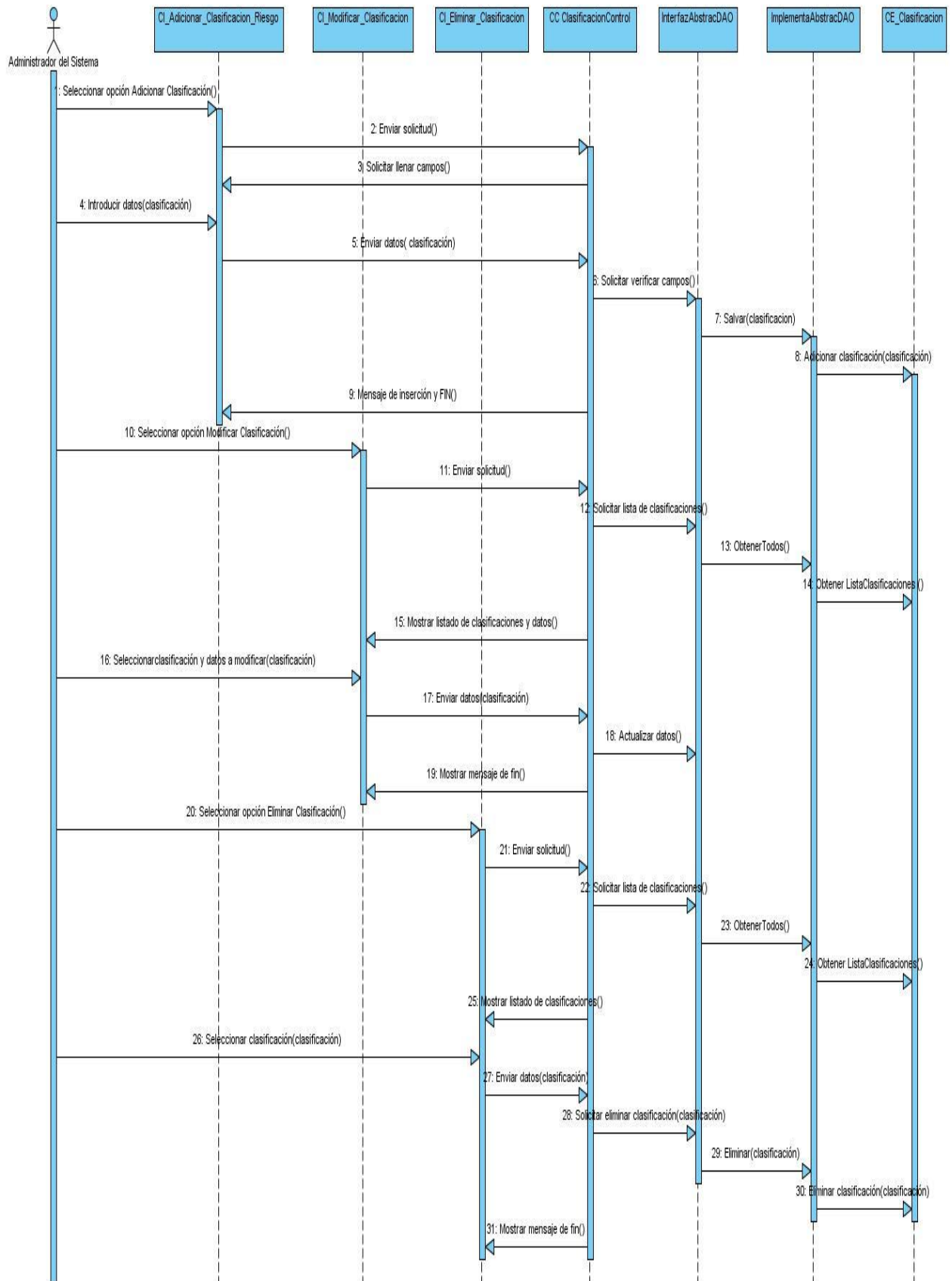


Diagrama de Secuencia CU Gestionar Polo

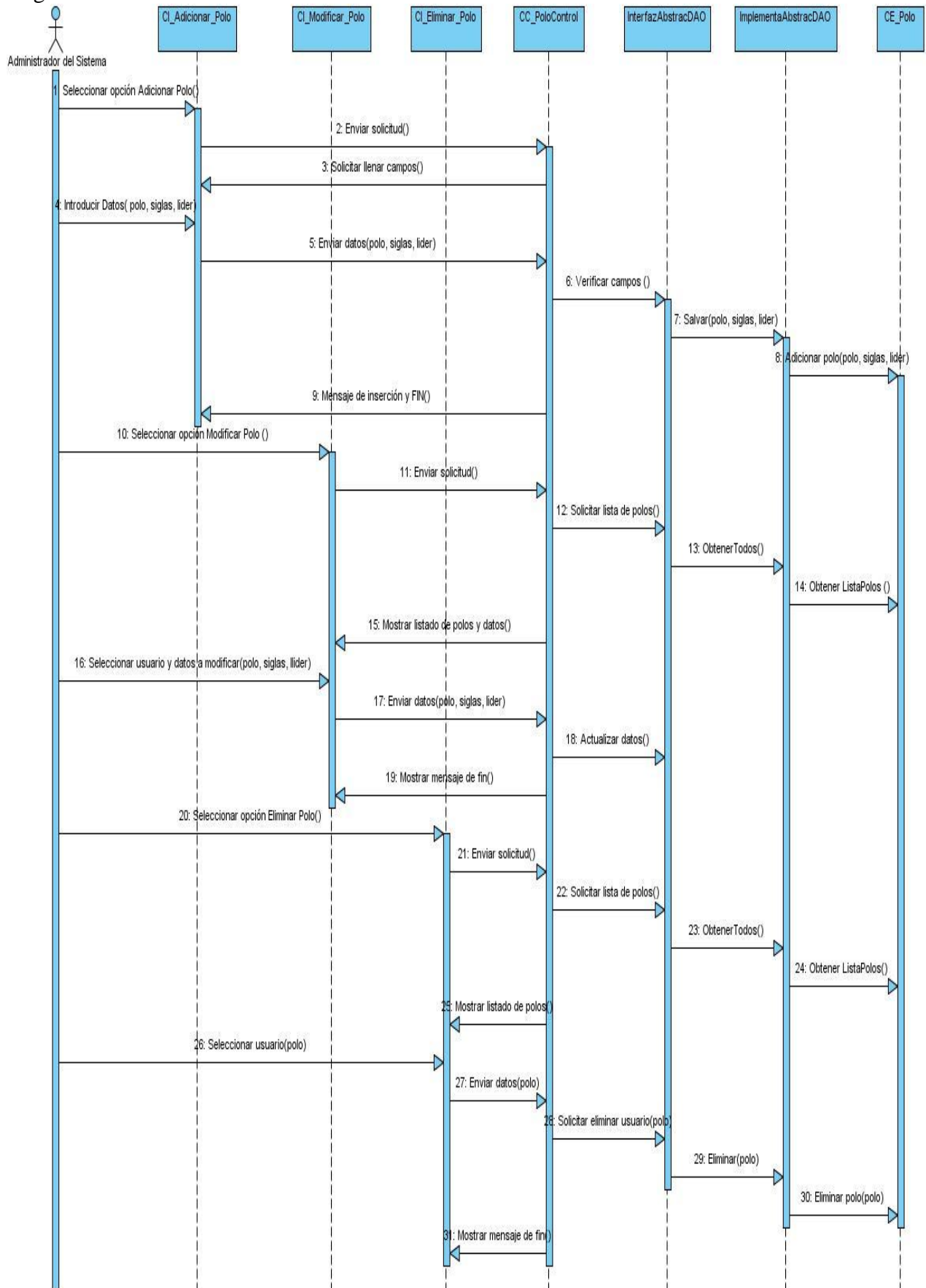
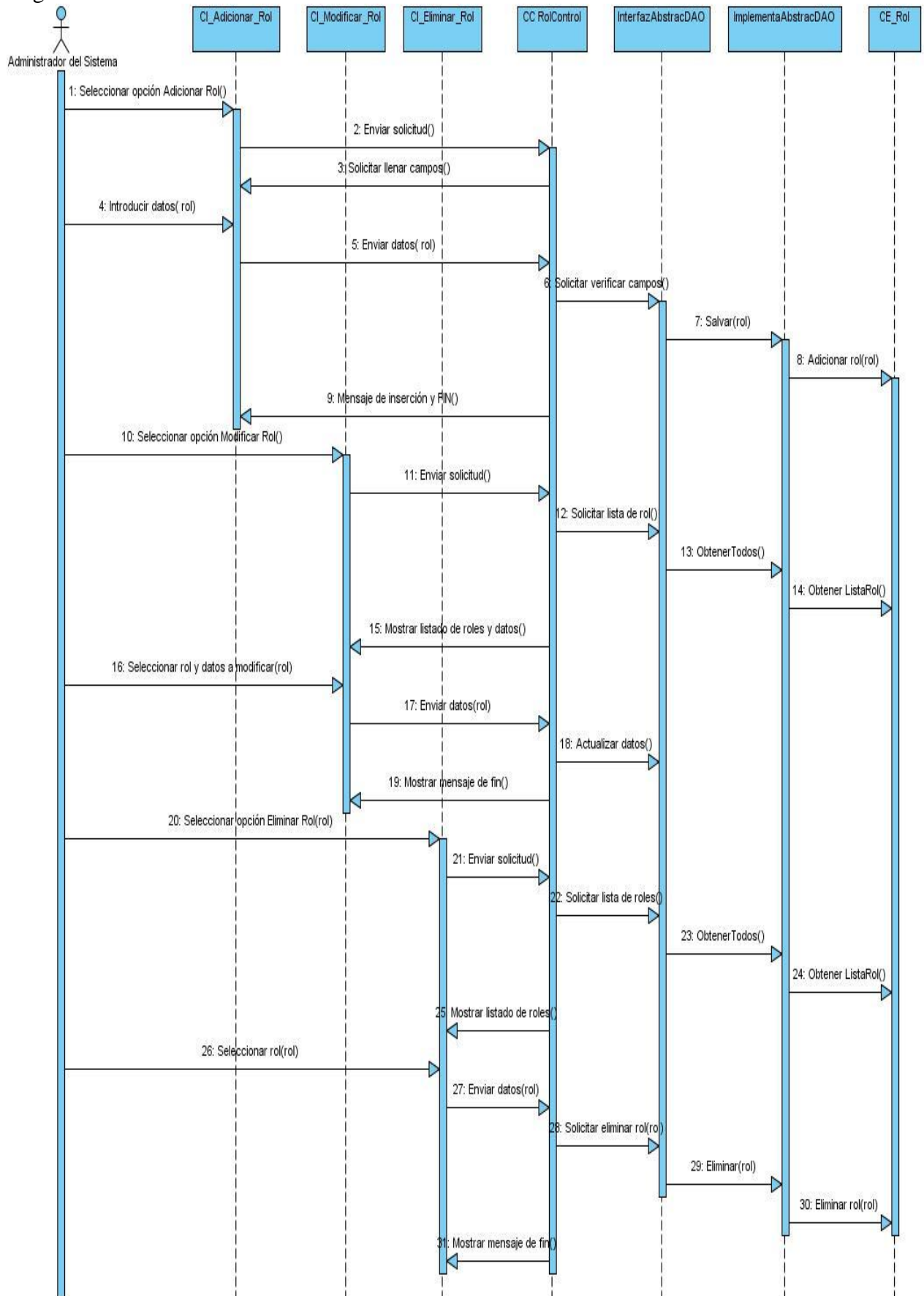
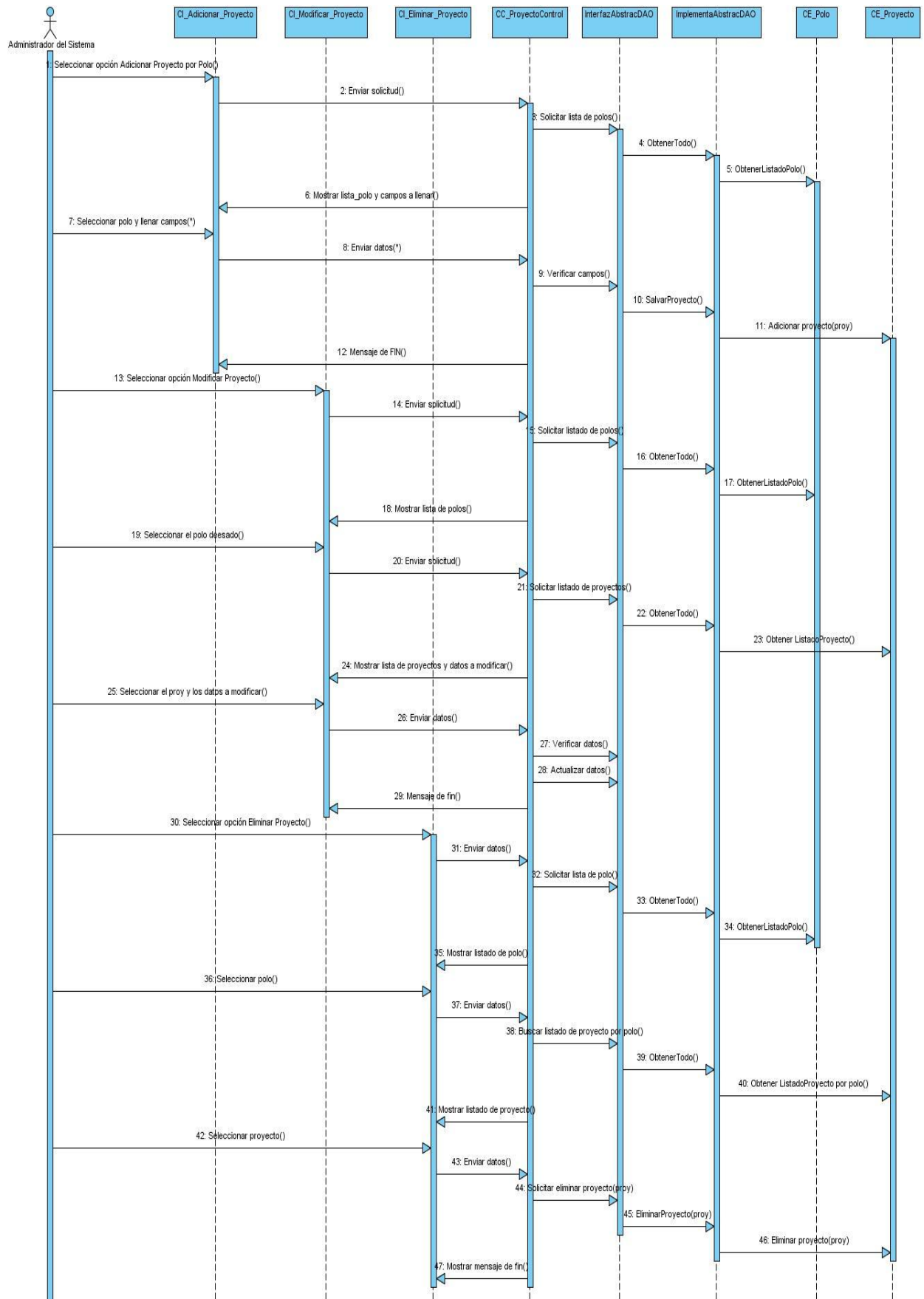


Diagrama de Secuencia CU Gestionar Rol



## Diagrama de Secuencia CU Gestionar Proyecto



## Diagrama de Secuencia CU Gestionar Riesgos

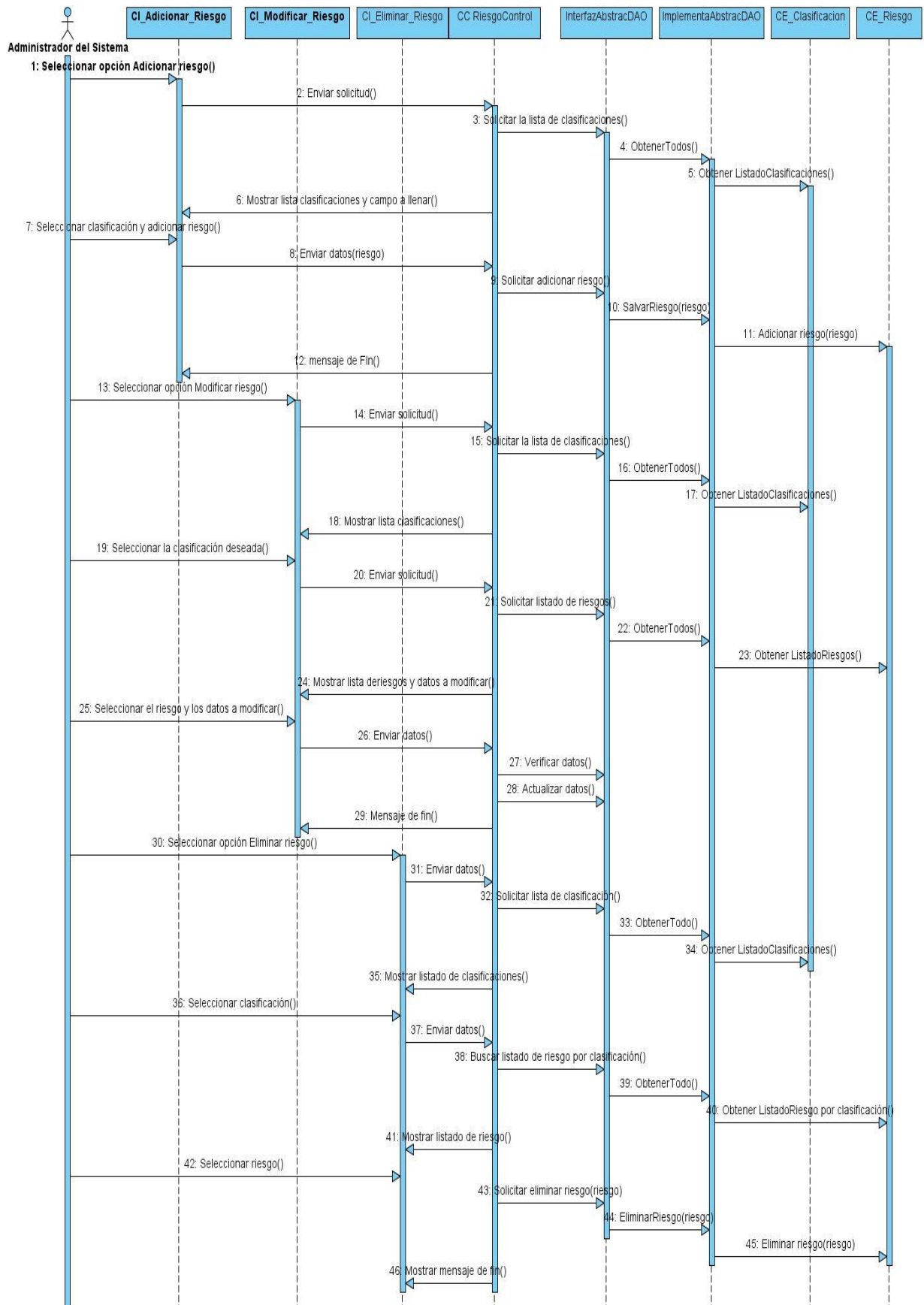


Diagrama de Secuencia CU Mostrar documento de Gestión de Riesgos

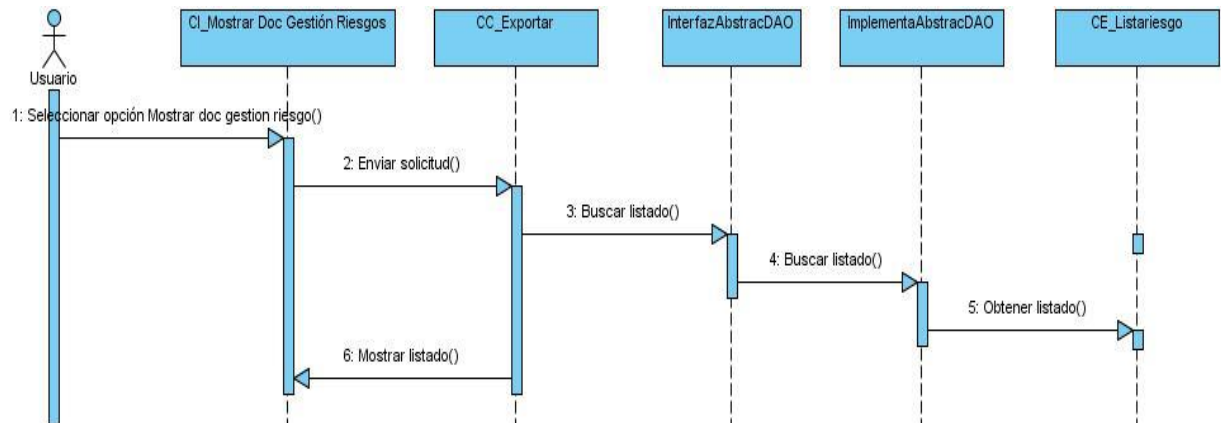
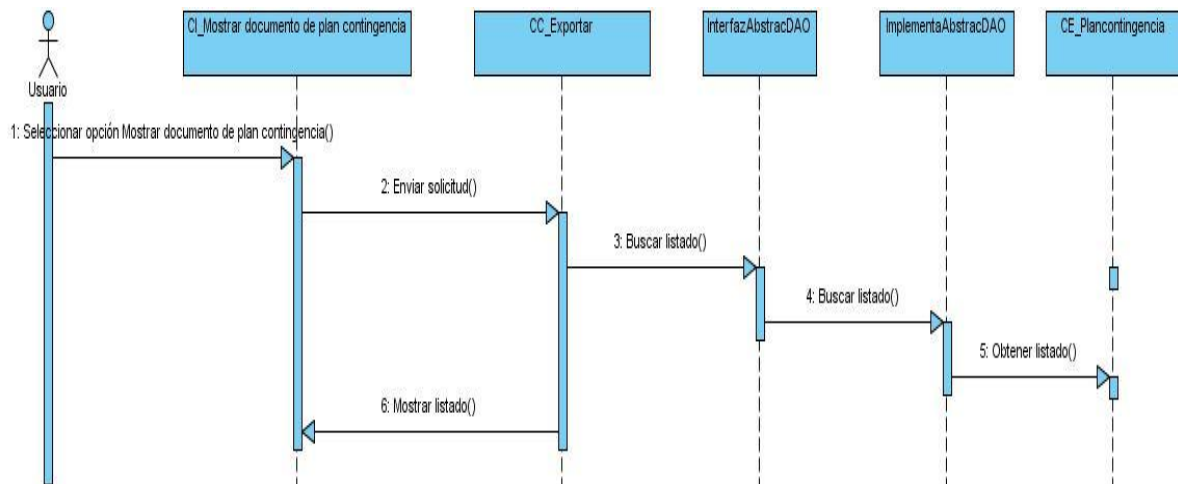


Diagrama de Secuencia CU Mostrar documento de plan de contingencia



### GLOSARIO DE TÉRMINOS

**RUP:** Proceso Unificado (Rational Unified Process) metodología para el desarrollo de sistemas informáticos, dirigidos por casos de uso.

**CASE:** Acrónimo inglés de Computer Aided Software Engineering, que significa Ingeniería de Software Asistida por Ordenador.

**Java:** Es un lenguaje de programación, de alto nivel, orientado a objetos y desarrollado por Sun Microsystems.

**Plugins:** Función o utilidad generalmente muy específica. Se adiciona a algún programa para ser ejecutado. Los plugins típicos tienen la función de reproducir determinados formatos de gráficos, reproducir datos multimedia, etc.

**UML:** Lenguaje de Modelado Unificado (Unified Model Language), lenguaje gráfico que brinda un vocabulario y reglas para especificar, construir, visualizar y documentar los artefactos de un sistema utilizando el enfoque orientado a objetos.

**XP:** Metodología para el desarrollo de sistemas informáticos que se basa en el intercambio constante con el cliente.

**Metodología:** Define quién hace qué, cómo y cuando.