

Universidad de las Ciencias Informáticas

Facultad 5

Laboratorio de Gestión de Proyectos (GESPRO)



Módulo de Gestión de Riesgos versión 2.0 para el sistema GESPRO 12.05

Trabajo para optar por el Título de Ingeniero en
Ciencias Informáticas.

Autor(es): Iván Martínez Vigil

Tutor(es): MSc. Iliana Pérez Pupo

La Habana, Junio 2012

Datos del Tutor:

Nombre y apellidos: Iliana Pérez Pupo

Especialidad de graduación: Ingeniero en Ciencias Informáticas.

Categoría docente: Profesor Instructor

Categoría científica: Máster en Ciencias en el Área de Gestión de Proyectos

Años de experiencia en el tema: 4 años

Años de graduado: 4 años

Correo electrónico: jperez@uci.cu

Agradecimientos:

A toda mi familia y amistades, por haber puesto su granito de arena que ha permitido ser la persona que soy hoy, por transmitirme buenos consejos, apoyo incondicional y dedicación.

A mi mamá por su dedicación, por sus consejos, por confiar en mí y darme ánimo en todo momento.

A mi abuela y a mi tía, por acogerme como un hijo, y por poner todo su empeño en ayudar a construir mis sueños.

A mi novia quien me ha brindado su apoyo durante más de 2 años y quien se ha comportado como mi compañera, mi hermana y mi madre, por comprenderme en momentos difíciles y exigirme éxito en mis estudios.

A mi tutora por toda la ayuda brindada.

Al Dr. Pedro Piñeiro, a Ernesto, Javier, Oniel, y a todos aquellos que de alguna forma contribuyeron a la realización de este trabajo.

Dedicatoria:

A mi mamá, a mi abuela y a mi tía, por desear tanto como yo este momento.

A Massiel, por tenerme en su lista de personas preferidas.

A mi novia, por quererme tanto, y por estar siempre ahí cuando la he necesitado.

A mi hermano, que le sirva de ejemplo y vea que todo lo que uno se propone lo logra.

Declaración de Autoría:

Declaro por este medio que yo Iván Martínez Vigil, con carné de identidad 88011910400, soy el único autor del trabajo final de tesis: "Módulo para la Gestión de Riesgos versión 2.0 del sistema GESPRO 12.05", y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Y para que así conste, firmo la presente declaración jurada de autoría en La Habana a los ____ días del mes de _____ del año _____.

Firma del autor(a)

Iván Martínez Vigil

Firma del tutor(a)

Iliana Pérez Pupo

Resumen

En este trabajo se propone el módulo de Gestión de Riesgos versión 2.0 para el sistema GESPRO. Para ello se realizó un estudio de las disposiciones establecidas por la Resolución 60/11 de la Contraloría General de la Republica, del estado del arte de diferentes modelos para la Gestión de Riesgos, y de los procesos de Gestión de Riesgos que realizan diferentes herramientas de Gestión de Proyectos, con el objetivo de mejorar las deficiencias que existían en la versión anterior de dicho módulo, pues el mismo no establece ninguna relación entre los riesgos genéricos o del centro y los riesgos del proyecto, las incidencias o desviaciones relacionadas con los riesgos hay que entrarlas de manera manual, muchas veces causando rechazo por el usuario, no visualiza gráficas que muestren información detallada de los riesgos, así como de las incidencias y desviaciones, las acciones de contingencia y mitigación hay que entrarlas manualmente, no permite exportar el listados, de incidencias o desviaciones a formato '.csv', no muestra alertas de ningún tipo cuando ciertos indicadores alcanzan valores pico, entre otras deficiencias. El módulo que se propone mejora todas estas deficiencias identificadas y se ajusta a la Resolución anteriormente mencionada. Para validar esto se realizó un análisis comparativo del sistema GESPRO en cuanto al proceso de Gestión de Riesgos con los sistemas estudiados. Se describen en este trabajo las nuevas funcionalidades que brinda esta nueva versión del módulo.

Palabras claves: Módulo, Gestión de Riesgos, Gestión de Proyectos, GESPRO

Abstract

This work proposes the Risk Management Module version 2.0 for GESPRO system. To do so a study was conducted of the provisions established by Resolution 60/11 of the Comptroller General of the Republic, state of the art of different models for Risk Management and Risk Management processes which perform different tools for project Management, in order to improve the deficiencies existing in the previous version of said module, since it does not establish any relationship between the center or generic risks and project risks, incidents or deviations relating to risks should be entered manually, often causing rejection by the user, does not display graphs showing details of the risks as well as incidents and deviations, contingency and mitigation activities must also be entered manually, it doesn't allows to export the lists of incidents or deviations to ".csv", does not show any alerts when certain indicators reach peak values , among other deficiencies. The module aims to improve these deficiencies identified and conforms to the aforementioned resolution. To validate this, a comparative analysis was performed of the system GESPRO in terms of Risk Management process with the studied systems. Are described in this work the new features offered by this new version of the module.

Keywords: Module, Risk Management, Project Management, GESPRO

Índice de Contenido

<i>Introducción</i>	1
<i>Capítulo 1: Fundamentación Teórica</i>	7
<i>Introducción</i>	7
1.1 <i>Gestión de Proyectos</i>	7
1.2 <i>Gestión de Riesgos</i>	9
1.3 <i>Principales modelos de Gestión de Riesgos</i>	11
1.3.1 <i>Modelo de Boehm</i>	11
1.3.2 <i>Administración Continuada de Riesgos (CRM, por sus siglas en inglés)</i>	12
1.3.3 <i>Evaluación de Riesgos del Software (SRE, por sus siglas en inglés)</i>	13
1.3.4 <i>Framework de Soluciones de Microsoft (MSF, por sus siglas en inglés)</i>	14
1.3.5 <i>Modelo de Gestión Riesgos en proyectos de Desarrollo de Software (MoGeRi)</i>	15
1.3.6 <i>PMBOK</i>	15
1.4 <i>Gestión de Riesgos según diferentes herramientas para la Gestión de Proyectos</i>	16
1.4.1 <i>Team Foundation Server (TFS)</i>	17
1.4.2 <i>Trac</i>	17
1.4.3 <i>DotProject</i>	18
1.4.4 <i>Jira</i>	18
1.4.5 <i>GESPRO 11.05</i>	20
1.5 <i>Metodologías de desarrollo de software</i>	20
1.6 <i>Herramientas de desarrollo de software</i>	24
1.6.1 <i>Herramientas CASE</i>	24
1.6.2 <i>Framework</i>	25

Módulo de Gestión de Riesgos versión 2.0

para el sistema GESPRO 12.05

1.6.3 Gestor de Base de Datos.....	26
1.6.4 Entornos de Desarrollo Integrado	27
1.7 Conclusiones Parciales	27
Capítulo 2: Fundamentos de la solución propuesta	28
2.1 Introducción.....	28
2.2 Descripción de la solución propuesta.	28
2.3 Diseño e implementación del módulo.	34
2.3.1 Arquitectura del módulo	34
2.3.2 Diagramas de Clases.....	41
2.3.3 Patrones de Diseño	44
2.3.4 Diseño de política de seguridad informática.....	45
2.3.5 Modelo Entidad Relación	46
2.3.6 Diagrama de Componentes	52
2.3.7 Diagrama de Despliegue.....	52
2.4 Conclusiones Parciales	53
Capítulo 3: Análisis de los resultados	54
3.1 Introducción.....	54
3.2 Diseño de Casos de Pruebas.....	54
3.3 Análisis comparativo de la herramienta GESPRO 12.05 respecto a otras herramientas de Gestión de Proyectos.....	54
3.4 Análisis del alineamiento de la solución a la Resolución 60/11.....	57
3.5 Validación de las nuevas funcionalidades a través de las pruebas de caja negra	59
3.5 Conclusiones Parciales	65
Conclusiones	66
Recomendaciones	67

Módulo de Gestión de Riesgos versión 2.0
para el sistema GESPRO 12.05

Referencias Bibliográficas 68

Introducción

La Gestión de Proyectos (GP) fue creada para dirigir proyectos industriales y de desarrollo técnico de gran complejidad. En sus comienzos, era un campo bastante técnico, más conocido quizás, por generar muchos trámites burocráticos. La GP consiste simplemente en conducir un proyecto desde el comienzo hasta un final satisfactorio, haciendo uso conjunto de procesos y sistemas que orienten y motiven al personal a realizar satisfactoriamente su trabajo dentro del proyecto (Piñero, 2010).

Existen varias herramientas de software para planificar y gestionar proyectos. Existen varios tipos de herramientas de administración de proyectos, entre ellas las utilizadas para la planificación de proyectos, manejo y control de presupuesto, asignación de recursos, software para colaboración, software para comunicación, manejo de la calidad y documentación o administración de sistemas, las cuales son usadas para manejar la complejidad que conlleva un proyecto grande (Caballero, 2006).

Una de las áreas fundamentales de la GP es la Gestión de Riesgos (GR), la cual incluye los procesos relacionados con su planificación, la identificación y el análisis de riesgos, las respuestas a los mismos, y su seguimiento y control en un proyecto. Un riesgo, en un sentido general, es un elemento que atenta en alguna medida contra el desarrollo satisfactorio de cualquier actividad (PMI, 2004).

Existen directivas que muestran el énfasis en la GR. Por ejemplo, el estándar EIA/IEEE en su sección 5.19.1 define: “El desarrollador debe ejecutar la Gestión de Riesgos a través de todo el proceso de desarrollo. Los desarrolladores deberán identificar, analizar y priorizar las áreas del proyecto de desarrollo de software en las cuales existen riesgos potenciales (riesgos técnicos, costos o calendario); desarrollando las estrategias para gestionarlos, recopilándolos en un plan de desarrollo e implementando dichas estrategias de acuerdo al plan” (Matsuo, 1999).

La Resolución No. 60/11 establecida por la Contraloría General de la República de Cuba, publicada en la Gaceta Oficial del Ministerio de Justicia el día 3 de Marzo de 2011, establece en varios de sus artículos la GR como un procedimiento necesario y obligatorio para los órganos, organismos, organizaciones y entidades nacionales. En su Sección Segunda “Gestión y Prevención de Riesgos”, establece y define las bases para la identificación y análisis de los riesgos que enfrentan los organismos para alcanzar sus objetivos (MINJUS, 2011).

La Universidad de las Ciencias Informáticas (UCI), centro de estudio basado en el concepto de universidad productiva, tiene como objetivos lograr recursos humanos formados e ingreso de divisas por exportación. En este sentido se produce en la universidad una modificación de su plan de estudio y del modelo de producción simultáneamente, potenciando la sinergia entre las áreas de producción, formación,

Módulo de Gestión de Riesgos versión 2.0

para el sistema GESPRO 12.05

investigación y postgrado (Piñero, 2010).

Teniendo en cuenta el tamaño de la organización y el volumen de datos que se manejan, se identifica como una necesidad el uso de herramientas informáticas que ayuden a la implantación del modelo de desarrollo tecnológico (Piñero, 2010).

Antes del año 2009 existían en nuestra Universidad alrededor de 6 herramientas de administración de proyectos. Con el uso de éstas no se cubrían las necesidades de la universidad en su totalidad, debido principalmente a que no se podía gestionar de una manera integrada toda la información, producto de esta diversidad.

En función de todo lo anterior planteado, en el año 2010 la UCI decide desarrollar el paquete para la Gestión Integrada de Proyectos (GESPRO), formado a partir de la integración de 18 herramientas libres comercializadas bajo licencia GPL, y que incluye módulos que permiten entre otras funcionalidades: la gestión de portafolios de proyectos, la gestión de alcance, la gestión de tiempo, la gestión de riesgos, la gestión de comunicaciones, la gestión de la calidad, la gestión de recursos humanos, monitoreo y control de la plataforma, el control de versiones y la gestión documental (Piñero, 2010).

Existe actualmente dentro de la herramienta GESPRO 11.05 un módulo para la GR en su versión 1.0, el cual cuenta con varias funcionalidades que se corresponden con los objetivos para los cuales fue creado. Dicho módulo es capaz de gestionar riesgos genéricos o del centro, categorías de riesgos, riesgos específicos para cada proyecto, incidencias¹, desviaciones², así como mostrar diversas estadísticas, que proporcionan mayor información al usuario.

La administración de los riesgos que realiza este módulo es insuficiente, pues no establece ninguna relación entre los riesgos genéricos o del centro y los riesgos del proyecto, las incidencias o desviaciones relacionadas con los riesgos hay que entrarlas de manera manual, muchas veces causando rechazo por el usuario, no visualiza gráficas que muestren información detallada de los riesgos, así como de las incidencias y desviaciones, las acciones de contingencia y mitigación hay que entrarlas manualmente, no permite exportar el listados, de incidencias o desviaciones a formato ‘.csv’, no muestra alertas de ningún tipo cuando ciertos indicadores alcanzan valores pico, entre otras deficiencias. De aquí parte entonces la necesidad de una nueva versión del módulo de Gestión de Riesgos para el sistema GESPRO que automatice todos estos procesos, además de alinearse con la Resolución No. 60/11 para el control interno establecida en nuestro país.

¹ Según (WordReference, 2012): Lo que sucede en el curso de un asunto o negocio y tiene relación con ello.

² Según (WordReference, 2012): Cambio de trayectoria o de intenciones.

Módulo de Gestión de Riesgos versión 2.0

para el sistema GESPRO 12.05

Por lo que el **problema científico** queda planteado de la siguiente forma: Las funcionalidades del módulo de Gestión de Riesgos en el sistema GESPRO no permiten garantizar una correcta ejecución de los procesos de planificación, análisis cualitativo y respuesta a los riesgos en proyectos de software.

El **objeto de estudio** se enmarca dentro de la Gestión de Riesgos en la Gestión de Proyectos.

El **campo de acción** se enmarca dentro de la implementación de la Gestión de Riesgos en el sistema GESPRO.

El **objetivo general** que se persigue es diseñar e implementar el módulo de Gestión de Riesgos para el sistema GESPRO 12.05 con nuevas funcionalidades que permitan mejorar procesos de planificación, análisis cualitativo y respuesta a los riesgos alineándose además con las disposiciones establecidas para el control interno en la Resolución 60/11.

Este objetivo general, puede desglosarse en varios **objetivos específicos**, como son:

- Elaborar el marco teórico de la investigación asociado a la Gestión de Riesgos.
- Definir nuevas funcionalidades al módulo de Gestión de Riesgos del sistema GESPRO 12.05.
- Implementar las funcionalidades definidas.
- Validar las nuevas funcionalidades a partir de las pruebas de caja negra y el despliegue del módulo en la plataforma GESPRO 12.05.

Por tanto queda planteada la siguiente **idea a defender**: Con el desarrollo de la versión 2.0 del módulo de Gestión de Riesgos para el sistema GESPRO 12.05 se garantizará una administración más completa y mejorada de los riesgos en las etapas de identificación, análisis cualitativo, respuesta y planificación, y monitoreo y control de los riesgos, que en su versión anterior 1.0, alineándolo además con la Resolución No. 60/11 para el Control Interno establecida por la Contraloría General de nuestro país.

Para dar cumplimiento a los objetivos de investigación, se proponen las siguientes **tareas investigativas**:

- Elaboración del marco teórico a partir del estado del arte actual referente al tema.
- Aprendizaje del desarrollo de módulos en el framework Ruby on Rails.
- Adición de nuevas funcionalidades a la versión 1.0 del módulo de GR del sistema GESPRO 11.05.

Módulo de Gestión de Riesgos versión 2.0

para el sistema GESPRO 12.05

- Validación de las nuevas funcionalidades a través de las pruebas de caja negra.
- Integración de la nueva versión del módulo en el sistema GESPRO 12.05.
- Elaboración del diseño de los casos de pruebas.
- Elaboración de un Manual de Usuario.

Al realizar las tareas, se debe arrojar como **posible resultado**: La implantación del módulo de Gestión de Riesgos versión 2.0 extendido con nuevas funcionalidades de administración de riesgos a diferentes niveles y cálculo del impacto en la gestión de los objetivos de la empresa, además de un manual de usuario.

Dentro de los **métodos y técnicas utilizados** en la presente investigación, como parte del diseño metodológico fue seguida una estrategia explicativa, pues los conocimientos precedentes acerca del problema han sido suficientes para plantear una idea a defender y la representación del problema es clara en lo referente a la caracterización del fenómeno en sus aspectos externos.

Se utilizaron además diferentes métodos de investigación:

- **Métodos Teóricos:**
 - **Histórico – Lógico:** Mediante este método se analizó la evolución de los diferentes problemas que traen consigo los riesgos en un proyecto de software, con el objetivo de realizar una gestión óptima de los mismos.
 - **Analítico – Sintético:** Con el uso de este método de investigación se analizaron las informaciones obtenidas, mediante su desglose, para después realizar una síntesis de las mismas y arribar a las principales ideas.
 - **Modelación:** Este método se utilizó para crear un prototipo funcional del sistema de interacción natural.
 - **Hipotético – Deductivo:** A partir del problema se plantearon objetivos específicos y una idea a defender que con el transcurso de la investigación es resuelta.
- **Métodos Empíricos:**
 - **Consulta de las fuentes de información:** Para seleccionar la información necesaria para construir el marco teórico.
 - **Observación:** Para reunir información visual sobre lo que el objeto de

Módulo de Gestión de Riesgos versión 2.0

para el sistema GESPRO 12.05

estudio hace y cómo se comporta.

- **Pruebas:** Para valorar el desempeño del sistema elaborado.

Quedan entonces definidos como **marco conceptual**, los siguientes elementos:

Riesgo: Es la vulnerabilidad de "bienes jurídicos protegidos" ante un posible o potencial perjuicio o daño (WordReference, 2012).

Riesgo Informático: La ISO (Organización Internacional de Estándares) define el riesgo informático como: "La posibilidad que una amenaza se materialice, utilizando vulnerabilidad existente en un activo o grupos de activos, generándose así pérdidas o daños (WordReference, 2012).

Gestión de Proyectos (GP): También conocida como Gerencia, Dirección o Administración de proyectos es la disciplina de planear, organizar, asegurar y coordinar recursos y personas para cumplir con los Objetivos, Entregables y Criterios de Éxito de los proyectos. Un proyecto es un conjunto de actividades relacionadas para lograr un fin específico, con un comienzo y fin claros, sujeto a tres "restricciones" principales: Tiempo, Costo y Alcance (WordReference, 2012).

Gestión de Riesgos (GR): Es un enfoque estructurado para manejar la incertidumbre relativa a una amenaza, a través de una secuencia de actividades humanas que incluyen evaluación de riesgo, estrategias de desarrollo para manejarlo y mitigación del riesgo utilizando recursos gerenciales. Las estrategias incluyen transferir el riesgo a otra parte, evadir el riesgo, reducir los efectos negativos del riesgo y aceptar algunas o todas las consecuencias de un riesgo particular (WordReference, 2012).

Módulo: En programación un módulo es una porción de un programa de computadora. De las varias tareas que debe realizar un programa para cumplir con su función u objetivos, un módulo realizará, comúnmente, una de dichas tareas (o varias, en algún caso) (WordReference, 2012).

Y se plantea además la siguiente **estructura para el contenido de la tesis:**

Capítulo I: Estudio del estado del Arte sobre la Gestión de Proyectos, y la Gestión de Riesgos. Se establece el fundamento teórico de la investigación a partir de hacer un análisis crítico de las disposiciones establecidas en la Resolución 60/11, y de los principales Modelos de Gestión de Riesgos, lo que permite sustentar teóricamente la investigación. Caracterización de las metodologías y herramientas de desarrollo de software que se utilizarán.

Capítulo II: Descripción de la solución propuesta tomando como punto de partida la arquitectura, y continuando con el diseño y descripción de las clases, el diseño y descripción de la base de datos, los patrones de diseño utilizados y los componentes

Módulo de Gestión de Riesgos versión 2.0

para el sistema GESPRO 12.05

del sistema.

Capítulo III: Análisis comparativo de la herramienta GESPRO con otras herramientas de Gestión de Proyectos en cuanto a Gestión de Riesgos tomando como criterios de comparación las nuevas funcionalidades que brindará el sistema con el objetivo de validar su mejora. Análisis del alineamiento con la Resolución 60/11. Resultados de las pruebas de caja negra aplicadas al sistema.

Capítulo 1: Fundamentación Teórica.

Introducción

En este capítulo se introduce el fundamento teórico de la investigación a partir del análisis crítico de la bibliografía existente. Se formalizan los conceptos de la GP y la GR. Se realiza un estudio de las disposiciones establecidas por la Resolución 60/11 de la Contraloría General de la República de Cuba. Se evalúan, además, diferentes marcos para la GR, así como herramientas, técnicas, experiencias alcanzadas y estudios sobre la GR en Cuba y en nuestra Universidad. Por último, se hace un estudio de las diferentes metodologías y herramientas de desarrollo de software existentes.

1.1 Gestión de Proyectos

La GP es una rama de la ingeniería que emplea metodologías bien definidas, realiza medidas repetibles y confiables, estima costos y tiempos, da elementos para la administración y control de los proyectos y replantea resultados para ajustar la información disponible. Es la aplicación de conocimientos, habilidades, herramientas y técnicas para proyectar actividades destinadas a satisfacer las necesidades y expectativas de todos los involucrados en un proyecto (PMI, 2004).

Esto supone buscar un equilibrio entre demandas contrapuestas en (Piñero, 2010):

- Alcance, tiempo, costo y calidad.
- Beneficiarios con diferentes necesidades y expectativas.
- Requerimientos identificados (necesidades) y requerimientos no identificados (expectativas).

Los proyectos necesitan ser ejecutados y entregados bajo ciertas restricciones. Tradicionalmente, estas restricciones han sido alcance, tiempo y costo. Esto también se conoce como el Triángulo de la GP, donde cada lado representa una restricción. Un lado del triángulo no puede ser modificado sin impactar a los otros. Un refinamiento posterior de las restricciones separa la calidad del producto del alcance, y hace de la calidad una cuarta restricción, aunque estas no son las únicas restricciones que pueden afectar el proceso de gestión (Piñero, 2010).

Capítulo 1: Fundamentación Teórica

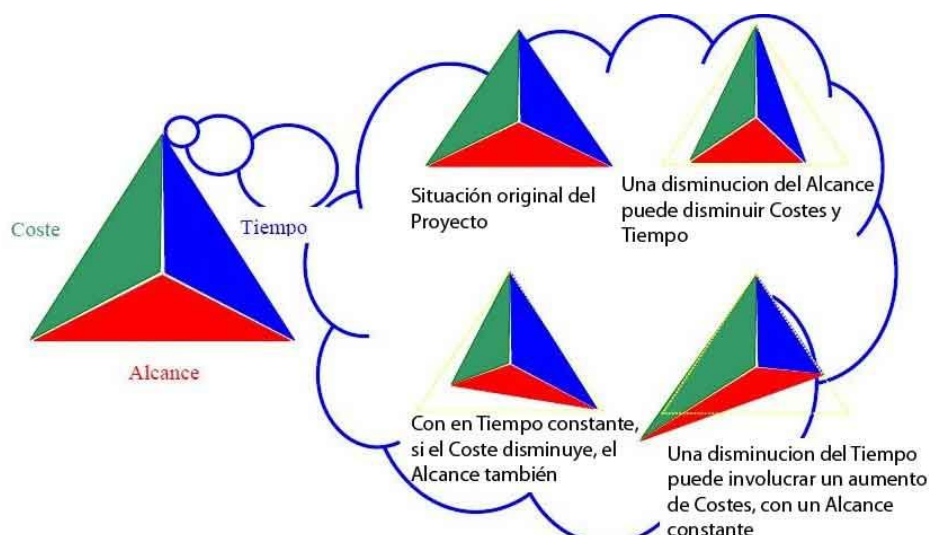


Figura 1. Interrelación entre coste, alcance y tiempo, tomado de (Piñero, 2010).

Una de las instituciones más reconocidas en el área de la GP es el Instituto de Administración de Proyectos (PMI, por sus siglas en inglés), organización internacional orientada a la difusión y determinación de las mejores prácticas de gestión de proyectos. Con ese propósito, produce documentos que describen las prácticas generalmente aceptadas de la administración de proyectos (PMI, 2004).

El más importante de los documentos publicados en la actualidad por el PMI es el “PMBOK, Una Guía de Cuerpo y Conocimiento para la Administración de Proyectos” (“PMBOK, A Guide to the Project Management Body of Knowledge”). El propósito de esta guía es describir el conocimiento y las prácticas aplicables a la mayoría de los proyectos, llegando a un consenso sobre su valor y utilidad. Las áreas de conocimientos definidas por la guía del PMBOK son las que se describen en la tabla siguiente (PMI, 2004):

Código	Área de conocimiento	Descripción
IM	Gestión de la Integración	Incluye los procesos requeridos para asegurar que los diversos elementos del proyecto están debidamente coordinados.
SM	Gestión del Alcance	Incluye los procesos requeridos para asegurar que el proyecto incluya todo el trabajo requerido y solo el trabajo requerido para completar el proyecto satisfactoriamente.
TM	Gestión del Tiempo	Incluye los procesos requeridos para asegurar la terminación a tiempo del proyecto.

Capítulo 1: Fundamentación Teórica

CoM	Gestión del Costo	Incluye los procesos requeridos para asegurar que el proyecto se complete dentro del presupuesto aprobado.
QM	Gestión de la Calidad	Incluye los procesos requeridos para asegurar que el proyecto satisfaga las necesidades por las cuales se llevó a cabo.
HRM	Gestión de recursos Humanos	Incluye los procesos requeridos para hacer el uso más eficaz de las personas involucradas con el proyecto.
CmM	Gestión de las Comunicaciones	Incluye los procesos requeridos para asegurar la oportuna y apropiada generación, recopilación, difusión, almacenamiento y disposición final de la información del proyecto.
RM	Gestión de Riesgos	Es el proceso sistemático de identificar, analizar y responder a los riesgos del proyecto.
PM	Gestión del Abastecimiento	Incluye los procesos requeridos para adquirir los bienes y servicios, para lograr el alcance del proyecto, desde fuera de la organización ejecutante.

Tabla 1. Áreas de conocimiento de la GP según (PMI, 2004).

1.2 Gestión de Riesgos

La definición de GR se puede aceptar de acuerdo a las posiciones de quienes, en el mundo, la estudian y desarrollan. Se trata de:

- Identificar, tratar y eliminar riesgos de software antes de que se tornen en amenazas que afecten objetivos esenciales para el proyecto (Rosenberg, et al., 2000).
- El proceso formal en el que los factores de riesgos son sistemáticamente identificados, evaluados y mitigados (Williams, et al., 1999).
- La identificación y tratamiento de riesgos en etapas tempranas de desarrollo de los proyectos reducen costos a largo plazo y ayudan a prevenir desastres (Boehm, 1991).
- Incluye los procesos relacionados con la planificación de la GR, la identificación y el análisis de riesgos, las respuestas a los riesgos, y el seguimiento y control de riesgos de un proyecto. Los objetivos son aumentar la probabilidad y el impacto de los eventos positivos, y disminuir la probabilidad y el impacto de los eventos adversos para el proyecto (PMI, 2004).
- Según Roger Pressman citando a Robert Charette (Pressman, 2005): El análisis y la GR son una serie de pasos que ayudan al equipo del software a

Capítulo 1: Fundamentación Teórica

comprender a gestionar la incertidumbre.

- La práctica compuesta de procesos, métodos y herramientas y que provee de un entorno disciplinado para la toma de decisiones proactiva en base a determinar constantemente que puede ir mal (riesgos), se emplea para identificar cuáles son los riesgos más importantes en los cuales enfocarse e implementar estrategias para gestionarlos (Murphy, et al., 2008).

A partir de las definiciones estudiadas, es posible concluir que la GR es el proceso de identificar, analizar, responder y medir o controlar la ocurrencia de eventos inciertos esperados o inesperados, que permite reducir los efectos de los mismos en los aspectos económico y social, y al mismo tiempo genera un ambiente de disciplina y adecuado manejo de los recursos de un proyecto.

Según (Boehm, 1991), los diez factores de riesgo que más se presentan en los proyectos de software son: insuficiencias del personal, presupuestos y planificaciones poco realistas, desarrollo de funciones y propiedades erróneas, desarrollo de una mala interfaz de usuario, incorporación de funciones o características innecesarias, cambios continuos en los requerimientos, insuficiencias en los componentes suministrados externamente, insuficiencias en las tareas ejecutadas externamente, insuficiencias en la ejecución en tiempo real y tensionadas capacidades de la ciencia de la computación.

1.2.1 Disposiciones implantadas por la Resolución 60/11 establecida por la Contraloría General de la República

La resolución 60/11 en su *Capítulo 1: Disposiciones Generales*, Artículo 3 establece:

“El control interno es el proceso integrado a las operaciones con un enfoque de mejoramiento continuo, extendido a todas las actividades inherentes a la gestión, efectuado por la dirección y el resto del personal; se implementa mediante un sistema integrado de normas y procedimientos, que contribuyen a prever y limitar los riesgos internos y externos, proporciona una seguridad razonable al logro de los objetivos institucionales y una adecuada rendición de cuentas” (MINJUS, 2011).

En el *Capítulo 2: De los Componentes y Normas de Carácter General, Sección Segunda: Gestión y Prevención de Riesgos*, se constituye un componente para la identificación y análisis de los riesgos que enfrentan los diferentes organismos. Implanta una clasificación de los mismos en internos y externos, y dicta una guía para conformar un Plan de Prevención de Riesgos, que define el modo en el que habrán de gestionarse.

El componente se estructura de la siguiente forma (MINJUS, 2011):

- **Identificación de riesgos y detección del cambio:**

En la identificación de los riesgos, se tipifican todos los que pueden afectar el cumplimiento de los objetivos. La identificación de riesgos se nutre de la

Capítulo 1: Fundamentación Teórica

experiencia derivada de hechos ocurridos, así como de los que puedan preverse en el futuro y se determinan para cada proceso, actividad y operación a desarrollar.

Los factores externos incluyen los económico-financieros, medioambientales, políticos, sociales y tecnológicos y los internos incluyen la estructura organizativa, composición de los recursos humanos, procesos productivos o de servicios y de tecnología, entre otros (MINJUS, 2011).

- **Determinación de los objetivos de control:**

Los objetivos de control son el resultado o propósito que se desea alcanzar con la aplicación de procedimientos de control, los que deben verificar los riesgos identificados y estar en función de la política y estrategia de la organización. Luego de identificar, evaluar y cuantificar, siempre que sea posible, los riesgos por procesos, actividades y operaciones, la máxima dirección y demás directivos de las áreas, con la participación de los trabajadores, realizan un diagnóstico y determinan los objetivos de control, dejando evidencia documental del proceso (MINJUS, 2011).

- **Prevención de riesgos:**

Esta norma constituye un conjunto de acciones o procedimientos de carácter ético-moral, técnico- organizativos y de control, dirigidas de modo consciente a eliminar o reducir al mínimo posible las causas y condiciones que propician los riesgos internos y externos, así como los hechos de indisciplinas e ilegalidades, que continuados y en un clima de impunidad, provocan manifestaciones de corrupción administrativa o la ocurrencia de presuntos hechos delictivos.

En función de los objetivos de control determinados de acuerdo con los riesgos identificados por los trabajadores de cada área o actividad y las medidas o acciones de control necesarias, se elabora el Plan de Prevención de Riesgos, cuyos aspectos más relevantes tributan al del órgano, organismo, organización o entidad, el que de forma general incluye los riesgos que ponen en peligro el cumplimiento de los objetivos y la misión. Los planes elaborados son evaluados por el Comité de Prevención y Control y aprobados por el órgano colegiado de dirección (MINJUS, 2011).

1.3 Principales modelos de Gestión de Riesgos

En la GR existen varios modelos que sintetizan cada uno de los pasos que se deben tener presentes a la hora de llevar a cabo la gestión, para realizar un proceso más organizado y eficaz. A continuación se presentan varios de ellos:

1.3.1 Modelo de Boehm

El modelo de GR de Boehm propone procesos y técnicas divididas en dos partes principales: medición de los riesgos y control de los riesgos. Dentro de la medición incluye procesos como Identificación, Análisis y Priorización. Estos procesos están

Capítulo 1: Fundamentación Teórica

bien ubicados en esta categoría de acuerdo a su intención (Boehm, 1989).



Figura 2. Modelo propuesto por Boehm para la GR, tomado de (Boehm, 1989).

Para el proceso de análisis propone modelos de rendimiento, modelos de costo, análisis de redes, análisis del factor de calidad. Este proceso se considera importante y con el objetivo de extraer o descubrir la mayor cantidad de datos posibles de los riesgos y el uso de estas técnicas contribuye en este sentido, sin embargo, la descripción precisa y objetiva del uso de estas técnicas se describen abstractamente y carecen de claridad para ser usadas (Rivera, 2010).

1.3.2 Administración Continuada de Riesgos (CRM, por sus siglas en inglés)

El Centro de Tecnología de Aseguramiento del Software (The Software Assurance Technology Center) de la NASA desarrolló en conjunto con el SEI el CRM a principios de los 90, un método de GR extendido a la comunidad de la NASA. El método reconoce la importancia de la GR como una práctica necesaria y define seis principios o funciones para ello, como se muestra en la siguiente figura (Rivera, 2010):

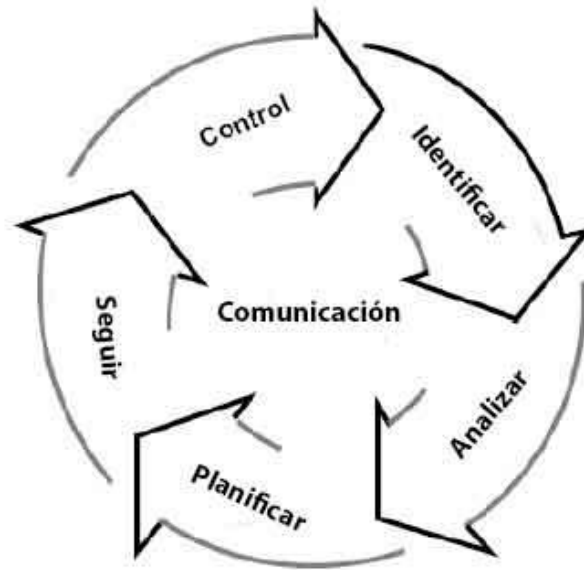


Figura 3. Modelo propuesto por el CRM para la GR, tomado de (Rosenberg, et al., 2000).

En este método estos procesos ocurren continuamente e iterativamente, pueden ocurrir simultáneamente, es decir, los riesgos son a la vez controlados mientras se están identificando nuevos riesgos o se está ejecutando el plan de mitigación para otro riesgo. No obstante no queda claro en este modelo cómo ocurre dicha simultaneidad y cómo es la comunicación durante la simultaneidad o durante su flujo secuencial (García, 2008).

1.3.3 Evaluación de Riesgos del Software (SRE, por sus siglas en inglés)

Es un método para identificar, analizar, evaluar y desarrollar estrategias de mitigación para los riesgos. Está concebido como un servicio a prestar a otras empresas, es implementado en cinco fases:



Figura 4. Etapas de la GR según (Williams, et al., 1999).

El método describe cómo tiene que estar estructurado el equipo para aplicarlo. Describe cada fase definiendo su objetivo, los elementos que maneja (entradas y salidas), las personas involucradas, las herramientas a utilizar, las tareas a realizar y el tiempo promedio necesario para realizarlas (Williams et al., 1999).

Prioriza los riesgos de acuerdo a su exposición y al criterio de los entrevistados, en este sentido el método no define técnicas “fuertes” para la priorización (Higuera, et al., 1996). Como resultado de esta priorización se obtiene un documento con la lista de riesgos que es entregado al gerente de proyecto.

1.3.4 Framework de Soluciones de Microsoft (MSF, por sus siglas en inglés)

MSF es un marco de trabajo propuesto por Microsoft que cubre el ciclo de vida del proyecto y propone métodos y herramientas para el desarrollo del mismo. Dentro de sus disciplinas proponen la “disciplina de administración de riesgos”³, como un mecanismo para enfrentar la incertidumbre en los proyectos, y proponen un enfoque proactivo donde continuamente se evalúen los riesgos (Rivera, 2010).

Se basa en los principios fundamentales de MSF: el principio de “mantenerse ágil, esperar el cambio”, el principio de las comunicaciones abiertas, el principio de “aprender de todas las experiencias” y el principio de la “responsabilidad compartida” (García, 2008).

Define seis procesos para la GR.

³ Del inglés: “Risk management discipline”.

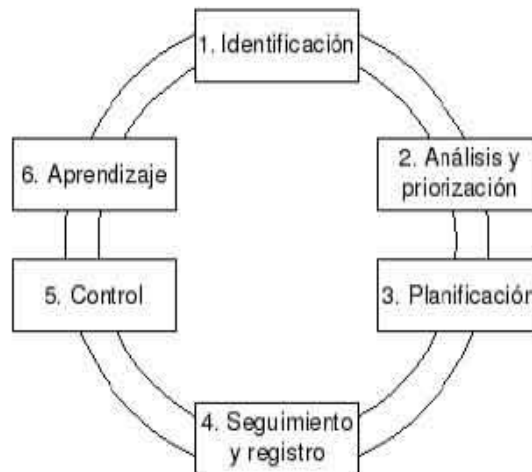


Figura 5. Procesos definidos por Microsoft para la GR, tomado de (Rivera, 2010).

MSF incluye un elemento interesante: la exposición del riesgo, que es una medida de su amenaza, y se define una función para calcularla dependiendo del grado de aceptación de la entidad. MSF propone una forma de calcularla y destaca la utilización de una matriz (probabilidad x impacto) (García, 2008).

1.3.5 Modelo de Gestión Riesgos en proyectos de Desarrollo de Software (MoGeRi)

El Modelo de GR para proyectos de Software en la UCI (MoGeRi) consta de seis procesos (Zulueta, 2007): Planificación de la gestión de los riesgos, Identificación de los riesgos, Análisis de los riesgos, Planificación de las respuestas a los riesgos, Seguimiento y Control de los riesgos, y Comunicación de la información de los riesgos.

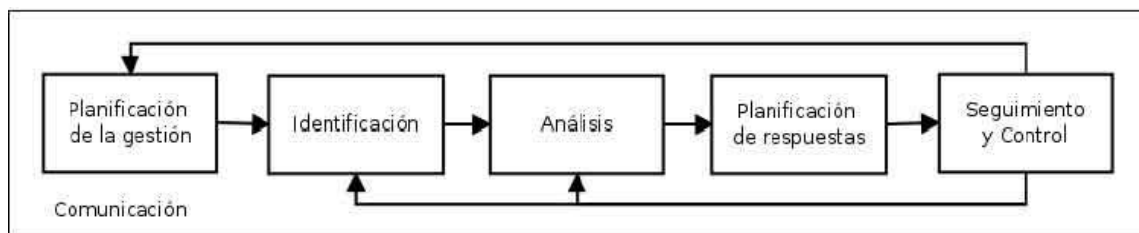


Figura 6. Modelo de procesos de GR, tomado de (Zulueta, 2007).

El funcionamiento del modelo se basa en la realización de un conjunto de actividades por cada proceso, y a su vez el cumplimiento de cada actividad está representado por el desarrollo de diferentes tareas (Zulueta, 2007).

1.3.6 PMBOK

El PMBOK®, es un cuerpo de conocimiento para la GP y estándar nacional de los Estados Unidos desde 2001 (ANSI/PMI 99-001-2000) y propone una guía genérica básica para la GR que involucra la siguiente serie de procesos (PMI, 2004):

Capítulo 1: Fundamentación Teórica

Planeación de la Gestión de Riesgos, Identificación de los riesgos, Análisis cualitativo, Análisis cuantitativo, Planeación de la respuesta a los riesgos y Monitoreo y control. De estos procesos, los más interesantes de mencionar por ser novedosos de entre los hasta ahora estudiados, son los siguientes:

- **Análisis Cualitativo**

El Análisis Cualitativo de Riesgos evalúa la prioridad de los riesgos identificados usando la probabilidad de ocurrencia, el impacto correspondiente sobre los objetivos del proyecto si los riesgos efectivamente ocurren, así como otros factores como el plazo y la tolerancia al riesgo de las restricciones del proyecto como coste, cronograma, alcance y calidad (PMI, 2004).

- **Análisis Cuantitativo**

En este modelo, el Análisis Cuantitativo de los Riesgos se realiza respecto a los riesgos priorizados en el proceso de Análisis Cualitativo por tener un posible impacto significativo sobre las demandas concurrentes del proyecto. Este proceso analiza el efecto de esos riesgos y les asigna una calificación numérica. Tiene como objetivos: cuantificar los posibles resultados del proyecto y sus probabilidades, evaluar la probabilidad de lograr los objetivos específicos del proyecto, identificar los riesgos que requieren una mayor atención mediante la cuantificación de su contribución relativa al riesgo general del proyecto, identificar objetivos de coste, cronogramas o alcance realistas y viables, dados los riesgos del proyecto, y determinar la mejor decisión de dirección de proyectos cuando algunas condiciones o resultados son inciertos (PMI, 2004).

A partir del análisis realizado de los diferentes modelos existentes para el área de la GR dentro de la GP, se decide utilizar el “PMBOK, Guía de Cuerpo y Conocimiento para la Administración de Proyectos” como guía rectora del análisis y GR a realizarse en el módulo que se desea implementar, debido a los siguientes elementos:

- Este modelo se utiliza en el Laboratorio de Gestión de Proyectos como guía para la administración de los mismos.
- Se adecúa además a las disposiciones establecidas en la Resolución No. 60/11 establecida por la Contraloría General de nuestro país.
- Se ajusta a las características del entorno porque los parámetros o elementos que se registran de los proyectos de la universidad son los adecuados para la GR según esta guía.

1.4 Gestión de Riesgos según diferentes herramientas para la Gestión de Proyectos

Los sistemas de GP permiten fundamentalmente la integración de toda la información de los proyectos, cubriendo de alguna manera sus diferentes necesidades funcionales, como pueden ser: gestión de tareas y actividades, gestión de recursos, calendarios,

Capítulo 1: Fundamentación Teórica

colaboración, gestión documental, administración de portafolios y gestión de riesgos. A continuación se caracterizan algunas de ellas de acuerdo a la manera en que administran los riesgos.

1.4.1 Team Foundation Server (TFS)

TFS no es simplemente una herramienta para el control de versiones de código fuente, su alcance va mucho más allá, facilitando la gestión completa del ciclo de vida de la aplicación, desde la fase de diseño hasta las pruebas, pasando por la integración continua o la calidad del código (Márquez, 2008).

La herramienta TFS permite definir y realizar el seguimiento de una lista de riesgos. Proporciona campos para describir el riesgo y la probabilidad de que se produzca en el formulario donde se crea. Permite definir acciones que se llevaran a cabo para administrar esos riesgos, orientadas a reducirlos o mitigarlos, las cuales requieren tiempo y recursos. Estas acciones se documentan en una sección específica para ello, como se muestra en la siguiente figura (MSDN, 2012).

The image shows a screenshot of the TFS Risk Management form. The form is titled 'Riesgo 303*' and has a toolbar with 'Guardar elemento de trabajo', 'Anterior', and 'Siguiente' buttons. The main form area is titled 'Riesgo 303 (modificado): Riesgo 1'. The form contains several sections: 'Título: Riesgo 1', 'Probabilidad: 50', 'Estado' (Asignado a: Michael Affronti, Estado: Propuesto, Motivo: Nuevo), 'Clasificación' (Ruta de acceso del área: CmmiTest1, Iteración: CmmiTest1), and 'Planeación' (Prioridad: 2, Gravedad: 3: Media, Bloqueado: No, Estimación original:). There are four callout boxes with arrows pointing to specific fields: 'Seleccione el área y la iteración' points to the 'Ruta de acceso del área' and 'Iteración' fields; 'Especifique la posibilidad de que se produzca el evento o la condición' points to the 'Probabilidad' field; 'Seleccione propietario' points to the 'Asignado a' field; and 'Indique el evento o condición que supone un riesgo para la programación del proyecto' points to the 'Título' field. A larger callout box at the bottom points to the 'Planeación' section, containing the text: 'Seleccione la prioridad relativa, la gravedad, el estado de la evaluación de errores, si un problema está afectando al progreso y la estimación del esfuerzo para mitigar el riesgo'.

Figura 7. GR en la herramienta TFS, tomado de (MSDN, 2012).

1.4.2 Trac

Trac es una herramienta ligera para el manejo de proyectos basada en la Web, desarrollada en el lenguaje de programación Python por Edgewall Software. Permite ampliar sus funcionalidades por medio de complementos o plugins (Passador, 2010).

La herramienta Trac no gestiona riesgos de proyecto ni de ningún tipo, según la

bibliografía consultada, sin embargo, realiza una administración de tickets, que pueden representar incidencias de proyectos (Passador, 2010).

1.4.3 DotProject

DotProject es una herramienta orientada a la GP. Se enfoca en la administración de recursos para desarrollar un producto, cuya producción requiera de un conjunto de actividades o tareas que se desarrollen entre ellas en forma paralela o independiente. Es una aplicación basada en web, multiusuario, soporta varios lenguajes y es Software libre (CaseySoftware, 2012).

Esta herramienta posee un módulo para la GR, el cual permite adicionar riesgos a un proyecto. Permite establecerle valores como la probabilidad de ocurrencia e impacto, y asignárselos a un integrante del proyecto, quien es el encargado de llevar a cabo las acciones para mitigar dicho riesgo. Además permite establecerle un estado de progreso (CaseySoftware, 2012).

1.4.4 Jira

Jira es una aplicación basada en web para el seguimiento de errores, de incidentes y para la gestión operativa de proyectos. Esta herramienta fue desarrollada por la empresa australiana Atlassian. Inicialmente Jira se utilizó para el desarrollo de software, sirviendo de apoyo para la gestión de requisitos, seguimiento del estatus y más tarde para el seguimiento de errores (Atlassian, 2012).

Jira posee un módulo que le permite la administración de los riesgos del proyecto. Es un módulo muy completo que ofrece muchas funcionalidades como son (Atlassian, 2012):

- Permite gestionar riesgos y darles seguimiento.
- Realiza un análisis cualitativo de los riesgos, a partir de establecer un valor de criticidad para los mismos multiplicando su probabilidad y consecuencia (valores que establece el usuario), y ubicándolos en una matriz ordenada por criticidad que visualiza el usuario.
- Además genera reportes y almacena todas las acciones, permitiendo visualizar riesgos anteriores.

Project	Red issues	Yellow issues	Green issues
StrawberryProject (STRAP)	3	2	3
CherryProject (CHEP)	1	0	1
BananaProject (BAP)	0	1	0

Figura 8. Organización de los Riesgos en la herramienta Jira, tomado de (Atlassian, 2012).

Capítulo 1: Fundamentación Teórica

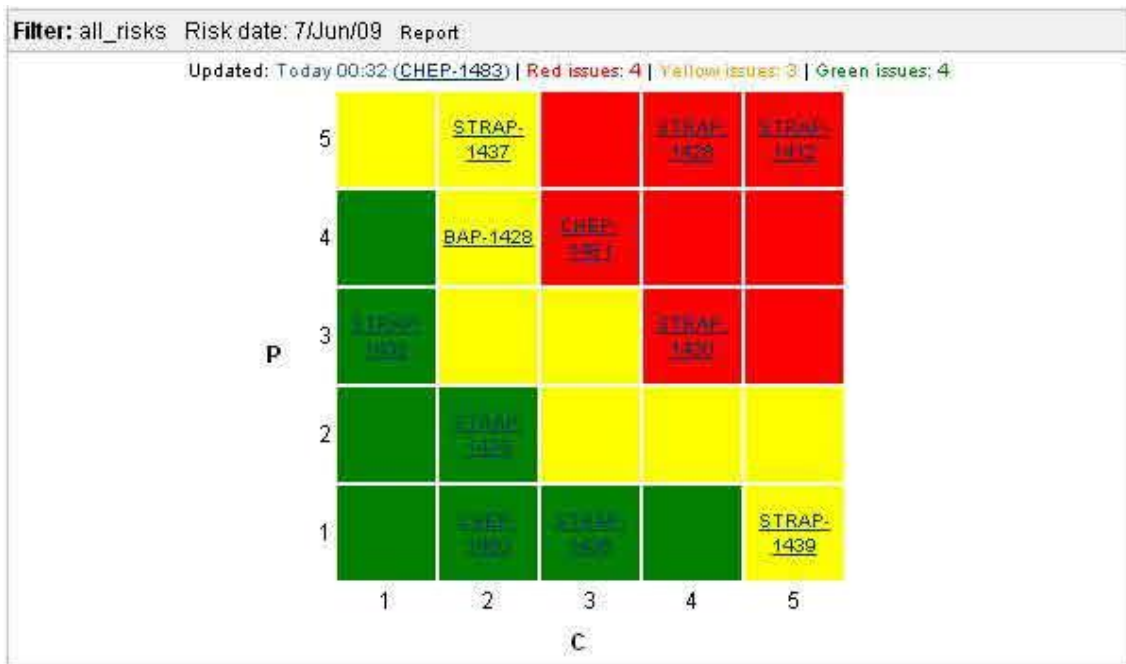


Figura 9. Matriz de criticidad, tomado de (Atlassian, 2012).

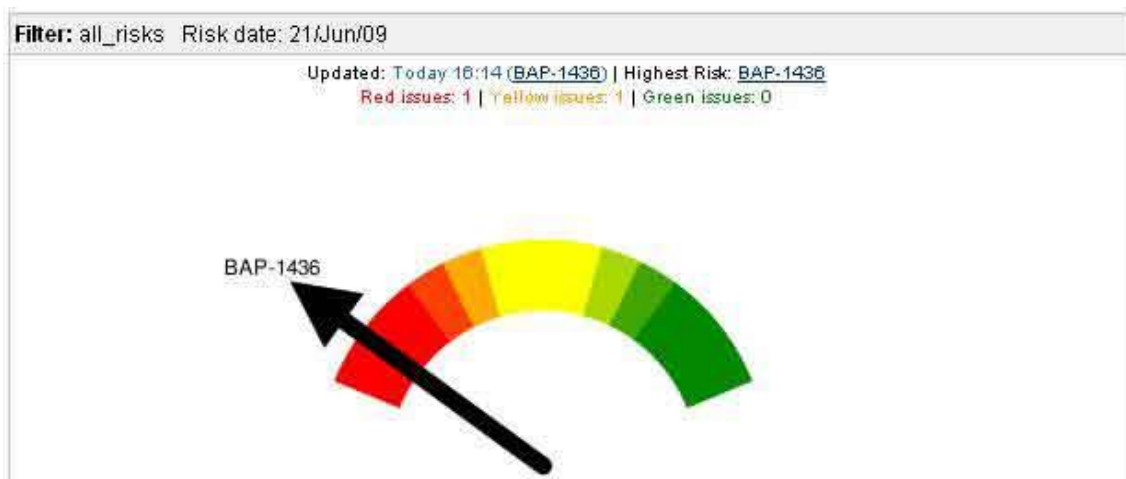


Figura 10. Tablero de alertas de riesgos críticos del proyecto, tomado de (Atlassian, 2012).

Es muy importante destacar que este módulo se encuentra bajo licencia privativa, pues Jira es un producto comercial, y aunque brinda licencias gratis para proyectos Open-Source, instituciones sin ánimo de lucro, organizaciones caritativas y personas individuales, no ha liberado este módulo.

A partir del análisis realizado se puede verificar que existen herramientas de software que administran riesgos a proyectos, algunas de manera más completa que otras. Además se puede comprobar la necesidad de mejorar sustancialmente la GR dentro de la herramienta GESPRO, con el objetivo de elevar sus funcionalidades y lograr competitividad dentro del mercado de herramientas de GP.

1.4.5 GESPRO 11.05

GESPRO en su versión 11.05 es una plataforma extensible que está formada a partir de la integración de más de 18 herramientas libres, comercializadas bajo licencia GPL. Se ha aplicado con buenos resultados en la red de centros de la Universidad de las Ciencias Informáticas, que incluye 13 centros de desarrollo de software ubicados en la sede central de la UCI y cinco centros regionales. Actualmente es utilizado por más de 6000 usuarios que gestionan actividades de más de 150 proyectos entre los que se encuentran proyectos para la informatización nacional como para la exportación. Este sistema ha sido desarrollado utilizando los modelos de líneas de productos de software como modelo industrial de desarrollo (Piñero, 2010).

Este sistema incluye un módulo para la GR, que contiene procesos para la identificación, análisis y respuesta a los riesgos de los proyectos. Este módulo funciona a dos niveles de administración, un nivel de centro y un nivel de proyecto.

A nivel de centro permite:

- Gestionar riesgos genéricos o del centro.
- Gestionar las categorías de los riesgos.

A nivel de proyecto permite:

- Gestionar riesgos del proyecto.
- Gestionar incidencias del proyecto.
- Gestionar desviaciones del proyecto.
- Asociar tareas a los riesgos, las incidencias y las desviaciones del proyecto.
- Visualizar los riesgos genéricos o del centro.
- Visualizar estadísticas relacionadas con los riesgos, las desviaciones y las incidencias del proyecto.

1.5 Metodologías de desarrollo de software

Una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Está formada por fases, cada una de las cuales se puede dividir en sub-fases, que guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo (Piñero, 2009).

Existen diferentes metodologías para el desarrollo de un proyecto de software, las 3 más utilizadas en la UCI son:

- **Rational Unified Process (RUP)**

Capítulo 1: Fundamentación Teórica

El Proceso Unificado de Rational (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 (Jacobson, et al., 2000).

El proceso de software propuesto por RUP tiene tres características esenciales: está dirigido por los Casos de Uso, está centrado en la arquitectura, y es iterativo e incremental. RUP identifica 6 “mejores prácticas”⁴ con las que define una forma efectiva de trabajar para los equipos de desarrollo de software (Jacobson, et al., 2000).

- Gestión de requisitos
 - Desarrollo de software iterativo
 - Desarrollo basado en componentes
 - Modelado visual (usando UML)
 - Verificación continua de la calidad
 - Gestión de los cambios
- **XP**

Constituye una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. (Calero, 2012).

Entre las principales características de esta metodología se encuentran (Calero, 2012):

- Permite introducir nuevos requisitos o cambiar los anteriores ágilmente.
- Adecuado para proyectos pequeños y medianos.
- Adecuado para proyectos con alto riesgo.
- Su ciclo de vida es iterativo e incremental.
- Cada iteración dura entre una y tres semanas.
- No produce demasiada documentación acerca del diseño o la planificación.

⁴ Del inglés: “Best practices”.

Capítulo 1: Fundamentación Teórica

XP se basa fundamentalmente en ser ligero, cercano al desarrollo, basado en Historias de Usuario, fuerte comunicación con el cliente, el código pertenece a todos, programación por parejas y pruebas como base de la funcionalidad. (Calero, 2012).

- **SCRUM**

SCRUM es un proceso ágil de desarrollo de software que nos permite la generación de valor en el mínimo tiempo y además tener un control continuo sobre el estado actual del software. El cliente establece las prioridades y el equipo se auto-organiza para determinar la mejor forma de entregar los requerimientos con más prioridad. SCRUM implica una filosofía de trabajo que no sólo implica al desarrollador sino también al contenido cliente dando prioridad a los individuos y las interacciones sobre los procesos y las tareas, prefiriendo el software funcional sobre la excesiva documentación (Piñero, 2009).

El proceso de desarrollo SCRUM se compone de 5 actividades importantes (Piñero, 2009):

- Planes de lanzamientos.
- Distribución, revisión y ajuste de los estándares de producto: En esta fase se efectúa una revisión de lo que hay que hacer y los detalles de la distribución actual (tecnologías, estándares).
- Sprint: En esta fase es donde el desarrollo de software se lleva a cabo. Un Sprint consta de los siguientes sub-actividades: Elaborar, Integrar, Revisar y Ajustar.
- Revisión del Sprint: Cada Sprint es seguido por una revisión de Sprint. Durante esta revisión, el software desarrollado en el Sprint anterior se revisa y si es necesario se le añaden nuevos ítems del backlog.
- Cierre: En esta fase tienen lugar las actividades de debugging, marketing y promoción. Al acabar esta fase el proyecto quedará cerrado.

Pila de Productos

La pila de producto es una lista priorizada de funciones que el producto debe tener. La prioridad de las funciones siempre estará sujeta a cambios, debido a que hay elementos que dependen de otros y continuamente aparecen nuevos elementos que antes se desconocían (Palacio, et al., 2009).

Lo bueno de Scrum es que la pila de producto es un elemento vivo. Lo normal es mantenerla en una hoja de cálculo o similar, de manera que permita añadir nuevos elementos o reorganizar lo existentes según van sucediendo cosas y cambiando las circunstancias (Palacio, et al., 2009).

Capítulo 1: Fundamentación Teórica

Pila de Sprint

Para realizar un Sprint se cogen una serie de elementos de la Pila de Productos (los que se prevén posibles de realizar en el tiempo estimado) y se construye con ellos la Pila de Sprint (Palacio, et al., 2009).

Al crearla habrá que eliminar los elementos seleccionados de la Pila de Producto, y desglosarlos en tareas concretas, a las que se les puede medir el esfuerzo que llevará realizarlas (Palacio, et al., 2009).

No existe una metodología universal para el desarrollo de un software, sino que de acuerdo con las características de cada proyecto, equipo de trabajo y recursos se exigirá un tipo de metodología específica que brinde la mejor solución al problema. Para esto se tienen en cuenta el siguiente conjunto de parámetros: Personal (se refiere al nivel de conocimiento o preparación del personal involucrado), Dinamismo (se refiere al nivel de modificaciones que sufren los requisitos en el periodo de 1 mes), Criticidad (se refiere al nivel de criticidad de las pérdidas que causaría el proyecto si fallase), Tamaño (se refiere al número de personas involucradas) y Cultura (se refiere al nivel de adaptación del personal a entornos que cambian a menudo), tal y como se muestra en la siguiente figura.

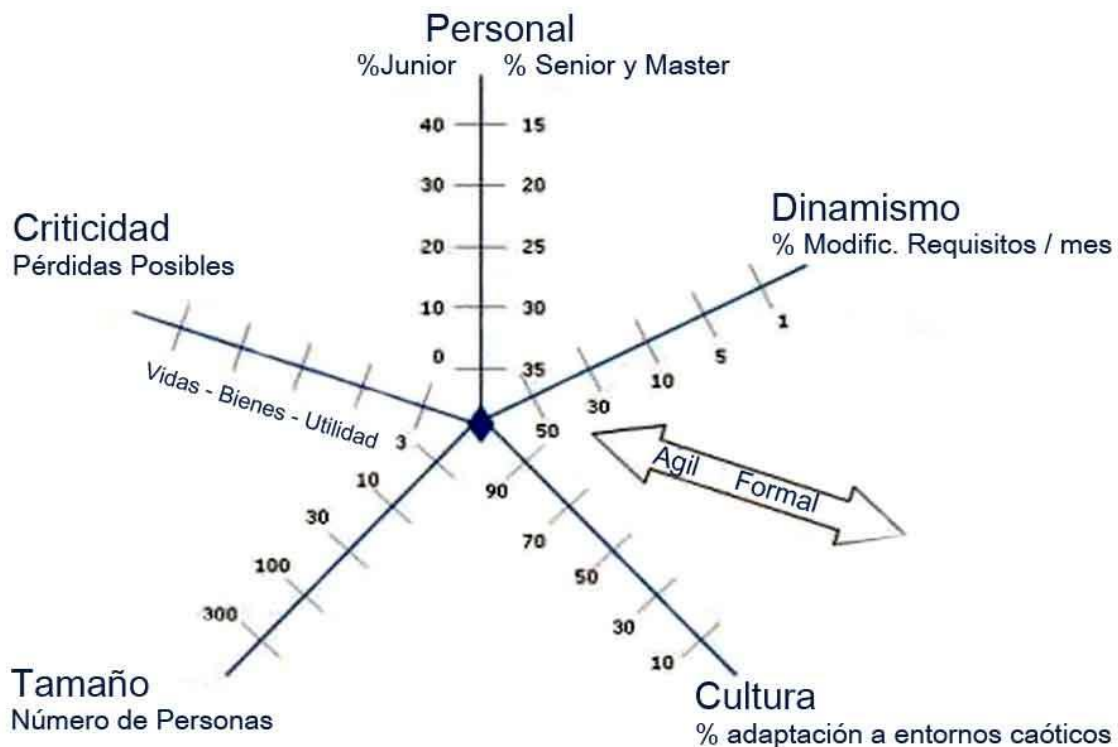


Figura 11. Criterios para la selección apropiada de una metodología, tomado de (Boehm, 1991).

A partir de la valoración realizada de diferentes metodologías de desarrollo de software, se decidió que para la realización del módulo de Gestión de Riesgos versión 2.0 para el sistema GESPRO se utilizará SCRUM como metodología que guiará la

fase de implementación del mismo, debido a los siguientes elementos:

- Scrum es una metodología muy flexible ante el cambio continuo de los requisitos, puesto que pretende controlar el caos producido por conflictos de interés y de necesidades. Ésta condición está presente en el proceso de desarrollo del módulo.
- Scrum propone un intercambio estrecho con el cliente, en el cual éste propone cambios constantes, dando prioridad a los individuos y las interacciones sobre los procesos y las tareas. Ésta condición también está presente.
- Además es Scrum la metodología que se utiliza para la implementación del sistema GESPRO por el equipo de desarrollo del mismo.

1.6 Herramientas de desarrollo de software

Las herramientas para el desarrollo de software desempeñan un importante papel en el desarrollo de aplicaciones. Ante la existencia de numerosas herramientas, es necesario abordar temas genéricos sobre éstas, para facilitar su comprensión y análisis.

1.6.1 Herramientas CASE

Las herramientas CASE (es el acrónimo de Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) constituyen sistemas para el modelado de software, que permiten aumentar la productividad reduciendo el coste de los mismos en términos de tiempo y de dinero, debido a que permiten realizar diseños del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores, entre otras funcionalidades (Itescam, 2012).

Las dos más usadas en el modelado de sistemas en la UCI son:

- **Rational Rose Enterprise Edition**

Rational Rose Enterprise Edition. Es una herramienta CASE desarrollada por Rational Corporation basada en el Lenguaje Unificado de Modelación (UML, por sus siglas en inglés), que permite crear los diagramas que se van generando durante el proceso de Ingeniería en el Desarrollo del Software (Itescam, 2012).

Rational Rose brinda muchas facilidades en la generación de la documentación del software que se esté desarrollando, además posee un gran número de estereotipos predefinidos que facilitan el proceso de modelación del software. Esta herramienta permite completar una gran parte de las disciplinas (flujos fundamentales) de RUP tales como captura de requisitos, análisis y diseño, implementación, control de cambios y gestión de configuración (Itescam, 2012).

- **Visual Paradigm (VP)**

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. También proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. Presenta licencia gratuita y comercial. Es fácil de instalar y actualizar (TodoProgramas, 2012).

Tanto Rational Rose como VP facilitan la organización de los modelos de acuerdo con el orden lógico que proponen las diferentes metodologías, pero además VP permite visualizar, diseñar y documentar diagramas de UML 2.1 en un entorno de diseño intuitivo, mientras que el Rational Rose Enterprise Edition sólo permite el diseño de diagramas de UML 1.x (TodoProgramas, 2012) .

VP ayuda al equipo de desarrollo a controlar el progreso del proyecto y brinda un medio de comunicación. Facilita el modelado de un sistema donde cada involucrado puede crear sus propias vistas arquitectónicas (Vista Lógica, Vista de Procesos, Vista de Desarrollo, Vista Física y Vista de Casos de Uso). Ante cambios que surjan en un negocio, VP permite adaptar rápidamente los modelos realizados a dichos cambios, lo cual evita escribir código sin analizar los cambios en el modelo. Con VP es posible descomponer un modelado de un sistema en unidades controladas, generar código en un lenguaje específico a partir de los diseños realizados.

El equipo de desarrollo de GESPRO sigue el principio de soberanía tecnológica, por esto, y por todo lo anterior planteado, se optó por modelar la solución informática a construir con VP – UML 6.4 Enterprise Edition.

1.6.2 Framework

Para lograr una implementación exitosa del módulo, es necesario primeramente entender el funcionamiento del framework sobre el cual está desarrollado el sistema GESPRO. Esto es un requisito para la creación de la aplicación, ya que los componentes que se desarrollarán para su posterior incorporación en la herramienta GESPRO se implementarán en este lenguaje.

El framework Ruby on Rails también conocido como RoR o Rails, es de código abierto y está escrito en el lenguaje de programación Ruby, el cual es un lenguaje interpretado, reflexivo y orientado a objetos. Creado por el programador japonés Yukihiro "Matz" Matsumoto. Combina una sintaxis inspirada en Python y Perl con características de programación orientada a objetos similares a Smalltalk (Ruby, et al., 2009).

Siguiendo el paradigma de la arquitectura Modelo Vista Controlador (MVC), RoR trata de combinar la simplicidad con la posibilidad de desarrollar aplicaciones del mundo real escribiendo menos código que con otros frameworks y con un mínimo de configuración (Ruby, et al., 2009).

Capítulo 1: Fundamentación Teórica

La arquitectura MVC en RoR está estructurada de la siguiente forma (Ruby, et al., 2009):

- **Modelo:** Es la representación específica de la información con la cual el sistema opera. RoR define en un directorio llamado “models” todas las clases consideradas como modelos, que se utilizan para la comunicación con la base de datos.
- **Vista:** Presenta el modelo en un formato adecuado para interactuar con el usuario. RoR define un conjunto de vistas en un directorio llamado “views” para cada funcionalidad básica, en un sub-directorio con el mismo nombre del controlador al que responden.
- **Controlador:** Responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo. Los controladores administran las acciones de RoR, y se encuentran en un directorio llamado “controllers”.

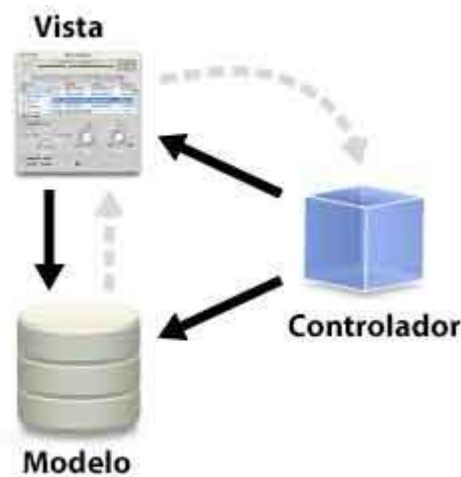


Figura 12. Arquitectura del framework RoR, basado en el patrón MVC.

1.6.3 Gestor de Base de Datos

GESPRO utiliza PostgreSQL, una base de datos de código abierto muy poderosa, dentro de la cual se pueden desarrollar funciones en varios lenguajes. El lenguaje PL/pgSQL es uno de los más utilizados dentro de PostgreSQL, debido a su facilidad de uso y a que guarda cierta similitud con PL/SQL de Oracle. Al usar PL/pgSQL es posible realizar cálculos, manejo de cadenas y consultas dentro del servidor de la base de datos, combinando el poder de un lenguaje procedimental y la facilidad de uso de SQL, minimizando el tiempo de conexión entre el cliente y el servidor (EcuRed, 2012).

PgAdmin III es una aplicación gráfica para el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones

comerciales de PostgreSQL como Pervasive Postgres, EnterpriseDB, Mammoth Replicator y SRA PowerGres (Ubuntu, 2012).

1.6.4 Entornos de Desarrollo Integrado

Un entorno de desarrollo integrado (en inglés Integrated Development Environment o IDE) es un programa compuesto por una serie de herramientas que utilizan los programadores para el desarrollo de código. Los entornos de desarrollo proporcionan un marco de trabajo para la mayoría de los lenguajes de programación existentes en el mercado (por ejemplo C, C++, C#, Java, Python y Visual Basic entre otros). Además es posible que un mismo entorno de desarrollo tenga la posibilidad de utilizar varios lenguajes de programación (Netbeans, 2012).

GESPRO utiliza NetBeans, un reconocido entorno de desarrollo integrado disponible para Windows, Mac, Linux y Solaris. El proyecto NetBeans está formado por un IDE de código abierto y una plataforma de aplicación que permite a los desarrolladores crear con rapidez aplicaciones web, empresariales, de escritorio y móviles utilizando la plataforma Java, así como JavaFX, PHP, JavaScript y Ajax, Ruby y Ruby on Rails, Groovy and Grails y C/C++ (Netbeans, 2012).

1.7 Conclusiones Parciales

En el presente capítulo se realizó un análisis de los conceptos fundamentales asociados a la GP y a la GR, de las disposiciones establecidas en la Resolución 60/11, así como de la administración de los riesgos por diferentes herramientas de GP. A partir de este análisis se escogió la guía del PMBOK como base para fundamentar el proceso de GR que realizará el módulo que se propone implementar, pues se adecúa a lo establecido por la mencionada resolución; además de ser la que se utiliza en el Laboratorio de Gestión de Proyectos de la UCI. También se realizó un estudio de las diferentes metodologías y herramientas de desarrollo de software que se utilizan para la implementación del módulo. A partir de este, se decidió utilizar Scrum como metodología de desarrollo, por las facilidades que brinda a un equipo de trabajo pequeño y con iteraciones de desarrollo cortas; VP – UML 6.4 Enterprise Edition como herramienta de modelado; el framework Ruby on Rails como marco de trabajo; PostgreSQL como gestor de bases de datos y NetBeans 6.9.1 como IDE de desarrollo, por las ventajas de soberanía tecnológica que los mismos brindan.

Capítulo 2: Fundamentos de la solución propuesta

2.1 Introducción

En este capítulo se establece una descripción detallada de las nuevas funcionalidades que se desarrollaron, contenidas en la Pila de Producto según lo dictado por la metodología escogida, y de las acciones que se llevaron a cabo para desarrollarlas. Además se describe la arquitectura del módulo y los diferentes artefactos generados en el proceso de diseño e implementación del mismo, como son los diagramas de clases, el modelo de la base de datos, el de los componentes del sistema y el de despliegue.

2.2 Descripción de la solución propuesta.

La propuesta de solución que se propone (módulo de Gestión de Riesgos versión 2.0) funciona a 2 niveles de administración: un nivel superior, en el que se administran los riesgos del centro, las categorías de los riesgos y los tipos de incidencias; y un nivel inferior, en el que se administran los riesgos, las incidencias y las desviaciones de cada proyecto, se pueden visualizar también los riesgos del centro e importarlos al proyecto.

Siguiendo el modelo establecido por la metodología Scrum, las nuevas funcionalidades del módulo para la Gestión de Riesgos en el sistema GESPRO están contenidas dentro de la siguiente Pila de Producto:

Id	Descripción	Prioridad	Estimación (días)
1	Importar riesgos desde el centro hacia un proyecto.	Alta	12
2	Redefinir la manera en que se calcula la exposición.	Alta	7
3	Formalizar las categorías de los riesgos.	Alta	7
4	Gestionar tipos de incidencias.	Alta	14
5	Generar tareas de mitigación genéricas a riesgos de manera semiautomática.	Alta	7
6	Generar tareas correctivas genéricas a desviaciones de manera semiautomática.	Alta	7
7	Mostrar alertas.	Media	7

Capítulo 2: Fundamentos de la Solución Propuesta

8	Modificar los atributos con los que se almacenan los riesgos del centro.	Media	7
9	Asociar las desviaciones a las incidencias.	Media	7
10	Generación de gráficos de barras para visualizar los riesgos por exposición.	Media	7
11	Exportar a formato '.csv'.	Baja	21
12	Generación de un manual de usuario para el módulo.	Baja	7

Tabla 2. Pila de Productos del módulo de Gestión de Riesgos.

Definida la Pila de Producto se procede a realizar la Pila de Sprint, que es una lista en la que se descomponen en tareas las funcionalidades definidas en la Pila de Productos para posteriormente desarrollarlas en la fase de Sprint.

La Pila de Sprint del módulo de Gestión de Riesgos para el Sistema GESPRO, se compone de tres iteraciones. En la primera iteración se relacionan las tareas necesarias para implementar las funcionalidades de prioridad Alta, en la segunda iteración, las funcionalidades de prioridad Media, y en la tercera las de prioridad Baja.

No. de Iteración	Funcionalidad a desarrollar	Responsable	Duración (semanas)
1	<p>Importar riesgos desde el centro hacia un proyecto.</p> <ul style="list-style-type: none"> • Redefinir el diseño de la vista de los riesgos del centro desde el proyecto de manera que exista una vinculación entre ellos a través de un botón "Importar". • Desarrollar el método 'import_risk' de la clase 'RiskListController', que permita importar un riesgo del centro hacia el proyecto. 	Iván Martínez Vigil	9

Capítulo 2: Fundamentos de la Solución Propuesta

	<p>Redefinir la manera en que se calcula la exposición.</p> <ul style="list-style-type: none">• Es necesario redefinir la manera en que se calcula la exposición. La misma debe calcularse de la siguiente forma: (probabilidad de ocurrencia + impacto + facilidad de detección) / 15.		
	<p>Formalizar las categorías de los riesgos.</p> <ul style="list-style-type: none">• Redefinir el diseño de la vista de categorías de riesgos de manera que las categorías que son por defecto no sean editables.		
	<p>Gestionar tipos de incidencias.</p> <ul style="list-style-type: none">• Definir el diseño de las vistas del CRUD 'Gestionar Tipos de Incidencias', teniendo en cuenta que los tipos de incidencias que son por defecto no son editables.• Adicionar la clase controladora 'ProjectIncidenceTypesController', así como el modelo 'ProjectIncidenceType', y desarrollar los métodos 'index', 'create', 'retrieve', 'update' y 'delete', así como todos los necesarios para el funcionamiento del CRUD.		

Capítulo 2: Fundamentos de la Solución Propuesta

	<p>Generar tareas de mitigación genéricas a riesgos de manera semiautomática.</p> <ul style="list-style-type: none">• Redefinir la vista 'project_risks/issues_index' a partir de la inclusión de un botón "<i>Generar tarea de mitigación</i>".• Crear el método 'generate_mitigation_issue' en la clase 'ProjectRisksController', además de todos los métodos necesarios para garantizar la funcionalidad de generar una tarea de mitigación automáticamente a partir de un riesgo.		
	<p>Generar tareas correctivas genéricas a desviaciones de manera semiautomática.</p> <ul style="list-style-type: none">• Redefinir la vista 'project_deviations/issues_index' a partir de la inclusión de un botón "<i>Generar tarea correctiva</i>".• Crear el método 'generate_mitigation_issue' en la clase 'ProjectDeviationsController', además de todos los métodos necesarios para garantizar la funcionalidad de generar una tarea correctiva automáticamente a partir de una desviación.		

Capítulo 2: Fundamentos de la Solución Propuesta

2	Mostrar Alertas. <ul style="list-style-type: none">• Redefinir el diseño de las vistas de riesgos del proyecto y riesgos del centro desde el proyecto, a partir de la inclusión de una alerta en el caso de que ciertos indicadores (exposición, tiempo planificado y tiempo real) alcancen valores pico. En el caso de la exposición, se deberá pintar de rojo el valor si se encuentra entre 0.75 y 1, de naranja si se encuentra entre 0.3 y 0.75, y de negro si es menor que 0.3. En el caso del tiempo planificado y el tiempo real, se deberá pintar de rojo si el valor es 0.	Iván Martínez Vigil	4
	Modificar los atributos con los que se almacenan los riesgos del centro. <ul style="list-style-type: none">• Es necesario que los formularios de los riesgos del centro y los riesgos de proyecto tengan los mismos campos, siempre que esto sea posible.		

Capítulo 2: Fundamentos de la Solución Propuesta

	<p>Asociar las desviaciones a las incidencias.</p> <ul style="list-style-type: none">• Redefinir la vista 'project_incidentes/index' y generar las vistas 'deviations_index' y 'deviations/_index' de manera que puedan asociarse las desviaciones a las incidencias.• Adicionar los métodos 'deviations_index', 'deviations_new' y 'deviations_delete' a la clase 'ProjectIncidentesController', así como todas las funcionalidades necesarias para garantizar esta funcionalidad.• Modificar las relaciones 'has_many' y 'belongs_to' entre los modelos 'ProjectIncidence' y 'ProjectDeviation'.		
	<p>Generación de gráficos de barras para visualizar los riesgos por exposición.</p> <ul style="list-style-type: none">• Redefinir la vista 'project_risk_statistics/index' a partir de la inclusión de un gráfico de barras que muestre los riesgos agrupados por nivel de exposición. Para ello es necesario incluir y utilizar la biblioteca 'highcharts.js'.• Adicionar los métodos 'get_high_exposure_risks_by_category', 'get_medium_exposure_risks_by_category' y 'get_low_exposure_risks_by_category' al módulo 'ProjectRiskStatisticsHelper' para obtener los valores necesarios para mostrarlos en el gráfico.		

Capítulo 2: Fundamentos de la Solución Propuesta

3	<p>Exportar a formato ‘.csv’.</p> <ul style="list-style-type: none"> • Redefinir el diseño de las vistas de riesgos del centro, riesgos de proyecto, incidencias y desviaciones a partir de la inclusión de un botón “<i>Exportar a csv</i>”. • Desarrollar el método ‘base_risk_export_to_csv’ de la clase ‘BaseRiskApplicationController’, así como todos los métodos que permitan la funcionalidad de exportar riesgos del centro, riesgos del proyecto, incidencias y desviaciones en formato ‘.csv’. 	Iván Martínez Vigil	5
	<p>Realizar cambios en la Base de Datos</p> <ul style="list-style-type: none"> • Realización de un ‘migration’ para modificar las tablas ‘risks’, ‘project_risks’, ‘project_incidences’, ‘project_deviations’, ‘risk_categories’, ‘project_incidence_types’ y ‘deviation_types’ de la base de datos del GESPRO de acuerdo con todas las modificaciones que ha sufrido el módulo de Gestión de Riesgos. 		
	<p>Generación de un manual de usuario para el módulo.</p>		

Tabla 3. Pila de Sprint a realizarse en la fase de Sprint.

2.3 Diseño e implementación del módulo.

A continuación se exponen los fundamentos del diseño del sistema, utilizando como punto de partida la arquitectura del módulo. La misma está basada en el patrón MVC según lo establecido por el framework RoR, el cual separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos.

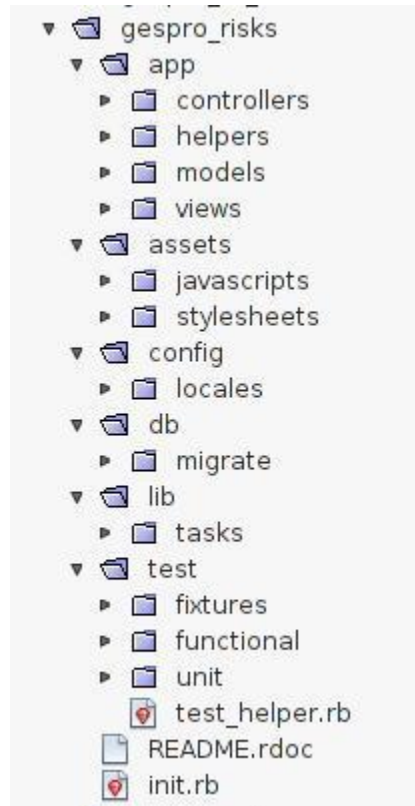
2.3.1 Arquitectura del módulo

EL framework RoR establece en su estructura clásica del patrón Modelo-Vista-

Capítulo 2: Fundamentos de la Solución Propuesta

Controlador un sistema de paquetes y carpetas con la siguiente distribución:

- *app*: Contiene el *core* del módulo y los subdirectorios *controllers*, *helpers*, *models* y *views*.
 - *controllers*: Aquí es donde se encuentran las clases controladoras. El objetivo del controlador es manejar las peticiones web realizadas por los usuarios desde el navegador.
 - *helpers*: En este subdirectorio se almacenan las clases de ayuda que se usan para asistir a las clases controladoras y las vistas principalmente. Esto evita el acumulamiento de código en las clases controladoras y los modelos.
 - *models*: Acumula las clases de modelación de datos almacenados en la base de datos de la aplicación.
 - *views*: Contiene las plantillas de visualización que se llenarán con los datos de la aplicación, las mismas que serán convertidas en HTML y devueltas al navegador del usuario.
- *assets*: Contiene los subdirectorios *javascripts* y *stylesheets*.
 - *javascripts*: Contiene los archivos *javascript*, que proveen de funciones dinámicas al módulo.
 - *stylesheets*: Contienen las hojas de estilo en cascada (*css*), que proveen de estilos visuales al módulo.
- *config*: Contiene el subdirectorio *locales*
 - *locales*: Se almacenan los archivos dedicados a los idiomas de la aplicación permitiendo la fácil migración a cualquiera de estos. Actualmente el módulo brinda soporte para inglés y español.
- *db*: Contiene el subdirectorio *migrate*.
 - *migrate*: Contiene archivos SQL para la creación y modificación de tablas en la base de datos. Los *migrate* permiten trabajar independiente del servidor central, y una vez terminado, es posible migrar la base de datos central.



Capítulo 2: Fundamentos de la Solución Propuesta

- *lib*: Contiene el subdirectorio *tasks*.
 - *tasks*: Contiene archivo de soporte de *Rails* que permiten la migración del módulo.
- *test*: Contiene los subdirectorios *fixtures*, *functional* y *unit*. En sentido general, este directorio permite realizar diversas pruebas, tanto de funcionamiento como de unidad al módulo, para comprobar sus niveles de rendimiento.

Cumpliendo con la estructura del sistema de archivos establecido por el framework RoR, el flujo de información general que se define para la arquitectura del módulo es el siguiente:

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace).
2. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos.
3. El controlador accede al modelo modificándolo de forma adecuada a la acción solicitada por el usuario. Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo.
5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

La siguiente tabla muestra las clases agrupadas por directorios, de la manera en que el framework las organiza:

Directorio	Clases	Descripción
app/control lers	BaseRiskApplicatio nController	Esta es la clase padre de todas las clases controladoras. Contiene todos los métodos genéricos heredados por otras clases. En adición, están implementadas en esta clase todas las funcionalidades que permiten exportar a formato '.csv'.
	ProjectDeviationsC ontroller	Controla el flujo de gestión de desviaciones. Contiene métodos que permiten además generar tareas correctivas genéricas a una desviación.

Capítulo 2: Fundamentos de la Solución Propuesta

	ProjectIncidenceTypesController	Controla el flujo de gestión de tipos de incidencias. Permite asociar un tipo de incidencia a una categoría de riesgos.
	ProjectIncidencesController	Controla el flujo de gestión de incidencias. Contiene métodos que permiten además asociar desviaciones a las incidencias.
	ProjectRiskRelationsController	Controla las relaciones de herencia que existen entre los riesgos de proyectos con ellos mismos. Un riesgo puede ser padre de ninguno o varios riesgos.
	ProjectRiskStatisticsController	Controla el flujo de información que permite visualizar gráficas y tablas en la sección “Estadísticas” que muestren información detallada de los riesgos del proyecto.
	ProjectRisksController	Controla el flujo de gestión de riesgos del proyecto. Permite además asociar tareas de mitigación y tareas afectadas a riesgos.
	RiskCategoriesController	Controla el flujo de gestión de categorías de riesgos.
	RiskListController	Esta clase permite visualizar los riesgos genéricos o del centro desde la vista de proyectos. Permite además la posibilidad de adicionar un riesgo del centro a un proyecto.
	RisksController	Esta es la clase que controla el flujo de gestión de los riesgos genéricos o del centro.
app/helpers	ProjectDeviationsHelper	Extensión del controlador ProjectDeviationsController. Contiene métodos que permiten visualizar determinados valores en las vistas de desviaciones.
	ProjectIncidenceTypesHelper	Extensión del controlador ProjectIncidenceTypesController. Contiene métodos que permiten visualizar determinados valores en las vistas de tipos de incidencias.
	ProjectIncidencesHelper	Extensión del controlador ProjectIncidencesController. Contiene métodos que permiten visualizar determinados valores en las vistas de incidencias.

Capítulo 2: Fundamentos de la Solución Propuesta

	ProjectRiskStatistic sHelper	Extensión del controlador ProjectRiskStatisticsController. Contiene métodos que permiten visualizar determinados valores en la vista de estadísticas.
	ProjectRisksHelper	Extensión del controlador ProjectRisksController. Contiene métodos que permiten visualizar determinados valores y mostrar ciertas alertas en las vistas de riesgos del proyecto.
	RiskCategoriesHel per	Extensión del controlador RiskCategoriesController. Contiene métodos que permiten visualizar determinados valores en las vistas de categorías de riesgos.
	RisksHelper	Extensión del controlador RisksController. Contiene métodos que permiten visualizar determinados valores en las vistas de riesgos del centro.
	RisksListHelper	Extensión del controlador RiskListController. Contiene métodos que permiten visualizar los riesgos del centro desde la vista de riesgos del proyecto.
app/models	DeviationType	Permite la conexión directa con la tabla deviation_types, y contiene la relación de la misma con las desviaciones del proyecto.
	IssuesAffectedProj ectRisk	Permite la conexión directa con la tabla issues_affected_project_risks, que almacena las tareas afectadas por los riesgos del proyecto. Contiene la relación de la misma con las tareas.
	ProjectDeviation	Permite la conexión directa con la tabla project_deviations. Contiene las relaciones de la misma con los usuarios, los proyectos, los riesgos del proyecto, los tipos de desviaciones y las incidencias del proyecto.
	ProjectIncidence	Permite la conexión directa con la tabla project_incidences. Contiene las relaciones de la misma con los usuarios, los proyectos, los riesgos del proyecto, los tipos de incidencias y las desviaciones.

Capítulo 2: Fundamentos de la Solución Propuesta

ProjectIncidenceType	Permite la conexión directa con la tabla <i>project_incidence_types</i> . Contiene las relaciones de la misma con las categorías de los riesgos y las incidencias.	
ProjectRisk	Permite la conexión directa con la tabla <i>project_risks</i> . Contiene las relaciones de la misma con los usuarios, los proyectos, las categorías de los riesgos, los riesgos del centro, los estados de los riesgos del proyecto, las fases del proyecto, las tareas afectadas por los riesgos del proyecto, y con ella misma.	
ProjectRiskRelation	Permite la conexión directa con la tabla <i>project_risk_relations</i> . Contiene las relaciones de la misma con los riesgos del proyecto. Es quien establece la relación de herencia de los riesgos del proyecto con ellos mismos.	
ProjectRiskStatus	Permite la conexión directa con la tabla <i>project_risk_statuses</i> , que representa todos los estados que puede tomar un riesgo del proyecto. Contiene las relaciones con los riesgos del proyecto.	
Risk	Permite la conexión directa con la tabla <i>risks</i> , que representa los riesgos genéricos o del centro. Contiene las relaciones con los usuarios y las categorías de riesgos.	
RiskCategory	Permite la conexión directa con la tabla <i>risk_categories</i> , que representa todas las categorías que puede tener un riesgo. Contiene la relación con los riesgos.	
app/views	deviations	Contiene la vista <i>_index</i> , que lista un conjunto de desviaciones. Es usada desde la vista de incidencias para mostrar las desviaciones asociadas a una incidencia.
	issues	Contiene las vistas <i>_index</i> e <i>_index_affected</i> , que listan un conjunto de tareas. Son usadas desde las vistas de riesgos del proyecto y desviaciones para mostrar las tareas afectadas o generadas por un riesgo del proyecto o una desviación.

Capítulo 2: Fundamentos de la Solución Propuesta

project_deviations	Contiene las vistas <i>_form, create, create_new, index, issues_index, retrieve</i> y <i>update</i> , que permiten la gestión de desviaciones de proyectos y la asociación de tareas correctivas y afectadas a las mismas.
project_incidence_types	Contiene las vistas <i>_form, create, index, retrieve</i> y <i>update</i> , que permiten la gestión de los tipos de incidencias y la asociación de estas a las categorías de riesgos.
project_incidences	Contiene las vistas <i>_form, create, deviations_index, index, retrieve</i> y <i>update</i> , que permiten la gestión de incidencias del proyecto y la asociación de desviaciones a las mismas.
project_risk_relations	Contiene la vista <i>_form</i> , que permite crear una nueva relación de herencia entre riesgos del proyecto.
project_risk_statistics	Contiene las vistas <i>_sidebar, _simple, deviations_status, deviations_types, index, risk_category, risk_exposure</i> y <i>risk_status</i> , que permiten visualizar parámetros de importancia para los riesgos del proyecto en la pestaña “Estadísticas” a través de tablas y gráficos.
project_risks	Contiene las vistas <i>_form, _relations, _select_risk, create, index, issues_affected_index, issues_index, retrieve</i> y <i>update</i> , que permiten la gestión de riesgos del proyecto, además de asociar tareas de mitigación y tareas afectadas a los mismos. También permiten crear relaciones de herencia entre los riesgos.
risk_admin_section	Contiene las vistas <i>_header</i> e <i>index</i> , que permiten visualizar los <i>tabs</i> (pestañas) que brindan acceso al usuario a las diferentes secciones con que cuenta el módulo.
risk_categories	Contiene las vistas <i>_form, create, index, retrieve</i> y <i>update</i> , que permiten la gestión de categorías de riesgos.

Capítulo 2: Fundamentos de la Solución Propuesta

	risk_list	Contiene las vistas <i>index</i> y <i>retrieve</i> , que permiten visualizar los riesgos del centro desde la vista de riesgos del proyecto y adicionarlos al mismo.
	risks	Contiene las vistas <i>_form</i> , <i>create</i> , <i>index</i> , <i>retrieve</i> y <i>update</i> , que permiten la gestión de riesgos del centro.
assets/java scripts	highcharts	Librerías <i>javascripts</i> utilizadas para dibujar gráficos de barras en la vista de estadísticas.
	risks	Contiene las funcionalidades <i>javascripts</i> utilizadas en el módulo principalmente para el completamiento dinámico de tareas.
assetes/stylesheets	highslide	Conjunto de clases de estilo que utiliza la biblioteca <i>highcharts</i> para dibujar gráficos.
	risks	Contiene las clases de estilo que utiliza el módulo.

Tabla 4. Estructura de clases por directorios del módulo de Gestión de Riesgos del Sistema GESPRO.

2.3.2 Diagramas de Clases

Para la construcción del diagrama de clases del módulo, fue necesario modelar todas las clases persistentes, controladoras y vistas que intervienen en los procesos de administración que éste maneja. Dicho diagrama por su extensión, fue necesario dividirlo en cuatro sub-diagramas más pequeños, que ayudan a entender mejor la estructura y funcionamiento del módulo.

2.3.2.1 Sub-diagrama de clases de los riesgos del proyecto.

La siguiente figura muestra la relación entre las clases principales que intervienen en la administración de los riesgos a nivel de proyecto:

Capítulo 2: Fundamentos de la Solución Propuesta

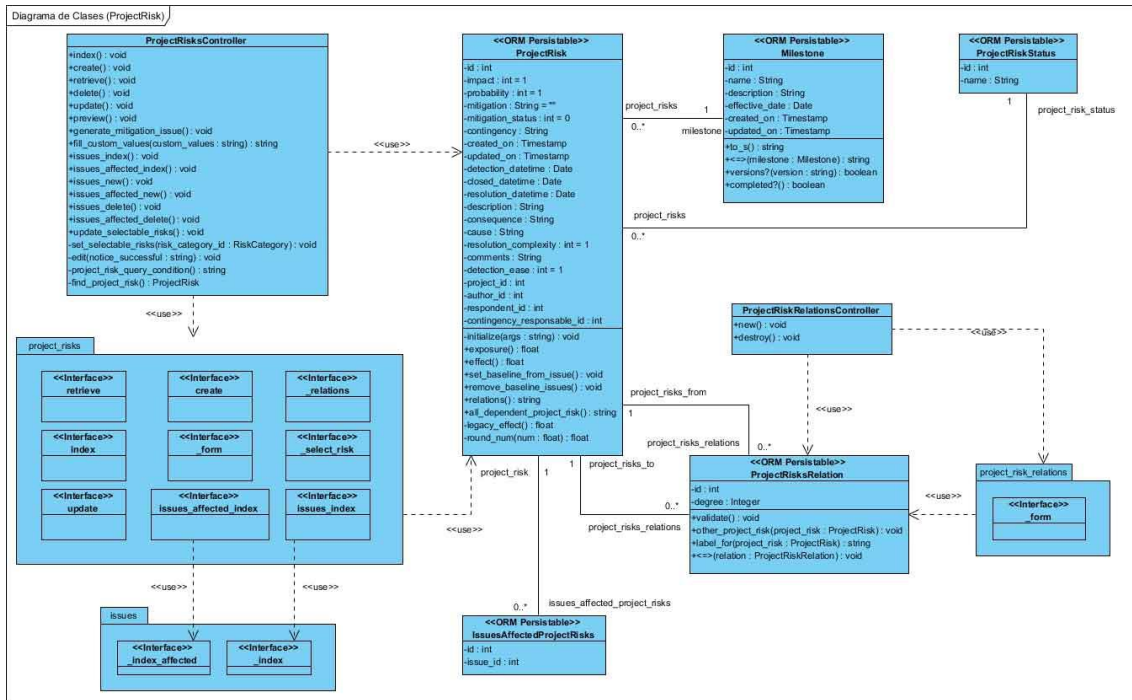


Figura 13. Diagrama de Clases de los riesgos del proyecto.

2.3.2.2 Sub-diagrama de clases de los riesgos del centro

La siguiente figura muestra la relación entre las clases principales que intervienen en la administración de los riesgos a nivel de centro:

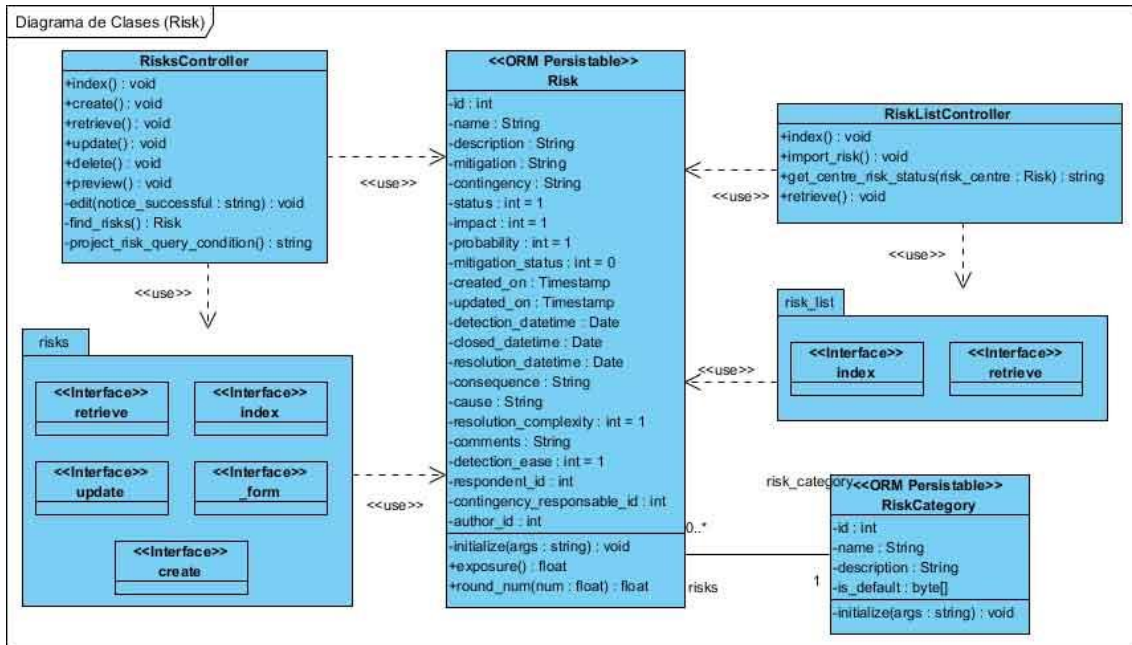


Figura 14. Diagrama de Clases de los riesgos del centro.

2.3.2.3 Sub-diagrama de clases de las incidencias del proyecto

Capítulo 2: Fundamentos de la Solución Propuesta

La siguiente figura muestra la relación entre las clases principales que intervienen en la administración de las incidencias del proyecto:

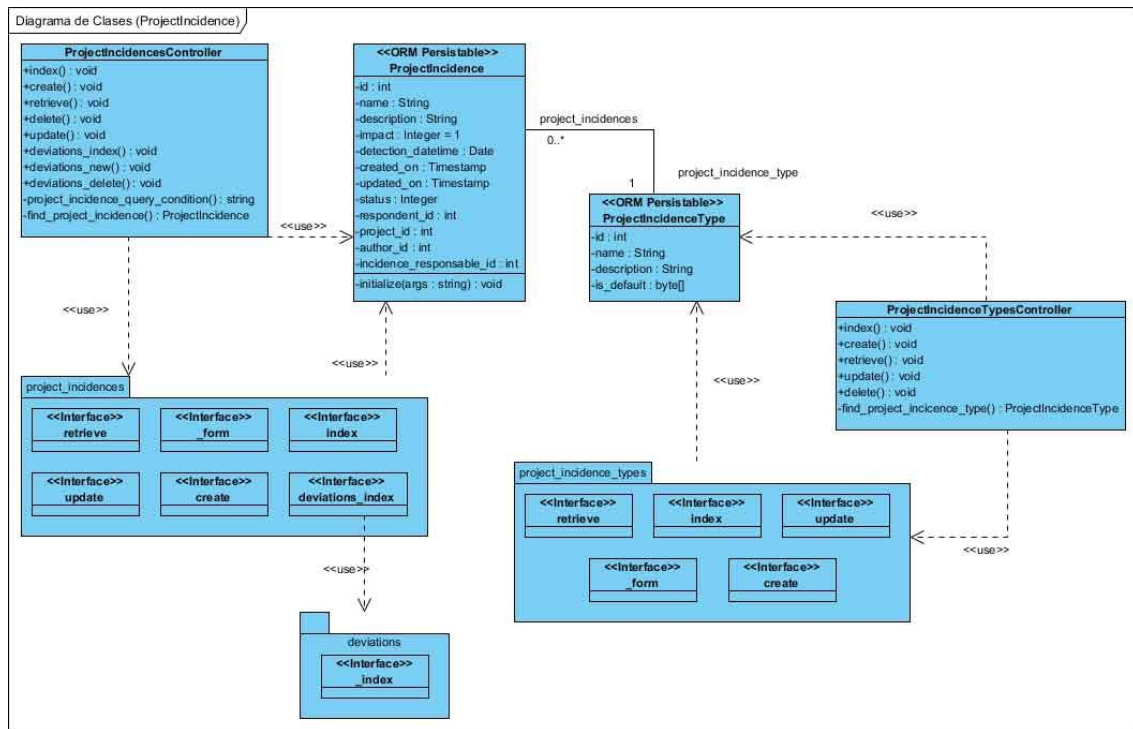


Figura 15. Diagrama de Clases de las incidencias del proyecto.

2.3.2.4 Sub-diagrama de clases de las desviaciones del proyecto

La siguiente figura muestra la relación entre las clases principales que intervienen en la administración de las desviaciones del proyecto:

Capítulo 2: Fundamentos de la Solución Propuesta

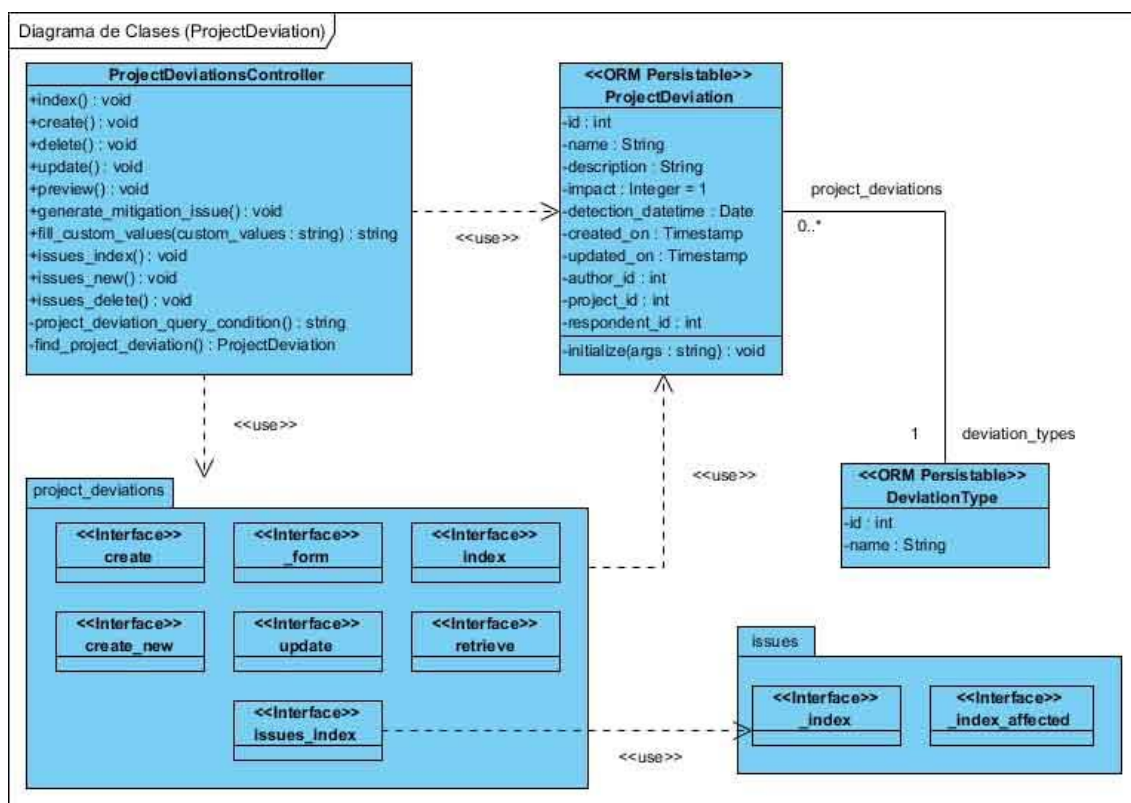


Figura 16. Diagrama de Clases de las desviaciones del proyecto.

En la tabla 4, anteriormente expuesta se puede ver detalladamente la información de cada clase diseñada.

2.3.3 Patrones de Diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Entre ellos se encuentran los patrones GRASP (es el acrónimo de General Responsibility Assignment Software Patterns), en español significa Patrones Generales de Software para Asignar Responsabilidades. Los patrones GRASP son aquellos que describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Los utilizados para el módulo son los siguientes:

- **Alta cohesión:** Se aplica para realizar un diseño que evite contener clases con un alto grado de abstracción, delegando las responsabilidades muy complejas a otros objetos. Se diseñaron las clases de forma tal que contengan las mínimas responsabilidades necesarias y colaboren con otras para llevar a cabo una tarea. Se tienen las clases controladoras que se encargan de ejecutar acciones de acuerdo con las peticiones que le llegan y las clases de acceso a datos (modelos) que interactúan con la base de datos. Además se cuenta con los “helpers”, que ayudan a no cargar de código innecesario a los controladores.

Capítulo 2: Fundamentos de la Solución Propuesta

- Bajo Acoplamiento: El acoplamiento mide la fuerza con que una clase está conectada a otra, de esta forma una clase con bajo acoplamiento debe tener un número mínimo de dependencias con otras clases. Las diferentes clases controladoras sólo dependen de un único controlador frontal para realizar sus funcionalidades, en este caso “BaseRiskApplicationController”. Una clase de acceso a datos solo depende de la clase controladora que la usa, así se pueden realizar cambios en cada clase de forma independiente.
- Creador: Se aplica para la asignación de responsabilidades a las clases relacionadas con la creación de objetos, de forma tal que una instancia de un objeto sólo pueda ser creada por el objeto que contiene la información necesaria para ello. De esta forma una clase controladora puede crear objetos, pero es el modelo quien sabe cómo crearse.
- Patrón Indirección: El patrón de indirección nos aporta mejorar el bajo acoplamiento entre dos clases asignando la responsabilidad de la mediación entre ellos a un tercer elemento. Este patrón es muy útil cuando es necesario tratar con relaciones complejas entre clases. Por ejemplo, los riesgos del proyecto presentan una relación de mucho a mucho con ellos mismos, pues un riesgo puede ser hijo y a la vez padre de varios riesgos. Para resolver este problema se crea la clase “ProjectRiskRelationsController”, que controla esta relación.
- Patrón Polimorfismo: Siempre que se tenga que llevar a cabo una responsabilidad que dependa de tipos de objetos, se tiene que hacer uso del polimorfismo, cuando las alternativas o comportamientos relacionados varían según éste. Este patrón se tuvo presente debido a la importancia atribuida al realizar un diseño de clases independientes que puedan soportar los cambios de una manera fácil y a la vez permitan la reutilización.

2.3.4 Diseño de política de seguridad informática

- Niveles de acceso de los usuarios en la aplicación: El módulo de Gestión de Riesgos para el sistema GESPRO ajusta su control de acceso a los permisos que brinda la plataforma en general para los roles previamente configurados y definidos.
- Protección de datos: Los servidores de base de datos cuentan con un firewall instalado el cual controla estrictamente el acceso desde las subredes del nodo central donde se encuentran hosteados los portales de la plataforma GESPRO y desde la subred de laboratorio donde se desarrolla GESPRO.
- Recuperación: GESPRO tiene un sistema de respaldo a través de un nivel de salvas que se realizan diariamente con la herramienta Bacula. Estas salvas se encuentran físicamente en los servidores de uno de los docentes de la Universidad (docente 4), garantizando además la recuperación de datos en caso de catástrofe total en el nodo central.

2.3.5 Modelo Entidad Relación

GESPRO como ecosistema⁵ posee una extensa base de datos de más de 140 tablas, que utiliza varios modelos, entre los que son muy interesantes destacar el modelo de bases de datos transaccionales y el modelo de bases de datos relacionales.

El módulo de riesgos utiliza la estructura establecida por GESPRO, y además incorpora nuevas tablas necesarias para el funcionamiento del mismo. La siguiente figura muestra la estructura de la base de datos del módulo de Gestión de Riesgos para el sistema GESPRO:

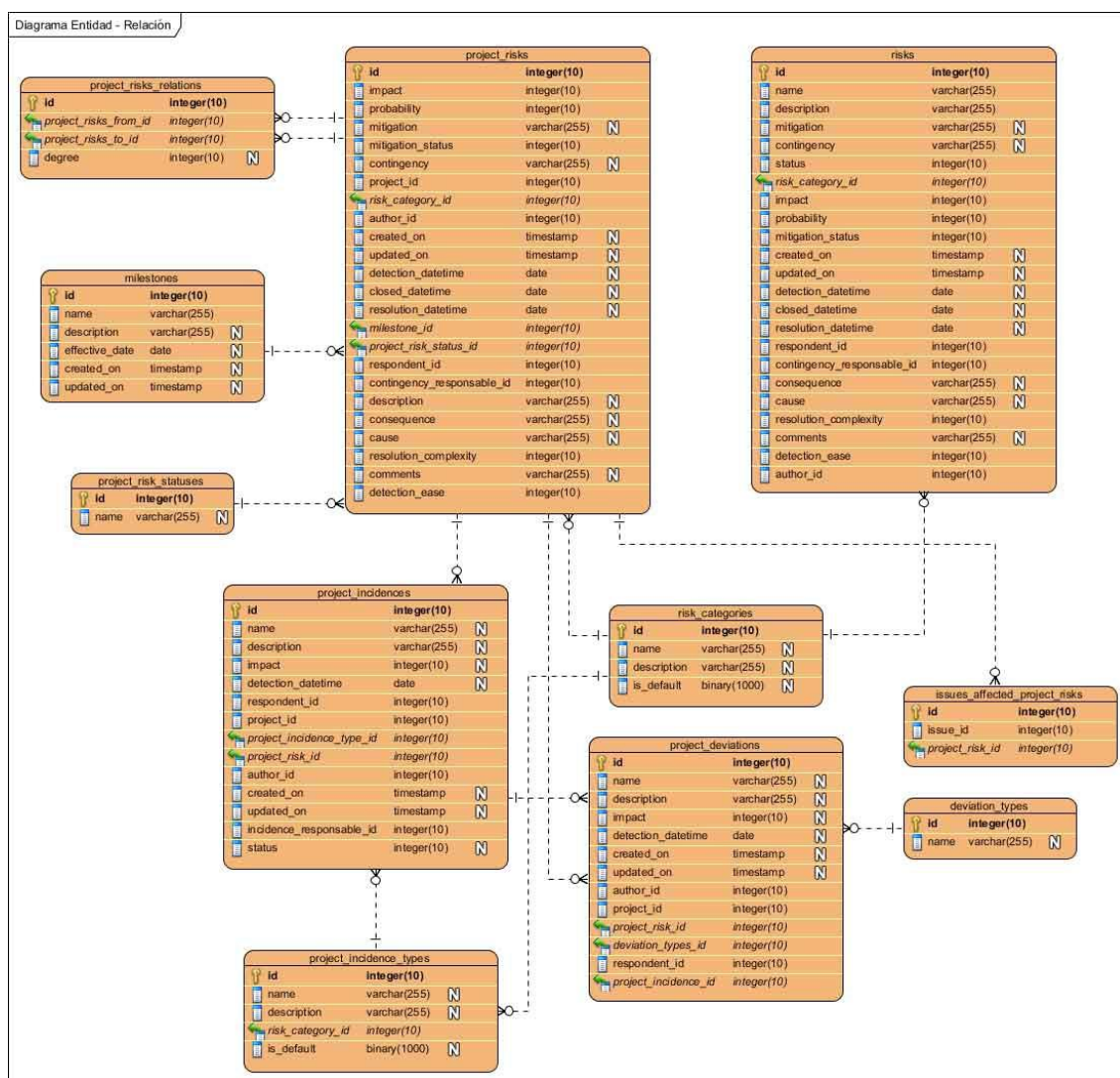


Figura 17. Diseño de Base de Datos del módulo de Gestión de Riesgos.

A continuación se establece una relación de las 4 tablas más importantes, dando una breve descripción de las mismas, y explicando sus atributos y relaciones:

⁵ Según (Santamaría, 2012): Es una infraestructura digital auto-organizada que intenta lograr un entorno digital para organizaciones en red que dan soporte a la cooperación, compartición de contenido, el desarrollo de tecnologías abiertas y adaptativas y modelos de negocio evolucionistas.

Capítulo 2: Fundamentos de la Solución Propuesta

Nombre	Descripción	Relaciones
project_risks	Almacena los riesgos del proyecto	<p>project_risks_relations: Relación de mucho a mucho que establece una herencia entre los riesgos del proyecto.</p> <p>milestones: Relación del riesgo del proyecto, con la fase del proyecto en la que se identifica.</p> <p>project_risk_statuses: Relación del riesgo con su estado.</p> <p>project_incidents: La nueva versión del módulo establece relaciones entre los riesgos del proyecto y las incidencias, de manera que a un riesgo del proyecto es posible asociarle incidencias.</p> <p>project_deviations: El módulo está definido de manera tal que los riesgos generen incidencias, y estas a su vez generen desviaciones, sin embargo, una desviación puede generarse a partir de un riesgo sin que medie ninguna incidencia.</p> <p>risk_categories: Relación del riesgo con su categoría.</p> <p>issues_affected_project_risks: Relación de las tareas afectadas por los riesgos.</p>
Atributos:		
id: Identificador. Llave primaria.		
name: Nombre que se le dará al riesgo. Campo de texto.		
impact: Mide la gravedad de los efectos adversos si el riesgo llega a producirse dentro del proyecto. Campo numérico.		
probability: Mide la probabilidad de que las consecuencias del riesgo lleguen a producirse realmente. Campo numérico.		
mitigation: Descripción de las actividades de mitigación que se deben realizar. Campo de texto.		
mitigation_status: Estado en el que se encuentra la mitigación del riesgo. Campo numérico.		

Capítulo 2: Fundamentos de la Solución Propuesta

contingency: Descripción de las actividades de contingencia ante la ocurrencia del riesgo. Campo de texto.
project_id: Representa el proyecto al que pertenece el riesgo. Llave foránea de la tabla projects.
risk_category_id: Representa la categoría del riesgo. Llave foránea de la tabla risk_categories.
autor_id: Representa el usuario que entró el riesgo. Llave foránea de la tabla users.
created_on: Fecha en la que fue creado el riesgo. Campo de fecha.
updated_on: Última fecha en la que fue modificado el riesgo. Campo de fecha.
detection_datetime: Fecha en que se detecta el riesgo. Campo de fecha.
closed_datetime: Fecha en que ya el riesgo no presenta ninguna probabilidad de ocurrir. Campo de fecha.
resolution_datetime: Fecha en la que se planea que ya esté resuelto el riesgo. Campo de fecha.
milestone_id: Representa la fase del proyecto en la que ocurre el riesgo. Llave foránea de la tabla milestones.
project_risk_status_id: Representa el estado del riesgo. Llave foránea de la tabla project_risk_statuses.
respondent_id: Representa el usuario que identificó el riesgo. Llave foránea de la tabla users.
contingency_responsible_id: Representa aquella persona que se le encargó la contingencia del riesgo. Llave foránea de la tabla users.
description: Descripción detallada del riesgo. Campo de texto.
consequence: Descripción de las consecuencias que pudiera traer la ocurrencia del riesgo. Campo de texto.
cause: Descripción de la o las causas que provocaron la ocurrencia del riesgo. Campo de texto.
resolution_complexity: Mide el nivel de complejidad que representa la resolución del riesgo. Campo numérico.
comments: Datos de interés relacionados con el riesgo. Campo de texto
detection_ease: Mide el nivel de visibilidad del riesgo, que permita su detección antes de que ocurra. Campo numérico.

Capítulo 2: Fundamentos de la Solución Propuesta

Nombre	Descripción	Relaciones
risks	Almacena los riesgos del centro	risk_categories: Relación del riesgo con su categoría.
Atributos:		
id: Identificador. Llave primaria.		
name: Nombre que se le dará al riesgo. Campo de texto.		
impact: Mide la gravedad de los efectos adversos si el riesgo llega a producirse dentro del proyecto. Campo numérico.		
probability: Mide la probabilidad de que las consecuencias del riesgo lleguen a producirse realmente. Campo numérico.		
mitigation: Descripción de las actividades de mitigación que se deben realizar. Campo de texto.		
mitigation_status: Estado en el que se encuentra la mitigación del riesgo. Campo numérico.		
contingency: Descripción de las actividades de contingencia ante la ocurrencia del riesgo. Campo de texto.		
risk_category_id: Representa la categoría del riesgo. Llave foránea de la tabla risk_categories.		
autor_id: Representa el usuario que entró el riesgo. Llave foránea de la tabla users.		
created_on: Fecha en la que fue creado el riesgo. Campo de fecha.		
updated_on: Última fecha en la que fue modificado el riesgo. Campo de fecha.		
detection_datetime: Fecha en que se detecta el riesgo. Campo de fecha.		
closed_datetime: Fecha en que ya el riesgo no presenta ninguna probabilidad de ocurrir. Campo de fecha.		
resolution_datetime: Fecha en la que se planea que ya esté resuelto el riesgo. Campo de fecha.		
respondent_id: Representa el usuario que identificó el riesgo. Llave foránea de la tabla users.		
contingency_responsible_id: Representa aquella persona que se le encargó la contingencia del riesgo. Llave foránea de la tabla users.		

Capítulo 2: Fundamentos de la Solución Propuesta

description: Descripción detallada del riesgo. Campo de texto.		
consequence: Descripción de las consecuencias que pudiera traer la ocurrencia del riesgo. Campo de texto.		
cause: Descripción de la o las causas que provocaron la ocurrencia del riesgo. Campo de texto.		
resolution_complexity: Mide el nivel de complejidad que representa la resolución del riesgo. Campo numérico.		
comments: Datos de interés relacionados con el riesgo. Campo de texto		
detection_ease: Mide el nivel de visibilidad del riesgo, que permita su detección antes de que ocurra. Campo numérico.		
Nombre	Descripción	Relaciones
project_incidences	Almacena las incidencias del proyecto	<p>project_risks: Una incidencia puede estar asociada a un riesgo.</p> <p>project_deviations: Una incidencia puede generar ninguna o varias desviaciones.</p> <p>project_incidence_types: Relación de la incidencia con su tipo.</p>
Atributos:		
id: Identificador. Llave primaria.		
name: Nombre que se le dará a la incidencia. Campo de texto.		
impact: Mide la gravedad de los efectos adversos de la incidencia. Campo numérico.		
project_id: Representa el proyecto al que pertenece la incidencia. Llave foránea de la tabla projects.		
autor_id: Representa el usuario que entró la incidencia. Llave foránea de la tabla users.		
created_on: Fecha en la que fue creada la incidencia. Campo de fecha.		
updated_on: Última fecha en la que fue modificada la incidencia. Campo de fecha.		
detection_datetime: Fecha en que se detecta la incidencia. Campo de fecha.		
respondent_id: Representa el usuario que identificó la incidencia. Llave foránea de la tabla users.		
description: Descripción detallada de la incidencia. Campo de texto.		

Capítulo 2: Fundamentos de la Solución Propuesta

project_incidence_type_id: Representa el tipo de incidencia. Llave foránea de la tabla <i>project_incidences_types</i>		
project_risk_id: Representa el riesgo con el que está relacionada la incidencia. Llave foránea de la tabla <i>project_risks</i> .		
incidence_responsible_id: Representa el usuario responsable de la incidencia. Llave foránea de la tabla <i>users</i> .		
status: Estado de la incidencia. Campo numérico.		
Nombre	Descripción	Relaciones
project_deviations	Almacena las desviaciones del proyecto.	<p>project_risks: Una desviación puede estar asociada a un riesgo del proyecto.</p> <p>project_incidents: Una desviación puede asociarse, incluso generarse a partir de una incidencia.</p> <p>deviation_types: Relación de la desviación con su tipo.</p>
Atributos:		
id: Identificador. Llave primaria.		
name: Nombre que se le dará a la desviación.		
impact: Mide la gravedad de los efectos adversos de la desviación. Campo numérico.		
autor_id: Representa el usuario que entró la desviación. Llave foránea de la tabla <i>users</i> .		
project_id: Representa el proyecto al que pertenece la desviación. Llave foránea de la tabla <i>projects</i> .		
created_on: Fecha en la que fue creada la desviación. Campo de fecha.		
updated_on: Última fecha en la que fue modificada la desviación. Campo de fecha.		
detection_datetime: Fecha en que se detecta la desviación. Campo de fecha.		
respondent_id: Representa el usuario que identificó la desviación. Llave foránea de la tabla <i>users</i> .		
description: Descripción detallada de la desviación. Campo de texto.		
comments: Datos de interés relacionados con la desviación. Campo de texto.		

Capítulo 2: Fundamentos de la Solución Propuesta

project_risk_id: Representa el riesgo con el que está relacionada la incidencia. Llave foránea de la tabla *project_risks*.

deviation_types_id: Representa el tipo de desviación. Llave foránea de la tabla *deviation_types*.

project_incidence_id: Representa la incidencia con la que está relacionada la desviación. Llave foránea de la tabla *project_incidences*.

Tabla 5. Descripción, relaciones y atributos de las tablas más importantes del módulo.

2.3.6 Diagrama de Componentes

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre éstos. Para su mejor entendimiento, en el diagrama de componentes del módulo de Gestión de Riesgos se agruparon los componentes comunes en paquetes, y se definió entonces solamente la relación entre éstos. De esta manera, todos los componentes controladores están dentro del paquete “controllers”, los componentes *vistas* están dentro del paquete “views”, los componentes *modelos* están dentro del paquete “models”, y los componentes *helpers* están dentro del paquete con su mismo nombre. También se representan algunos componentes externos, como las bibliotecas gráficas utilizadas para la generación de gráficos de barras, las clases de estilos (css), las clases javascripts, y la base de datos.

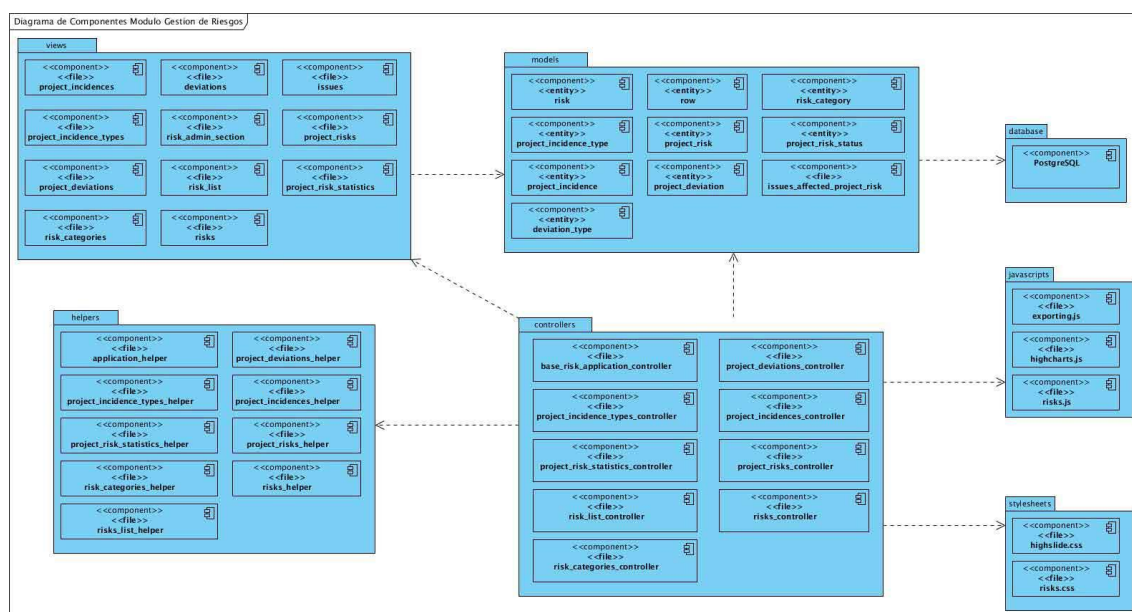


Figura 18. Diagrama de Componentes del módulo de Gestión de Riesgos.

2.3.7 Diagrama de Despliegue

El diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en una aplicación. Las máquinas físicas y los procesadores se representan como nodos. El nodo Servidor BD está conectado con el nodo Servidor

Capítulo 2: Fundamentos de la Solución Propuesta

Web a través de un protocolo de red TCP/IP, y éste a su vez está conectado con los nodos clientes a través de protocolos de comunicación HTTP, garantizando que en cada estación de trabajo los usuarios tengan acceso a la herramienta. GESPRO está instalado en un servidor web Apache, que utiliza PostgreSQL como SGBD (Sistema de Gestión de Bases de Datos), encargado de almacenar los datos con que trabajará la aplicación.

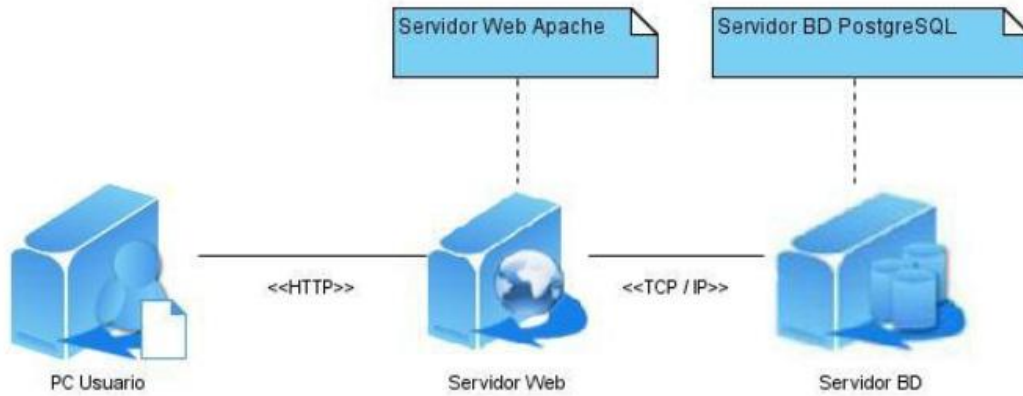


Figura 19. Diagrama de Despliegue del módulo de Gestión de Riesgos.

2.4 Conclusiones Parciales

En este capítulo se plantean algunos conceptos necesarios para el diseño del módulo. Se describe detalladamente la arquitectura del sistema, y se puede comprobar su consistencia y robustez. Se explican los patrones de diseño utilizados, y se muestran los diferentes diagramas de clases, de bases de datos, de componentes y de despliegue generados. Partiendo de todo lo planteado y analizado en este capítulo se afirma que se ha logrado una eficaz implementación, pues el diseño desarrollado cumple con las métricas arquitectónicas necesarias para su correcto funcionamiento e integración en el sistema. Por tanto se concluye que en este capítulo se implementó un módulo para el GESPRO en estado funcional.

Capítulo 3: Análisis de los resultados

3.1 Introducción

En el siguiente capítulo se precisan los casos de prueba que se utilizarán por los probadores para validar la calidad del producto. Además se realiza un análisis del impacto de la aplicación de la nueva versión del módulo a partir de una comparación de las funcionalidades que este brinda con respecto al resto de las herramientas estudiadas. Por último se exponen los resultados de la aplicación de las pruebas de caja negra aplicadas al sistema con el objetivo de validar su correcto funcionamiento.

3.2 Diseño de Casos de Pruebas

Para la posterior evaluación de las funcionalidades implementadas en el módulo, se diseñaron 20 casos de pruebas, que serán útiles a los probadores para detectar funciones incorrectas, como por ejemplo, errores en la interfaz, rendimiento o acceso a la información almacenada.

Los casos de pruebas diseñados se encuentran en la carpeta Anexos, en el CD que se adjunta a la investigación.

3.3 Análisis comparativo de la herramienta GESPRO 12.05 respecto a otras herramientas de Gestión de Proyectos

A partir del estudio realizado de la guía “PMBOK, Una Guía de Cuerpo y Conocimiento para la Administración de Proyectos”, se establecen para el análisis comparativo de la herramienta GESPRO 12.05 con las herramientas mencionadas en el Capítulo 1, un conjunto de fases, que representan los procesos que dicta dicha guía para la realización de la GR de un proyecto. Dicho análisis busca validar la mejora sustancial de la GR realizada por el sistema GESPRO una vez aplicadas las nuevas funcionalidades desarrolladas para el mismo. Se tienen en cuenta por tanto como criterios de comparación las nuevas funcionalidades que brinda el módulo de Gestión de Riesgos y algunas que brindan el resto de las herramientas.

Las fases que representan los procesos que dicta el PMBOK son las siguientes:

- ✓ Fase 1: Identificación de los riesgos
- ✓ Fase 2: Análisis cualitativo
- ✓ Fase 3: Análisis cuantitativo
- ✓ Fase 4: Planeación de la respuesta a los riesgos
- ✓ Fase 5: Monitoreo y control

Capítulo 3: Análisis de los Resultados

A continuación se muestra el análisis comparativo, donde se agrupan cada una de las funcionalidades según la fase a la que respondan:

Fase según el PMBOK	Criterio de Comparación	TFS	Trac	dotProject	Open Proj	Jira	GESPRO 11.05	GESPRO 12.05
Fase 1	Gestión de Riesgos (GR) a un proyecto	Si	No	Si	NE ⁶	Si	Si	Si
	Gestión de Incidencias (GI) a un proyecto	Si	Si	No	NE	No	Si	Si
	Gestión de Desviaciones (GD) a un proyecto	No	No	No	NE	No	Si	Si
	Facilidades de copia entre riesgos de proyectos o de una base de conocimientos genérica	No	No	No	NE	NE	No	Si
Fase 2	Análisis cualitativo de riesgos	Si	No	Si	NE	Si	Si	Si
	Mostrar Alertas a partir de indicadores	No	No	No	NE	Si	No	Si
Fase 3	Análisis cuantitativo de riesgos	NE	No	NE	NE	NE	No	No
Fase 4	Asociación de tareas a riesgos (TM ⁷ o TA ⁸)	Si	No	Si	NE	Si	Si	Si

⁶ NE: No encontrado en la bibliografía consultada.

⁷ Tarea de Mitigación

⁸ Tarea Afectada

Capítulo 3: Análisis de los Resultados

	Asociación de desviaciones a incidencias (DG ⁹)	No	No	No	NE	No	No	Si
	Asociación de tareas a desviaciones (TC ¹⁰ o TA)	No	No	No	NE	No	Si	Si
	Generación semiautomática de TM a riesgos	No	No	No	NE	No	No	Si
	Generación semiautomática de TC a desviaciones	No	No	No	NE	No	No	Si
Fase 5	Generar reportes de riesgos	NE	No	NE	NE	Si	No	No
	Visualización de Gráficos	NE	No	Si	NE	Si	No	Si
NR ¹¹	Formalización de Categorías de Riesgos y Tipos de Incidencias	No	No	No	NE	No	No	Si
	Importar riesgos	NE	No	Si	NE	NE	No	No
	Exportar riesgos	NE	No	NE	NE	NE	No	Si

Tabla 6. Análisis comparativo de la herramienta GESPRO 12.05 respecto a otras herramientas de GP.

Una vez acabado este análisis comparativo que se muestra en la tabla 6, se llega a la conclusión que la herramienta GESPRO implementa la mayoría de las funcionalidades tenidas en cuenta en la comparación para cada una de las fases definidas por el PMBOK, ubicándola en una posición superior al resto de las comparadas. De esta forma se demuestra la mejoría o evolución alcanzada por GESPRO respecto a su versión anterior y a su vez, su alineamiento a la guía del PMBOK respecto al resto de las herramientas analizadas.

⁹ Desviaciones Generadas.

¹⁰ Tarea Correctiva.

¹¹ No está relacionado con ninguna de las fases.

3.4 Análisis del alineamiento de la solución a la Resolución 60/11

El siguiente análisis pretende demostrar el alineamiento de la solución propuesta (módulo de Gestión de Riesgos versión 2.0) con la Resolución No 60/11 establecida por la Contraloría General de la República de Cuba el día 3 de marzo de 2011.

Según lo estudiado en el Epígrafe 1.2.1 del Capítulo 1, la Resolución 60/11 dicta 3 procesos principales: Identificación de riesgos y detección del cambio, Determinación de los objetivos de control, y Prevención de riesgos; los cuales para su mejor estudio, se han dividido en los siguientes:

✓ Proceso 1: Identificación

Se identifican los riesgos y se clasifican internos o externos

✓ Proceso 2: Análisis

Los riesgos son evaluados y cuantificados¹²

✓ Proceso 3: Determinación de los objetivos de control¹³

Se definen las medidas o procedimientos de control a aplicar

✓ Proceso 4: Prevención de riesgos¹⁴

Conjunto de acciones dirigidas a reducir al mínimo posible las condiciones que propician los riesgos internos o externos, así como incidencias disciplinarias e ilegalidades. Se elabora el Plan de Prevención de Riesgos.

Una vez analizadas las fases que dicta la guía del PMBOK para el proceso de Gestión de Riesgos en el epígrafe anterior, y los procesos que establece la Resolución 60/11 de la Contraloría General en el presente epígrafe, podemos entablar similitudes:

Fases que establece el PMBOK	Procesos que dicta la Resolución 60/11
Fase 1: Identificación de los riesgos	Proceso 1: Identificación
Fase 2: Análisis cualitativo	Proceso 2: Análisis
Fase 3: Análisis cuantitativo	
Fase 4: Planeación de la respuesta a los	Proceso 3: Determinación de los objetivos

¹² Según (WordReference, 2012): **Cuantificar**: Expresar numéricamente una magnitud.

¹³ Según (MINJUS, 2011): Resultado que se desea alcanzar. Se realizan acciones de control para evitar manifestaciones negativas.

¹⁴ Según (MINJUS, 2011): Da seguimiento sistemático a los objetivos de control.

Capítulo 3: Análisis de los Resultados

riesgos	de control
Fase 5: Monitoreo y control	Proceso 4: Prevención de riesgos

Tabla 7. Similitud entre las fases del PMBOK y los procesos establecidos por la Resolución 60/11.

Dado que el modulo de Gestión de Riesgos realiza su proceso de administración a través de lo dictado por la guía del PMBOK, se puede afirmar que el mismo se ajusta a los procesos establecidos por la Resolución antes mencionada.

Otro factor interesante de mencionar para este análisis dentro de las disposiciones establecidas por esta Resolución, es el Plan de Prevención de Riesgos, elaborado en el proceso de Prevención de Riesgos, en el cual se identifican los riesgos, posibles manifestaciones negativas, medidas a aplicar, responsable y fecha de cumplimiento de las medidas (MINJUS, 2011).

MODELO DEL PLAN DE PREVENCION DE RIESGOS

No.	ACTIVIDAD O AREA	RIESGOS	POSIBLES MANIFESTACIONES NEGATIVAS	MEDIDAS A APLICAR	RESPONSABLE	EJECUTANTE	FECHA DE CUMPLIMIENTO DE LAS MEDIDAS

Figura 20. Modelo de Plan de Prevención de Riesgos, tomado de (MINJUS, 2011).

En este plan también es posible encontrar similitudes con la manera de administrar los riesgos del módulo del GESPRO 12.05, pues en éste de cada riesgo se almacena:

- Numero identificador.
- Categoría del riesgo.
- Nombre del riesgo.
- Incidencias que el mismo puede generar.
- Tareas de mitigación y/o contingencia.
- Persona que lo identificó.
- Persona a la que se le encargó llevar a cabo las acciones de mitigación y/o contingencia.
- Fecha de cierre.

Estos y el resto de los atributos de cualquier riesgo, es posible visualizarlos, ya sea desde la vista de riesgos del proyecto, como accediendo a la vista de información detallada de los riesgos, al darle clic a su nombre. Por tanto podemos afirmar que los valores almacenados por el módulo de Gestión de Riesgos del sistema GESPRO para cada riesgo identificado se corresponden con los propuestos por el Plan de Prevención de Riesgos establecido por la Resolución 60/11

De esta manera concluimos que existe un alineamiento de la solución desarrollada con la Resolución antes mencionada, dando por cumplido este propósito.

3.5 Validación de las nuevas funcionalidades a través de las pruebas de caja negra

Conociendo la función específica para la que fue diseñado el producto, se pueden diseñar pruebas que demuestren que dicha función está bien realizada. Tales pruebas son llevadas a cabo sobre la interfaz del software, es decir, de la función, actuando sobre ella como una caja negra, proporcionando unas entradas y estudiando las salidas para ver si concuerdan con las esperadas.

A continuación se muestran los resultados obtenidos a partir de la aplicación de pruebas de caja negra a dos de las nuevas funcionalidades con las que cuenta el sistema. El resto de las pruebas aplicadas se encuentran en la carpeta Anexos, en el CD que se adjunta a la investigación:

Funcionalidad 1 Importar riesgos desde un centro hacia un proyecto:

- ✓ **Escenario 1: Importar un riesgo del centro hacia el proyecto que no esté ya en el mismo:**

Capítulo 3: Análisis de los Resultados

Inicio | Mi página | Proyectos | Estado Proyectos | Dominios | Recursos | Riesgos | Administración | Ayuda | Conectado como admin | Mi cuenta | Desconexión

Proyecto Caso estudio para liberación | Búsqueda: | Proyecto Caso estudio para liberación

Vistazo | Planificación | Ejecución | Gestión Logística | Gestión documental | Trabajo Colaborativo | Configuración

Riesgos | Riesgos del Centro | Incidencias | Desviaciones | Estadísticas

Riesgos del Centro

Categoría: Ninguno | Estados: Ninguno | Probabilidad: Ninguno | Impacto: Ninguno | Exposición: Ninguno | Aceptar

#	NOMBRE	CATEGORIA	ESTADO	PROB	IMPACTO	DETEC	EXPO	
16	Riesgo de Prueba	Gestión	Inactivo	Muy bajo	Muy bajo	Muy bajo	0.2	Adicionar al Proyecto
12	Entrega tardía de equipamiento	Relaciones comerciales y legales	Activo	Muy bajo	Muy bajo	Muy bajo	0.2	Adicionar al Proyecto
13	Pérdida de personal en la e...	Relaciones comerciales y legales	Activo	Muy bajo	Muy bajo	Muy bajo	0.2	Adicionar al Proyecto
14	Cambios en política que afe...	Relaciones comerciales y legales	Activo	Muy bajo	Muy bajo	Muy bajo	0.2	Adicionar al Proyecto
3	Fallo en el suministro eléc...	Relaciones comerciales y legales	Activo	Muy bajo	Muy bajo	Muy bajo	0.2	Adicionar al Proyecto

(1-5/5) | Por página: 10, 15, 30

Exportar a: CSV

El usuario da clic en el botón "Adicionar al Proyecto" de un riesgo del Centro que no se encuentra aún dentro de los riesgos del Proyecto.

Paquete de Gestión de Proyectos (GESPRO 11.05), Dirección Técnica, UCI, 2010-2011. Powered by Redmine © 2006-2011, Jean-Philippe Lang

Figura 21. Acción del Usuario.

Inicio | Mi página | Proyectos | Estado Proyectos | Dominios | Recursos | Riesgos | Administración | Ayuda | Conectado como admin | Mi cuenta | Desconexión

Proyecto Caso estudio para liberación | Búsqueda: | Proyecto Caso estudio para liberación

Vistazo | Planificación | Ejecución | Gestión Logística | Gestión documental | Trabajo Colaborativo | Configuración

Riesgos | Riesgos del Centro | Incidencias | Desviaciones | Estadísticas

Riesgos del Centro

Categoría: Ninguno | Estados: Ninguno | Probabilidad: Ninguno | Impacto: Ninguno | Exposición: Ninguno | Aceptar

Riesgo de Prueba será adicionado al proyecto. ¿Está seguro?

Cancel OK

#	NOMBRE	CATEGORIA	ESTADO	PROB	IMPACTO	DETEC	EXPO	
16	Riesgo de Prueba	Gestión	Inactivo	Muy bajo	Muy bajo	Muy bajo	0.2	Adicionar al Proyecto
12	Entrega tardía de equipamiento	Relaciones comerciales y legales	Activo	Muy bajo	Muy bajo	Muy bajo	0.2	Adicionar al Proyecto
13	Pérdida de personal en la e...	Relaciones comerciales y legales	Activo	Muy bajo	Muy bajo	Muy bajo	0.2	Adicionar al Proyecto
14	Cambios en política que afe...	Relaciones comerciales y legales	Activo	Muy bajo	Muy bajo	Muy bajo	0.2	Adicionar al Proyecto
3	Fallo en el suministro eléc...	Relaciones comerciales y legales	Activo	Muy bajo	Muy bajo	Muy bajo	0.2	Adicionar al Proyecto

(1-5/5) | Por página: 10, 15, 30

Exportar a: CSV

Figura 22. Confirmación del Sistema.

Capítulo 3: Análisis de los Resultados

Inicio Mi página Proyectos Estadísticas Dominios Recursos Riesgos Administración Ayuda Conectado como admin Mi cuenta Desconexión

Proyecto Caso estudio para liberación Búsqueda: Proyecto Caso estudio para liberación

Vistazo Planificación Ejecución Gestión Logística Gestión documental Trabajo Colaborativo Configuración

✓ Riesgo adicionado correctamente al proyecto

Riesgos Riesgos del Centro Incidencias Desviaciones Estadísticas

Riesgos del Centro

Categoría Ninguno Estados Ninguno Probabilidad Ninguno Impacto Ninguno Exposición Ninguno Aceptar

#	NOMBRE	CATEGORÍA	ESTADO	PROB	IMPACTO	DETEC	EXPO	
16	Riesgo de Prueba	Gestión	Inactivo	Muy bajo	Muy bajo	Muy bajo	0.2	➕ Adicionar al Proyecto
12	Entrega tardía de equipamiento	Relaciones comerciales y legales	Activo	Muy bajo	Muy bajo	Muy bajo	0.2	➕ Adicionar al Proyecto
13	Pérdida de personal en la e...	Relaciones comerciales y legales	Activo	Muy bajo	Muy bajo	Muy bajo	0.2	➕ Adicionar al Proyecto
14	Cambios en política que afe...	Relaciones comerciales y legales	Activo	Muy bajo	Muy bajo	Muy bajo	0.2	➕ Adicionar al Proyecto
3	Fallo en el suministro eléc...	Relaciones comerciales y legales	Activo	Muy bajo	Muy bajo	Muy bajo	0.2	➕ Adicionar al Proyecto

(1-5/5) | Por página: 10, 15, 30

Exportar a: CSV

Paquete de Gestión de Proyectos (GESPRO 11.05) Dirección Técnica, UCI, 2010-2011. Powered by Redmine © 2005-2011 Jean-Philippe Lang

Figura 23. Respuesta del Sistema

- ✓ Escenario 2: Importar un riesgo del centro hacia el proyecto que esté ya en el mismo

Capítulo 3: Análisis de los Resultados

Inicio | Mi página | Proyectos | Estado Proyectos | Dominios | Recursos | Riesgos | Administración | Ayuda | Conectado como **admin** | Mi cuenta | Desconexión

Proyecto Caso estudio para liberación | Búsqueda: | Proyecto Caso estudio para liberación

Vistazo | **Planificación** | Ejecución | Gestión Logística | Gestión documental | Trabajo Colaborativo | Configuración

Riesgos | **Riesgos del Centro** | Incidencias | Desviaciones | Estadísticas

Riesgos del Centro

Categoría: Ninguno | Estados: Ninguno | Probabilidad: Ninguno | Impacto: Ninguno | Exposición: Ninguno | Aceptar

#	NOMBRE	CATEGORÍA	ESTADO	PROB	IMPACTO	DETEC	EXPO	
16	Riesgo de Prueba	Gestión	Inactivo	Muy bajo	Muy bajo	Muy bajo	0.2	Adicionar al Proyecto
12	Entrega tardía de equipamiento	Relaciones comerciales y legales	Activo	Muy bajo	Muy bajo	Muy bajo	0.2	Adicionar al Proyecto
13	Pérdida de personal en la e...	Relaciones comerciales y legales	Activo	Muy bajo	Muy bajo	Muy bajo	0.2	Adicionar al Proyecto
14	Cambios en política que afe...	Relaciones comerciales y legales	Activo	Muy bajo	Muy bajo	Muy bajo	0.2	Adicionar al Proyecto
3	Fallo en el suministro eléc...	Relaciones comerciales y legales	Activo	Muy bajo	Muy bajo	Muy bajo	0.2	Adicionar al Proyecto

(1-5/5) | Por página: 10, 15, 30

Exportar a: CSV

El usuario da clic en el botón "Adicionar al Proyecto" de un riesgo del Centro que se encuentra ya dentro de los riesgos del Proyecto.

Paquete de Gestión de Proyectos (GESPRO 11.05) | Dirección Técnica, UCI, 2010-2011. Powered by Redmine © 2006-2011 Jean-Philippe Lang

Figura 24. Acción del Usuario

Inicio | Mi página | Proyectos | Estado Proyectos | Dominios | Recursos | Riesgos | Administración | Ayuda | Conectado como **admin** | Mi cuenta | Desconexión

Proyecto Caso estudio para liberación | Búsqueda: | Proyecto Caso estudio para liberación

Vistazo | Planificación | **Ejecución** | Gestión Logística | Gestión documental | Trabajo Colaborativo | Configuración

Riesgos | Riesgos del Centro | Incidencias | Desviaciones | Estadísticas

Riesgos del Centro

Categoría: Ninguno | Estados: Ninguno | Probabilidad: Ninguno | Impacto: Ninguno | Exposición: Ninguno | Aceptar

Riesgo de Prueba será adicionado al proyecto. ¿Está seguro?

Cancel OK

#	NOMBRE	CATEGORÍA	ESTADO	PROB	IMPACTO	DETEC	EXPO	
16	Riesgo de Prueba	Gestión	Inactivo	Muy bajo	Muy bajo	Muy bajo	0.2	Adicionar al Proyecto
12	Entrega tardía de equipamiento	Relaciones comerciales y legales	Activo	Muy bajo	Muy bajo	Muy bajo	0.2	Adicionar al Proyecto
13	Pérdida de personal en la e...	Relaciones comerciales y legales	Activo	Muy bajo	Muy bajo	Muy bajo	0.2	Adicionar al Proyecto
14	Cambios en política que afe...	Relaciones comerciales y legales	Activo	Muy bajo	Muy bajo	Muy bajo	0.2	Adicionar al Proyecto
3	Fallo en el suministro eléc...	Relaciones comerciales y legales	Activo	Muy bajo	Muy bajo	Muy bajo	0.2	Adicionar al Proyecto

(1-5/5) | Por página: 10, 15, 30

Exportar a: CSV

Figura 25. Confirmación del Sistema

Capítulo 3: Análisis de los Resultados

Inicio Mi página Proyectos Estado Proyectos Dominios Recursos Riesgos Administración Ayuda Conectado como admin Mi cuenta Desconexión

Proyecto Caso estudio para liberación Búsqueda: Proyecto Caso estudio para liberación

Vistazo Planificación Ejecución Gestión Logística Gestión documental Trabajo Colaborativo Configuración

Este riesgo ya ha sido añadido a los riesgos del proyecto

Riesgos Riesgos del Centro Incidencias Desviaciones Estadísticas

Riesgos del Centro

Categoría Ninguno Estados Ninguno Probabilidad Ninguno Impacto Ninguno Exposición Ninguno Aceptar

#	NOMBRE	CATEGORÍA	ESTADO	PROB	IMPACTO	DETEC	EXPO	
16	Riesgo de Prueba	Gestión	Inactivo	Muy bajo	Muy bajo	Muy bajo	0.2	Adicionar al Proyecto
12	Entrega tardía de equipamiento	Relaciones comerciales y legales	Activo	Muy bajo	Muy bajo	Muy bajo	0.2	Adicionar al Proyecto
13	Pérdida de personal en la e...	Relaciones comerciales y legales	Activo	Muy bajo	Muy bajo	Muy bajo	0.2	Adicionar al Proyecto
14	Cambios en política que afe...	Relaciones comerciales y legales	Activo	Muy bajo	Muy bajo	Muy bajo	0.2	Adicionar al Proyecto
3	Fallo en el suministro eléc...	Relaciones comerciales y legales	Activo	Muy bajo	Muy bajo	Muy bajo	0.2	Adicionar al Proyecto

(1-5/5) | Por página: 10, 15, 30

Exportar a: CSV

Paquete de Gestión de Proyectos (GESPRO 11.05). Dirección Técnica, UCI, 2010-2011. Powered by Redmine © 2006-2011 Jean-Philippe Lang

Figura 26. Respuesta del Sistema

Funcionalidad 7: Mostrar alertas.

- ✓ Escenario 1 Mostrar alertas

Capítulo 3: Análisis de los Resultados

Inicio Mi página Proyectos Estado Proyectos Dominios Recursos Riesgos Administración Ayuda Conectado como admin Mi cuenta Desconexión

Proyecto Caso estudio para liberación Búsqueda: Proyecto Caso estudio para liberación

Vistazo Planificación Ejecución Gestión Logística Gestión documental Trabajo Colaborativo Configuración

Riesgos Riesgos del Centro Incidencias Desviaciones Estadísticas

Riesgos El usuario visualiza el listado de riesgos del Proyecto

Categoría Ninguno Estados Ninguno Probabilidad Ninguno Impacto Ninguno Exposición Ninguno

Aceptar

#	NOMBRE	ESTADO	PROB	IMPACTO	DETEC	EXPO	TIEMPO PLAN	TIEMPO REAL	TAREAS MIT	TAREAS AFEC		
192	riesgoCentro1	Identificado	Muy bajo	Muy bajo	Muy bajo	0.2	8.0	0.0	1	1	Editar	Eliminar
193	Entrega tardía de equipamiento	Identificado	Muy bajo	Muy bajo	Muy bajo	0.2	0.0	0.0	0	0	Editar	Eliminar
194	Pérdida de personal en la e...	Identificado	Muy bajo	Muy bajo	Muy bajo	0.2	0.0	0.0	0	0	Editar	Eliminar
195	Cambios en política que afe...	Identificado	Muy bajo	Muy bajo	Muy bajo	0.2	0.0	0.0	0	0	Editar	Eliminar
196	Fallo en el suministro eléc...	Identificado	Muy bajo	Muy bajo	Muy bajo	0.2	0.0	0.0	0	0	Editar	Eliminar
197	Riesgo de Prueba	Identificado	Muy bajo	Muy bajo	Muy bajo	0.2	0.0	0.0	0	0	Editar	Eliminar

(1-6/6) | Por página: 10, 15, 30

Nuevo riesgo

Exportar a: CSV

Paquete de Gestión de Proyectos (GESPRO 11.05), Dirección Técnica, UCI, 2010-2011. Powered by Redmine © 2006-2011 Jean-Philippe Lang

Figura 27. Acción del Usuario

Inicio Mi página Proyectos Estado Proyectos Dominios Recursos Riesgos Administración Ayuda Conectado como admin Mi cuenta Desconexión

Proyecto Caso estudio para liberación Búsqueda: Proyecto Caso estudio para liberación

Vistazo Planificación Ejecución Gestión Logística Gestión documental Trabajo Colaborativo Configuración

Riesgos Riesgos del Centro Incidencias Desviaciones Estadísticas

Riesgos

Categoría Ninguno Estados Ninguno Probabilidad Ninguno Impacto Ninguno Exposición Ninguno

Aceptar

#	NOMBRE	ESTADO	PROB	IMPACTO	DETEC	EXPO	TIEMPO PLAN	TIEMPO REAL	TAREAS MIT	TAREAS AFEC		
194	Pérdida de personal en la e...	Identificado	Muy alto	Muy alto	Muy alto	1.0	0.0	0.0	0	0	Editar	Eliminar
193	Entrega tardía de equipamiento	Identificado	Muy alto	Muy alto	Muy alto	1.0	0.0	0.0	0	0	Editar	Eliminar
195	Cambios en política que afe...	Identificado	Medio	Medio	Medio	0.6	0.0	0.0	0	0	Editar	Eliminar
197	Riesgo de Prueba	Identificado	Muy bajo	Muy bajo	Muy bajo	0.2	0.0	0.0	0	0	Editar	Eliminar
196	Fallo en el suministro eléc...	Identificado	Muy bajo	Muy bajo	Muy bajo	0.2	0.0	0.0	0	0	Editar	Eliminar
192	riesgoCentro1	Identificado	Muy bajo	Muy bajo	Muy bajo	0.2	8.0	0.0	1	1	Editar	Eliminar

(1-6/6) | Por página: 10, 15, 30

Nuevo riesgo

El sistema muestra de manera correcta las alertas para los parámetros "Exposición", "Tiempo Planificado" y "Tiempo Real"

Exportar a: CSV

Paquete de Gestión de Proyectos (GESPRO 11.05), Dirección Técnica, UCI, 2010-2011. Powered by Redmine © 2006-2011 Jean-Philippe Lang

Figura 28. Respuesta del Sistema

3.5 Conclusiones Parciales

En el presente capítulo se realizó un análisis comparativo de la herramienta GESPRO con otras herramientas de GP en cuanto a los procesos de GR, una vez aplicadas las nuevas funcionalidades implementadas para el mismo, arribando a la conclusión de que el módulo de Gestión de Riesgos eleva la calidad de la administración de los riesgos en el sistema, haciéndolo competitivo con otras herramientas respecto a las fases de identificación, análisis cualitativo, respuesta a los riesgos y monitoreo y control, pues implementa la mayoría de las funcionalidades tenidas en cuenta en la comparación para cada fase. Además se exponen algunos de los resultados obtenidos tras la aplicación de pruebas de caja negra al sistema, arribando a la conclusión de que se ha logrado una exitosa implementación, pues las respuestas obtenidas por las funcionalidades probadas son las esperadas.

Conclusiones

Con la realización de este trabajo se logró mejorar el proceso de administración de los riesgos en GESPRO en las áreas de: identificación de los riesgos, análisis cualitativo, planificación y respuesta a los riesgos, y medición y control de los mismos. Se logró la creación de un módulo que cumple con las normas de diseño e implementación y con las disposiciones establecidas por la Contraloría General de la República en la Resolución No 60/11.

Para lograr este resultado final se cumplieron los objetivos específicos trazados:

- Se definió el marco teórico de la investigación el cual ayudó a la comprensión y desarrollo del sistema así como la elección de las herramientas usadas en el diseño e implementación.
- Se definieron e implementaron las nuevas funcionalidades, cumpliendo con los patrones arquitectónicos y de diseño requeridos, logrando un módulo funcional para el sistema.
- Se validaron las nuevas funcionalidades a través de pruebas de caja negra, y se diseñaron casos de prueba que serán útiles a probadores para detectar funciones incorrectas
- Se elaboró un análisis comparativo de la herramienta GESPRO con otras herramientas de GP en cuanto a GR, comprobando la ejecución de un correcto y mejorado proceso de GR de la misma con respecto a las fases anteriormente mencionadas que establece el PMBOK.

Con lo que se le dio cumplimiento a las tareas de investigación propuestas. Concluyendo de esta manera que el trabajo realizado proporciona una mejor administración de los riesgos al sistema GESPRO.

Recomendaciones

El análisis cuantitativo de riesgos dentro de la GR es una de las áreas menos tratadas por su complejidad. Por ello, es recomendable realizar un modelo para el análisis cuantitativo de riesgos basado en técnicas de *softcomputing* de modo que robustezca el módulo de Gestión de Riesgos en el GESPRO.

Referencias Bibliográficas

Atlassian. 2012. JIRA- Track bugs, tasks, and projects for software development | Atlassian. [En línea] 2012. [Citado el: 7 de Mayo de 2012.] <http://www.atlassian.com/software/jira/overview>.

Boehm, B. W. 1989. *Software Risk Management*. s.l. : ESEC '89, Lecture Notes in Computer, 1989.

—. **1991.** Software risk management: principles and practices [Electronic Version]. [En línea] 1991. [Citado el: 11 de Junio de 2012.] http://www.inf.ed.ac.uk/teaching/courses/seoc/2006_2007/resources/PM_SRM.pdf.

Caballero, Omar Higinio. 2006. *Tecnologías de Información y Herramientas para la Administración de Proyectos de Software*. Monterrey : Instituto Tecnológico de Estudios Superiores de Monterrey, 2006.

Calero, Manuel. 2012. *Una explicación de la programación extrema (XP)*. 2012.

CaseySoftware. 2012. New dotProject Module: Risk Management | CaseySoftware. LLC - Supporters & Developers of web2project. [En línea] 2012. [Citado el: 5 de Junio de 2012.] <http://caseyssoftware.com/blog/new-dotproject-module-risk-management#comment-285>.

EcuRed. 2012. PostgreSQL. [En línea] 2012. [Citado el: 12 de Junio de 2012.] <http://www.ecured.cu/index.php/PostgreSQL>.

García, Eduardo. 2008. *Propuesta de modelo para la Gestión de Riesgos en los proyectos de producción de software*. La Habana, Cuba : Universidad de las Ciencias Informáticas, 2008.

Higuera, R. P. y Haimés, Y. Y. 1996. *Software risk management*. Carnegie Mellon University, EE.UU : Technical Report, Software Engineering Institute, 1996.

Itescam. 2012. Herramientas CASE. [En línea] 2012. [Citado el: 11 de Junio de 2012.] www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r19670.DOC..

Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000. *El Proceso Unificado de Desarrollo de Software*. España : Pearson Educación, S.A., 2000.

Márquez, José Manuel. 2008. ¿Qué es Ingeniería de Requisitos (IR)? [En línea] 2008. [Citado el: 14 de Marzo de 2012.] <http://danielvn7.wordpress.com/2008/03/27/%C2%BFque-es-ingenieria-de-requisitos-ir/>.

Matsuo, E. K. 1999. *Risk assessment in incremental software development*. Monterrey, California : Naval Postgraduate School, 1999.

Módulo de Gestión de Riesgos versión 2.0

para el sistema GESPRO 12.05

MINJUS. 2011. *Resolución No. 60/11.* La Habana, Cuba : Contraloría General de la República de Cuba, Ministerio de Justicia, 2011.

MSDN. 2012. Administrar riesgos. [En línea] 2012. [Citado el: 3 de Junio de 2012.] <http://msdn.microsoft.com/es-es/library/ee461564.aspx..>

Murphy, R. L., y otros. 2008. Continuous Risk Management Guidebook. [En línea] Software Engineering Institute, Carnegie Mellon University, EE.UU, 2008. [Citado el: 11 de Junio de 2012.] <http://www.sei.cmu.edu/publications/books/other-books/crm.guidebk.html>.

Netbeans. 2012. Netbeans. [En línea] 2012. [Citado el: 12 de Junio de 2012.] http://netbeans.org/community/releases/69/index_es.html.

Palacio, Juan y Ruata, Claudia. 2009. *Scrum Manager Gestión de Proyectos.* 2009.

Passador, Victor. 2010. Nuevos conceptos en TFS. *VEMN Team Blog.* [En línea] 2010. [Citado el: 16 de Febrero de 2012.] <http://www.vemn.com.ar/Blog/?tag=/Team-Foundation-Server&page=3>.

Piñero, Pedro. 2010. *Conferencia 1: Contratación, Factibilidad de Proyectos y Negociación.* s.l. : Universidad de las Ciencias Informáticas, 2010.

—. **2010.** *GESPRO 11.05 Un Sistema para la Dirección Integrada de Proyectos para la Gestión de la Producción.* La Habana : Universidad de las Ciencias Informáticas, 2010.

—. **2009.** *Metodologías ágiles y formales o robustas.* La Habana, Cuba : Universidad de las Ciencias Informáticas, 2009.

PMI. 2004. *Guía de los Fundamentos de la Dirección de Proyectos.* Newtown Square, Pennsylvania, EE.UU : Project Management Institute, Inc., 2004.

Pressman, R. S. 2005. *Ingeniería de Software. Un enfoque práctico (Quinta Edición).* La Habana, Cuba : Editorial Félix Valera, 2005.

Rivera, Sergio Michel. 2010. *Modelo de un Sistema de Razonamiento basado en Casos para el análisis en la Gestión de Riesgos.* La Habana, Cuba : Universidad de las Ciencias Informáticas, 2010.

Rosenberg, L. H., Hammer, T. y Gallo, A. 2000. CrossTalk, The Journal of Defense Software Engineering. *Continuous Risk Management at NASA.* [En línea] 2000. [Citado el: 3 de Mayo de 2012.] <http://www.stsc.hill.af.mil/Crosstalk/2000/02/rosenberg.html..>

Ruby, Sam, Thomas, Dave y Heinemeier, David. 2009. *Agile Web Development with Rails.* EE.UU : s.n., 2009.

Santamaría, Fernando. 2012. Una introducción a los ecosistemas digitales | Fernando Santamaría. [En línea] 2012. [Citado el: 14 de Junio de 2012.] <http://fernandosantamaria.com/blog/2010/07/una-introduccion-a-los-ecosistemas->

Módulo de Gestión de Riesgos versión 2.0 para el sistema GESPRO 12.05

digitales/.

TodoProgramas. 2012. Visual Paradigm. [En línea] 2012. [Citado el: 11 de Junio de 2012.] <http://www.todoprogramas.com/programalinux/visualparadigmforum/>.

Ubuntu. 2012. PgAdmin. [En línea] 2012. [Citado el: 12 de Junio de 2012.] http://www.guia-ubuntu.org/index.php?title=PgAdmin_III.

Williams, R. C., Pandelios, G. J. y Behrens, S. G. 1999. *Software risk evaluation method description (Version 2.0)*. EE.UU : Software Engineering Institute, Carnegie Mellon University, 1999.

WordReference. 2012. WordReference.com. [En línea] 2012. <http://www.wordreference.com>

Zulueta, Y. 2007. *Modelo de gestión de riesgos en proyectos de desarrollo de software*. La Habana, Cuba : Universidad de las Ciencias Informáticas, 2007.