



Universidad de las Ciencias Informáticas

Facultad 4

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

Sistema para la gestión de riesgos en la Facultad 4

Autoras:

Mislaidy Padín Gutierrez

Ailen Lores Delgado

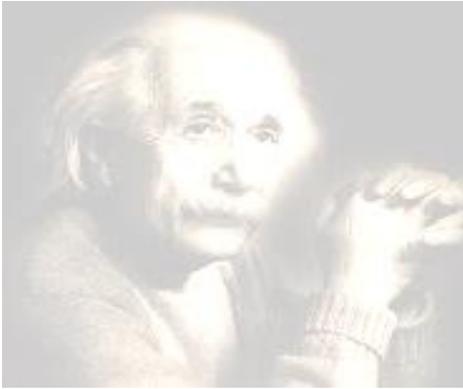
Tutores:

Ing. Yasirys Terry González

Ing. Leonardo Rodríguez González

La Habana, junio 2015

“Año 57 de la Revolución”



"La vida es muy peligrosa. No por las personas que hacen el mal, sino por las que se sientan a ver lo que pasa."

Albert Einstein

"A veces podemos pasarnos años sin vivir en absoluto, y de pronto toda nuestra vida se concentra en un solo instante."

Oscar Wilde



"He sido un hombre afortunado en la vida: nada me fue fácil."

Sigmund Freud

Declaración de autoría

Declaramos ser autoras de este trabajo de diploma y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes _____ del año 2015.

Firma del autor

Mislaidy Padín Gutierrez

Firma del autor

Ailen Lores Delgado

Firma del tutor

Ing. Yasirys Terry González

Firma del tutor

Ing. Leonardo Rodríguez González

Dedicatoria

De Ailen Lores Delgado

Con todo mi cariño y amor dedico este trabajo a la persona más importante que tengo en este mundo, mi mamá. Ella siempre me ha apoyado en todas las decisiones que he tomado en esta vida ya sean buenas o malas.

Gracias mima por darme la mano cuando no veía la salida del camino, por estar siempre presente cuando te he necesitado, gracias por ser un ejemplo para mí, gracias por decirme adelante, tú si puedes por todo esto y más, mil gracias.

Te amo mamita

De Mislaidy Padín Gutierrez

A mis padres María y Lázaro por el amor y la educación que me han dado, por el apoyo incondicional a lo largo de toda mi vida y por ser simplemente, los mejores padres del mundo. Los amo.

A mi hermana Misleydis por estar siempre a mi lado, por ser un ejemplo a seguir, por ser esa luz que ilumina e iluminará mi vida. Te amo tata.

Agradecimientos

De Ailen Iores Delgado

Son muchas las personas a las cuales tengo que agradecer por haberme ayudado en el transcurso de mi vida y de mi carrera, primero que todo quisiera agradecerle a la persona más importante de mi vida, mi madre, esa luchadora incansable que daría la vida por mí y por mi hermano.

A mi hermano Karel mi hermanito del alma, por cuidar a mi mami mientras estoy lejos de ellos, por ser un hermano ejemplar, por apoyarme en todo, por ser tan cariñoso conmigo.

A mi novio Manuel que desde que nos conocimos en 2do año me ha brindado todo su apoyo, cariño, amor y dedicación. Gracias por estar ahí, y hacerme sentir que no estaba sola. Gracias por tus consejos, por hacerme reír, llorar y sobre todo ver la vida de otra forma. Por escucharme y no criticar mis opiniones, por apoyarme con mi tesis, y hacerme ser una persona más fuerte.

A mis abuelos del alma Enicia y mi abuelito Marcos que sé que desde donde quiera que esté, va a estar muy orgulloso de mí.

A toda mi familia por el apoyo brindado en todo momento y por confiar en mí, a mis tíos, Marcolina, Ruberth, Ramoncito y mi tiita Cecilia, estos siempre han estado apoyando a mi mamá para que todo me salga bien en mi carrera.

A mis suegros Ernesto y Graciela, por todo su apoyo, por exigirme tanto para que me hiciera una profesional, por tratarme como una hija más, por darme buenos consejos, por siempre estar pendiente de mí, los amo.

A Diannelis que además de ser mi prima la tengo como la hermana que nunca tuve, gracias primi por contar contigo para todo lo que he necesitado.

A mi cuñado Ernestico, que desde que lo conocí ha demostrado ser una persona incondicional conmigo, gracias por ayudarme en todo lo que ha estado a su alcance, gracias por hacerme parte de tu familia y por confiar en mí.

A mi compañera de tesis y mi amiga, Mislaidy, por todo su apoyo durante la carrera y durante la realización de este trabajo de diploma.

A mis tutores Yasirys y Leonardo porque han sido el eslabón más importante en este proceso de desarrollo como profesional. Por su dedicación y atención hacia nosotras.

A mis profesores de la universidad que hicieron posible que yo creciera como profesional, en especial a, Maritza, Sandra y María Caridad.

A mis compañeros de aula que tanto me ayudaron en el transcurso de estos años, Gino, Albertico, Yaniel, Jessica y Pedrito.

A mi amigo Victor, ese que no me abandona nunca, que siempre está presente cuando necesito un consejo, gracias por ser mi amigo del alma, por respetar mis decisiones aunque no esté de acuerdo con ellas.

A mis amistades, Yudita, Yayi, Lisbeth, Gloria, Bety, gracias por su apoyo y su confianza.

A mis amistades de la novia, Raúl, Nelis, Ingrit Liuba y Maribel.

No quisiera terminar sin antes agradecer a una persona muy especial para mí, que sé que aunque lejos te encuentres, espero que cuando te enteres que soy ingeniera, estés muy orgulloso de mí. Te quiero mucho papá.

A todos muchísimas gracias

De Mislaidy Padín Gutierrez

A mi mamá, por todo su esfuerzo y sacrificio, por su preocupación, por la educación que me brindó, su amor incondicional, por todos los ratos malos que pasó por mi culpa, gracias por todo, gracias por existir.

A mi padre que siempre me ha apoyado en todo, por demostrarme que padre no es cualquiera, por su esfuerzo y sacrificio, por estar siempre pendiente de mí, por ser el mejor padre del mundo, por saber escucharme, comprenderme, muchas gracias por tu amor, muchas gracias por ser mi papá.

A mi otra mamá, mi hermana querida, por estar presente en cada etapa de mi vida, en las más importantes, por demostrarme lo fuerte que podemos ser cuando se quiere a alguien, lo mucho que se puede luchar para obtener una victoria, gracias por nunca rendirte, por siempre sonreír, aun con lágrimas en los ojos, gracias por ser el espejo donde tengo que mirarme, gracias por darme el regalo más lindo de la vida, que son los dos loquitos míos. Gracias por ser mi ángel, gracias por ser mi tata.

A mis loquitos, mis sobrinos Mily y Geo por llenarme de alegrías por hacerme sentir un amor que nunca había sentido, por sus sonrisas, por sus ocurrencias, por sus pesadeces, por ser mis niños lindos.

A mi tío Fidel que esté donde esté sé que está orgulloso por este logro alcanzado, por estar pendiente de mi carrera hasta donde pudo, por todo lo que aprendí de él, porque su recuerdo me hace querer ser mejor cada día.

A mi cuñado Alfredo por ayudarme a salir adelante, por cuidar tan bien de mi hermana y por llegar para formar parte de nuestra pequeña familia.

A mi abuela Margarita por incluirme en sus nietos, por cuidar a mi papá, por sus buenos consejos, por darme tanto amor a lo largo de mi vida, por ser la abuela que nunca tuve.

A mi novio José Antonio por estar conmigo día tras día y obligarme a estudiar, por todo su amor y dedicación, muchas gracias mi vida.

A Marlon por toda su dedicación, paciencia y entrega, por todos los ratos buenos y malos que pasé a su lado, por ser mi guía, muchas gracias.

A Chavelo y Maylin por haberme acogido en su familia, por todo el cariño, la preocupación, los buenos momentos, muchas gracias.

A Dayi, Danne y a Yoandri, por todos los momentos que compartimos, porque me hicieron parte del piquete.

A mis amigas Yailín, Mari Carmen y Miléisis por ser mis hermanitas, por estar siempre pendientes de mí, porque la distancia no ha podido romper este gran cariño que sentimos. Las quiero gordis.

A los locos de Ale, Iván y Abelito que de una forma u otra siempre han estado presentes en momentos importantes en mi vida y en la de mis amigas.

Al piquetón de la escuela, Malidia, Carlos, Yasmani, Lijandy, Manuel Reina, los chicos que no se pierden una.

A mi compañera de tesis, que más que compañera es una amiga, una hermana, por compartir tantos momentos buenos, tantos malos, por sus alteraciones que me hacen reír, por su locura, por ser la persona con la que puedo contar para todo. Gracias amiguis.

A mi tutora Yasirys por su exigencia, su preocupación, sus consejos, por saber guiarme durante todo el proceso de la tesis, por ser profesora, tutora y amiga.

A mi tutor, mi amigo Leo por cargar conmigo desde primer año, por toda su ayuda a lo largo de este camino.

A mis compañeros de aula, los que están en la escuela y los que no, a mis buenos amigos, Gino, Claudio y Pedro.

A todos los profesores que me dieron clases, por enseñarme a crecer, por convertirme en una profesional, a todos los profesores de la Facultad, que nunca perdieron su fe en mí, que me dieron miles de oportunidades para lograr este sueño.

A la profesora Yinimary por estar siempre pendiente de la salud de mi hermana, por su preocupación, por su cariño, muchas gracias.

A mis compañeros de la FEU, Lester, Yadiel, Sandy, Malidia, Carlos, Ulises por todos los malos ratos y los buenos, por ayudarme y entenderme siempre, por su buen trabajo con la Facultad.

Al equipo de Somos4, Carlos Ariel, Reimel, Yoelkis, Yordanis, Alejo, por esos momentos mágicos, por esas risas hermosas y sinceras que siempre habían en cada una de las grabaciones, gracias a todos por enseñarme a pararme frente a una cámara sin temblar, me llevo muy buenos recuerdos, malos no hay, gracias por esa experiencia única.

A Geniel, Arturo y Eduardo a quien molesté varias veces para que me ayudara y nunca dijeron no.

A todas aquellas personas que he conocido aquí en la UCI, a todos los que han formado parte de estos años de estudio y sacrificio.

A todos ellos muchas gracias...

Resumen

Este trabajo aborda el desarrollo de un sistema para informatizar el proceso de gestión de riesgos en la Facultad 4 de la Universidad de las Ciencias Informáticas, que contribuya a la toma de decisiones.

El estudio del problema científico posibilitó a las autoras identificar las limitaciones que genera el trabajo sobre la gestión de riesgos mediante un amplio cúmulo de documentación impresa que se requiere elaborar, revisar y conservar por los actores del área implicados en el proceso.

Desde ese punto de partida la investigación se orienta hacia la propuesta de un sistema informático que permita gestionar con mejor eficiencia los riesgos en el área de la universidad seleccionada, utilizando tecnologías libres como principio de soberanía tecnológica al que aspira Cuba.

El desarrollo del sistema fue guiado por la metodología de desarrollo Programación Extrema. Como lenguaje de programación se utilizó PHP en su versión 5.4.12, como *Framework* del lado del cliente se utilizaron JQuery 1.10 y Bootstrap 3.3. Como *framework* del lado del servidor se utilizó PHPUnit 4.1.0 y *Symfony* 2.6.5. Se utilizó el *NetBeans* 8.0 como Entorno de Desarrollo Integrado, Apache 2.4.4 como servidor *web* y MySQL 5.6.12 como gestor de bases de datos. Las autoras realizaron pruebas de software a la propuesta de solución, las cuales aseguran en gran medida la correcta implementación de las funcionalidades y un alto grado de satisfacción por parte del cliente.

Palabras Clave: gestión de riesgo, toma de decisiones, informatizar

Índice

Introducción	1
Capítulo 1: Fundamentación teórica.....	5
1.1. Introducción.....	5
1.2. Gestión de riesgos.....	8
1.3. Descripción de la gestión de riesgos.....	9
1.4. Toma de decisiones.....	10
1.5. Metodología de desarrollo de software	11
1.6. Herramienta CASE	15
1.7. Herramientas y tecnologías para el desarrollo de software.....	16
1.7.1 Lenguajes de programación.....	16
1.7.2 Framework de desarrollo.....	19
1.7.3 Entorno de desarrollo integrado	21
1.7.4 Sistema Gestor de Base de Datos	23
1.7.5 Servidor web	24
1.7.6 Resumen de metodología, herramientas y tecnologías de desarrollo a utilizar.....	25
1.8. Sistemas similares.....	26
Conclusiones.....	28
Capítulo 2: Descripción de la propuesta de solución.....	29
2.1 Introducción.....	29
2.2 Modelo conceptual de negocio	29
2.3 Especificación de los requisitos de software	31
2.4 Roles del sistema	39
2.5 Exploración.....	39
2.5.1 Historias de usuario	40
2.6 Planificación	45
2.6.2 Plan de entregas.....	47
2.6.3 Plan de iteraciones	49
Conclusiones.....	50
Capítulo 3 Validación de la propuesta	51
3.1 Fase de diseño	51
3.2 Arquitectura	51
3.3 Patrón de diseño.....	53
3.3.1 Patrones GRASP	53
3.3.2 Patrones GOF.....	54

3.4	Tarjetas CRC.....	54
3.4.1	Diagrama Entidad Relación.....	59
3.5	Tareas de ingeniería.....	59
3.6	Pruebas.....	61
3.6.1	Pruebas unitarias.....	61
3.6.2	Pruebas de aceptación:.....	63
3.7	Análisis de los resultados de las pruebas.....	69
	Conclusiones.....	70
	Conclusiones Generales.....	71
	Recomendaciones.....	72
	Referencias Bibliográficas.....	73
	Bibliografía.....	78
	Anexos.....	78

Introducción

En la actualidad las Tecnologías de Información y Comunicación (TIC) están transformando la sociedad y han propiciado que las organizaciones enfaticen en crear productos y ofrecer servicios profesionales con calidad, para satisfacer las necesidades y expectativas de los clientes y usuarios. Además, han surgido centros de formación especializados que ofrecen superación, entrenamientos, capacitaciones y otros espacios formativos relacionados con el conocimiento de las TIC. Estas tecnologías pueden ser instrumentos transversales en las diferentes sociedades, penetran, se integran en variadas actividades y pautan el tiempo, la manera de trabajar, de aprender y comunicarse. (Mela, 2011)

Cuba apuesta actualmente a la informatización de la sociedad en el contexto de un escenario complejo pero clarificado por la necesidad de elevar la soberanía tecnológica en el desarrollo de la infraestructura de telecomunicaciones potenciando proyectos de inversiones en la industria de la informática y comunicaciones, preservando lo alcanzado para obtener niveles superiores en las exportaciones y los servicios internos. Uno de los principales logros desde la pasada década ha sido la creación y funcionamiento de la Universidad de las Ciencias Informáticas (UCI), como principal centro de educación superior del país para la formación de profesionales de esa rama del saber destinados a lograr un importante aporte a la economía del país.

En cada una de las Facultades de la UCI se realiza la aplicación de la guía de autocontrol, regida por la Resolución 60 de la Contraloría General de la República de Cuba. Esta guía está integrada por cinco componentes interrelacionados entre sí, en el marco de los principios básicos y las características generales; que son los siguientes: Ambiente de Control, Gestión y Prevención de Riesgos, Actividades de Control, Información y Comunicación, y Supervisión y Monitoreo, los cuales se encuentran estructurados en normas.

La gestión y prevención de riesgos constituye un componente que se ha ido trabajando para lograr los objetivos propuestos en la UCI. El componente en el que se centra esta investigación es el de Gestión y Prevención de Riesgos, el cual establece las bases para la identificación y análisis de los riesgos que enfrentan las organizaciones para alcanzar sus objetivos. (Gaceta, 2011.)

La Facultad 4 de la UCI desarrolla como procesos sustantivos: la formación de profesionales de las Ciencias Informáticas, la investigación, extensión universitaria y la producción; además desarrolla otros procesos administrativos que apoyan los antes mencionados. La totalidad de

Sistema para la gestión de riesgos en la Facultad 4

los procesos está amenazada constantemente por riesgos de diversos tipos, por lo que gestionarlos constituye una prioridad.

En la Facultad 4 se observa la contradicción entre la necesidad de actualización de los planes de riesgos a través de la comunicación y el intercambio de información de los recursos humanos implicados en el proceso y la insuficiente eficiencia en el procesamiento por parte de los responsables de analizar y procesar los datos incluidos en los planes de riesgo.

Actualmente, la gestión de riesgos de la Facultad y de cada una de las áreas (decanato y dependencias, vicedecanatos, departamentos y centro de desarrollo y dependencias) está documentada en archivos que cuentan con varias versiones, pero la recuperación de estos datos en este formato manual dificulta la agilidad de la recuperación de la información que se requiere. A través de la aplicación de la entrevista realizada a los responsables de las áreas de la facultad, se constató que no es suficiente la seguridad de la información, debido a que cualquier persona con acceso a las áreas donde se almacenan estos documentos puede sustraerlos sin permiso del personal encargado; por otra parte esta información no llega a su destino en tiempo, pues las personas responsabilizadas con la toma de decisiones deben obtenerla por varias vías: personalmente, por correo electrónico o por teléfono.

El seguimiento de las acciones que se proponen para la mitigación o eliminación de los riesgos se hace con despachos continuos y personalizados. Con el paso del tiempo muchas de las evidencias de las actividades desarrolladas dentro del proceso de gestión de riesgo se pierden, lo que genera un gran cúmulo de información en forma de texto, que unido a los problemas anteriormente descritos, dificultan la toma de decisiones con respecto a los riesgos.

A partir de lo fundamentado, se define el siguiente **problema a resolver**: ¿Cómo gestionar el proceso de los riesgos en la Facultad 4 para contribuir a la toma de decisiones?

Lo anteriormente lleva a las autoras a declarar como **objeto de estudio**: el proceso de gestión de riesgos.

En consecuencia, el **objetivo general** de la investigación es: Desarrollar un sistema para informatizar el proceso de gestión de riesgos en la Facultad 4, que contribuya a la toma de decisiones con respecto a los riesgos.

En este contexto se define como **campo de acción**: Informatización del proceso de gestión de riesgos en la Facultad 4.

A partir del objetivo general declarado se derivan los siguientes **objetivos específicos**:

Sistema para la gestión de riesgos en la Facultad 4

1. Fundamentar los presupuestos teóricos de la gestión de riesgos.
2. Diseñar el sistema para la gestión de los riesgos en la Facultad 4.
3. Implementar el sistema para la gestión de los riesgos en la Facultad 4
4. Validar el sistema implementado a través de pruebas de software.

El proceso de la investigación científica conduce a las autoras a declarar como **hipótesis** la siguiente: el desarrollo de un sistema para informatizar el proceso de gestión de riesgos en la Facultad 4, contribuirá a la toma de decisiones con respecto a los riesgos.

Para lograr el cumplimiento de los objetivos se proponen las siguientes **tareas de investigación**

- Caracterización de los sistemas para la gestión de riesgos.
- Entrevistas a los directivos y trabajadores de la Facultad 4 que están involucrados en la gestión de los riesgos.
- Análisis comparativo de las características y funcionalidades de soluciones similares.
- Selección de las herramientas y tecnologías a utilizar en el desarrollo del sistema para la gestión de riesgos en la Facultad 4.
- Diseño del sistema para la gestión de riesgos en la Facultad 4.
- Diseño de la base de datos para el sistema a desarrollar.
- Modelado de los artefactos definidos en la metodología seleccionada.
- Diseño de casos de prueba.
- Implementación del sistema para la gestión de riesgos en la Facultad 4.
- Realizar pruebas de caja negra al sistema implementado para validarlo.

Para el desarrollo de las tareas de investigación se combinan diferentes métodos teóricos y empíricos en la exploración y procesamiento de la información.

Los métodos teóricos permiten estudiar las características del objeto de estudio que no son observables directamente, facilitan la construcción de modelos y crean las condiciones para ir más allá de las características superficiales de la realidad, contribuyendo al desarrollo de las teorías científicas. (Sampieri, 2006) Entre los utilizados en el presente trabajo se encuentran:

Métodos teóricos:

Histórico-Lógico: permitió estudiar y analizar los antecedentes y las tendencias actuales referidas a la gestión de riesgos de la Facultad 4, además de los conceptos, procesos y terminologías propios, que contribuyen a la comprensión del objeto de estudio.

Analítico-Sintético: para estudiar en partes el contexto y la problemática existente, la documentación, las características de las metodologías, herramientas y tecnologías, propiciando a las autoras asumir la propuesta de cuáles son las más adecuadas para el desarrollo del sistema que se aporta en la investigación.

Método empírico

Entrevista: se aplicó una entrevista de tipo abierta a los responsables de cada una de las áreas de la Facultad 4 para identificar las necesidades reales que existían en cuanto al tema de Gestión de riesgos, y poder obtener las funcionalidades que propone el sistema.

El presente trabajo está estructurado por los siguientes capítulos:

- **Capítulo 1 Fundamentación teórica:** en este capítulo se abordan los principales presupuestos teóricos del objeto de estudio de la investigación. Se describen elementos fundamentales sobre los riesgos en procesos generales, como son, su declaración y su clasificación. Se analizan también algunos sistemas para la gestión de riesgos y se determinan las herramientas y las metodologías a utilizar en el desarrollo del sistema.
- **Capítulo 2 Descripción de la propuesta de solución:** este capítulo abarca las dos primeras fases de la metodología de desarrollo Programación Extrema: Exploración y Planificación. En la primera fase se definen las Historias de usuario, el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas utilizadas en la implementación del sistema. Durante la Planificación se realiza el Plan de Iteraciones y el Plan de Entregas del producto, además, se establecen las prioridades de cada Historia de usuario y se realiza una estimación del esfuerzo para implementarlas.
- **Capítulo 3 Validación de la propuesta:** contiene una descripción de la arquitectura y de los patrones de diseño, utilizados para implementar el sistema. Además, se determinarán las tarjetas Clase-Responsabilidades-Colaboradores y el modelo de datos a través de un Diagrama Entidad Relación. También se muestran las tareas de ingeniería necesarias para llevar a cabo el proceso de desarrollo. Por otra parte se define la estrategia de pruebas a utilizar, y se describen los casos de pruebas para realizar las pruebas de aceptación o caja negra junto al cliente.

Capítulo 1: Fundamentación teórica

1.1. Introducción

En el presente capítulo se plantean los principales conceptos y términos abordados en la investigación, necesarios para cumplir el objetivo fundamental del trabajo, profundizando en conceptos como riesgos, gestión de riesgos, entre otros. Se analizan además sistemas desarrollados para la gestión de riesgos.

El riesgo constituye una probabilidad de daño o condición de vulnerabilidad. No implica necesariamente daño, no otorga certezas, ni relaciones directas y deterministas, sino una probabilidad de ocurrencia de este. El punto de vista de los riesgos negativos es poder anticiparse al daño, y centrarse en la prevención del mismo, por lo que es indispensable que se realice una buena distinción entre riesgo y daño. Con el transitar de los años se han llevado a cabo disímiles estudios acerca de este tema por lo que han surgido cuantiosos conceptos sobre lo que en exactitud es un riesgo. A continuación se citan definiciones expuestas por varios autores:

- El riesgo es la probabilidad de que una amenaza se convierta en un desastre. La vulnerabilidad o las amenazas, por separado, no representan un peligro. Pero si se juntan, se convierten en un riesgo, o sea, es la probabilidad de que ocurra un desastre. Sin embargo, los riesgos se pueden reducir o manejar. Si se es cuidadoso en las relaciones con el ambiente, estando conscientes de las debilidades y vulnerabilidades frente a las amenazas existentes, se pueden tomar medidas para asegurarse de que las amenazas no se conviertan en desastres. (Cano, 2008)
- Un riesgo es cualquier suceso cuya aparición no se puede determinar a priori y que pueda influir negativamente en el devenir del proyecto. Este está asociado a cualquier actividad que se realice en el proyecto y que imponga una decisión entre varias opciones, ya que siempre habrá un riesgo a equivocarse en la decisión tomada. Por tanto, irá acompañado de todo cambio o decisión que se produzca en el proyecto ya que estas siempre representan un marco de incertidumbre ante lo que puede ocurrir. (García, 2006)
- Riesgo es la posibilidad de sufrir daño o pérdidas. (Alberts, 2006)

De la definición de riesgo, además de los ya expresados hay muchas acepciones y diferentes usos de este término, defendidas por varios autores, y visto desde diversos puntos de vista.

Una de las conclusiones que se determinaron a partir del estudio realizado acerca de la definición de riesgo es el alcance internacional del término. En los conceptos anteriormente citados en el documento, se asume el riesgo como una amenaza que representa un peligro cuya aparición no se puede determinar y que influyen negativamente en las decisiones tomadas.

En la ley 60 de la Gaceta Oficial de la República de Cuba, Resolución No. 60/11 de la Contraloría General de la República, en el Capítulo 3, se define un riesgo como: la incertidumbre de que ocurra un acontecimiento que pudiera afectar o beneficiar el logro de los objetivos y metas de la organización. El riesgo se puede medir en términos de consecuencias favorables o no y de probabilidad de ocurrencia. (Gaceta, 2011.)

Las autoras del trabajo, comparten la definición que propone la Ley 60 de la Gaceta Oficial de la República de Cuba ya que la consideran más abarcadora y completa, al aportar una definición que permite aclarar todos los puntos de vista de un riesgo. Además es la que se maneja en todas las instituciones del país y por tanto en la Facultad 4 de la UCI.

Tipos de riesgos

No existe una clasificación oficial de los riesgos; estos suelen ser clasificados de diversas formas y en función de diferentes parámetros.

- Parámetros de vulnerabilidad: la capacidad de los riesgos de afectar o no a grandes colectivos.
- Parámetros temporales: si los efectos de los riesgos son o no inmediatos.
- Parámetros socioeconómicos y medioambientales: origen de los riesgos.

(sostenible, 2010)

Parámetro de vulnerabilidad

Esta clasificación separa aquellos que pueden afectar a grandes colectivos.

Riesgos colectivos: aunque no existe una clara división entre ellos se observa que hay riesgos que al materializarse afectan o pueden afectar a muchas personas al mismo tiempo; estos riesgos son los colectivos o riesgos graves. Ejemplos de estos riesgos colectivos son las inundaciones, los terremotos, los ciclones, entre otros.

Riesgos no colectivos: riesgos cuya materialización nunca supondrá una afectación importante de personas y normalmente se limita a una única persona o a un número muy limitado. El

elemento vulnerable o aquel que sufre los efectos del riesgo tiene un papel activo, ya que hay una voluntariedad en la asunción del riesgo. (sostenible, 2010)

Parámetro temporal

Esta clasificación diferencia los riesgos que generan efectos inmediatos de los que lo hacen a medio o largo plazo.

Riesgos episódicos o puntuales: se caracterizan por sus efectos inmediatos y que requieren una respuesta inmediata para minimizar sus consecuencias. Ejemplos de estos son: los sismos, los incendios y otros.

Riesgos a medio y largo plazo: en contraposición a los anteriores existen las situaciones a largo plazo. Habitualmente se trata de fenómenos de carácter medioambiental y especialmente de cambio ambiental o climático. Su particularidad es que no requiere una respuesta inmediata porque los efectos no lo son. Estos riesgos se pueden producir no solo a escala local sino que también a escala regional.

Parámetros socioeconómicos y medioambientales

Riesgos con origen en el entorno natural o riesgos naturales: dentro de esta categoría se encuentran aquellos riesgos en los que el agente causante de peligro, es el medio natural. En función del vector ambiental o de la esfera del medio natural que causa el peligro ejemplo de estos riesgos son los climáticos y meteorológicos.

Riesgos de sociedad: Los riesgos de sociedad son aquellos que surgen del hecho de que las personas se agrupen y vivan en sociedad.

- Riesgos tecnológicos: aquellos que se derivan de la propia actividad tecnológica. La situación de riesgo es generada por la tecnología desarrollada por el ser humano.

Existe una amplia variedad de amenazas que afectan a la actividad tecnológica. La paralización del sistema puede conllevar a otro impacto aún mayor: la destrucción o desaparición de la información almacenada, que muchas veces es casi imposible de recuperar. Entre los diversos riesgos informáticos se pueden mencionar los siguientes:

- Obsolescencia de los soportes de almacenamiento: la rápida evolución de las tecnologías de almacenamiento (tarjetas perforadas, cintas magnéticas, casetes, discos magnéticos, discos compactos, etc.) implica que, al pasar el tiempo, la información grabada en un determinado soporte sea prácticamente irre recuperable al no disponerse de los periféricos de lectura adecuados.

- Problemas eléctricos y electromagnéticos: los fallos del suministro eléctrico y las radiaciones electromagnéticas pueden alterar el funcionamiento de los equipos y los datos almacenados de forma magnética.
- Riesgos que afectan a los sistemas lógicos: este tipo de riesgo suele ser uno de los más peligrosos y difíciles de detectar, ya que al alterar el funcionamiento normal del sistema y no detectarse a tiempo puede provocar daños irreparables a la información, a los usuarios e incluso al sistema físico.
- Ciberplagas: a veces también se le denomina como software malintencionado. Abarca un conjunto diverso de programas (virus, gusanos, caballos de Troya, entre otros), cuyos objetivos es adueñarse del control del sistema operativo con el fin de provocar, en la mayoría de los casos, la destrucción de la información u otros tipos de daños a los sistemas informáticos.
- Copias ilegales: cada vez más circulan por la red todo tipo de programas que permiten la copia de otros programas, documentos, lo cual ocasiona un fraude.
- Denegación de servicios: consiste en el envío de mensajes masivos a un servidor, mediante programas, con el único fin de saturarlo y bloquearlo, impidiendo el normal funcionamiento del sistema.

1.2. Gestión de riesgos

La gestión de riesgos ha sido un término polémico, no existe una definición de gestión de riesgos perfecta o tomada por consenso, como tampoco existe convergencia hacia los objetivos que se persiguen. Los diferentes autores asumen diversas perspectivas al respecto, siendo consecuentes, con la definición de riesgos que defienden, es obvio que no logran verse de igual forma las metas de la gestión de riesgo. A continuación se relacionan varias definiciones de gestión de riesgo:

La gestión de riesgos es “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 qué puede ir mal, se emplea para identificar cuáles son los riesgos más importantes en los cuales enfocarse e implementar estrategias para gestionarlos” (SEI, 2004.)

Según [Rosenberg, et al., 1999], la gestión de riesgos es importante debido a que ayuda a evitar desastres, retraso de trabajo y sobre-trabajo, pero aún más importante, porque estimula la generación de situaciones del tipo ganar-ganar. Una correcta gestión de riesgos posibilita,

por tanto, el aprovechamiento óptimo de recursos y provoca, como consecuencia, el aumento de ganancias y la disminución de pérdidas. (Maguerit, 2008.)

La gestión de riesgos se puede definir entonces como el proceso de identificación, análisis y prevención de los riesgos que amenazan activos, ganancias o personal de una organización, además de afectar los servicios que ésta provee.

1.3. Descripción de la gestión de riesgos

El proceso de la gestión de riesgos, es aplicable a un evento, a una política, a una acción, a un período de un proceso, entre otras. Del mismo modo este puede aplicarse en todos los niveles de una organización, dígame importante o estratégico. No obstante, el punto de vista que se tome y las metodologías utilizadas para tomar las decisiones, variarán de un programa a otro, o bien de una actividad a otra. Esto significa que la aplicación del proceso de gestión de riesgos se adaptará al contexto en el que este se utilice. (Báez, M. P, 2002)

El proceso de gestión y prevención de riesgos planteados en el artículo 11 de la Ley 60 de la Gaceta Oficial de la República de Cuba establece las bases para la identificación y análisis de los riesgos que enfrentan los organismos para alcanzar sus objetivos. Una vez clasificados los riesgos en internos y externos, por procesos, actividades y operaciones, y evaluadas las principales vulnerabilidades, se conforma el Plan de Prevención de Riesgos para definir el modo en que habrán de gestionarse. (Gaceta, 2011.) A continuación se presenta las etapas de la gestión y prevención de riesgos.

- Identificación de riesgos y detección del cambio
- Prevención de riesgos

Identificación de riesgos y detención del cambio (Gaceta, 2011.)

La identificación de riesgos se nutre de la 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, tecnológicos y los internos incluyen la estructura organizativa, composición de los recursos humanos, procesos productivos o de servicios.

Toda entidad debe disponer de procedimientos capaces de captar e informar oportunamente los cambios registrados o inminentes en su ambiente interno y externo, que puedan conspirar contra la posibilidad de alcanzar sus objetivos en las condiciones deseadas. Una vez identificados los riesgos se procede a su análisis, aplicando para ello el principio de importancia

relativa, determinando la probabilidad de ocurrencia y en los casos que sea posible, cuantificar una valoración estimada de la afectación o pérdida de cualquier índole que pudiera ocasionarse.

Prevención de riesgos (Gaceta, 2011.)

La prevención de riesgos constituye un conjunto de acciones o procedimientos de carácter ético-moral, técnico-organizativos y de control, dirigida 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.

1.4. Toma de decisiones

Tomar una buena decisión consiste en trazar el objetivo que se quiere conseguir, reunir toda la información relevante y tener en cuenta las preferencias del que tiene que tomar dicha decisión. El proceso de toma de decisiones se debe asumir con riesgo, porque toda decisión lo conlleva, y aunque haya sido planificada y planeada cuidadosamente en todas sus alternativas, toda elección comporta en sí misma un riesgo. A continuación se citan definiciones que exponen varios autores:

La toma de decisiones consiste en encontrar una conducta adecuada para resolver una situación problemática, en la que, además, hay una serie de sucesos inciertos. Una vez que se ha detectado una amenaza, real, imaginaria, probable o no, y se ha decidido hacer un plan para enfrentarse a ella, hay que analizar la situación: hay que determinar los elementos que son relevantes y obviar los que no lo son y analizar las relaciones entre ellos y la forma que se debe influir en ellos. (M. Corona, 2012)

La toma de decisiones se define como la selección de un curso de acciones entre alternativas, es decir, que existe un plan, un compromiso de recursos de dirección o reputación. Además es solo un paso de la planeación ya que forma la parte esencial de los procesos que se siguen para la elaboración de los objetivos o metas trazadas a seguir. (F. Barcenás, 2009)

Luego de haber realizado un estudio de varias definiciones de toma de decisiones, las autoras del trabajo deciden declarar como concepto de toma de decisiones, el siguiente: es la capacidad de elegir un curso de acción entre varias alternativas. Supone un análisis que requiere de un objetivo y una comprensión clara de las alternativas mediante las que se puede alcanzar dicho objetivo. Además de comprender la situación que se presenta, se debe analizar, evaluar, reunir alternativas y considerar las variables, comparar varios cursos de acción y finalmente seleccionar la acción que se va a realizar. La calidad de las decisiones tomadas marca la diferencia entre el éxito o el fracaso. (José A García, 2003). La selección está fundamentada en que se considera el concepto más completo y permite abarcar varios puntos de vista.

1.5. Metodología de desarrollo de software

Para cumplir el objetivo de esta investigación es imprescindible el uso de una metodología de desarrollo de software que controle cada uno de los procesos que lleva a cabo el equipo de trabajo.

El desarrollo de software es una tarea compleja y riesgosa que involucra un equipo de personas trabajando en conjunto, por lo que se hace necesario mantener un estricto control sobre los procesos de manera que se garantice la organización y coordinación de todo el trabajo. Un proceso de desarrollo de software es un conjunto de actividades para transformar los requerimientos de un usuario en un software, define quién está haciendo qué, cuándo y cómo para alcanzar un determinado objetivo. (Canós J)

Para garantizar la gestión del desarrollo de un sistema están definidas varias metodologías, pero no solo se trata de regirse por una metodología sino de seleccionar la adecuada, cada una de ellas está definida para proyectos con características específicas, por lo que se debe analizar bien cuál de las existentes va a resultar más factible emplear, pues esta decisión influye directamente en el éxito de un proyecto. (Canós J, 2003)

1.5.1 Metodologías Ágiles

Las metodologías ágiles constituyen una solución casi a medida, orientadas para proyectos pequeños y con requisitos muy cambiantes, aportando una elevada simplificación que a pesar de ello no renuncia a las prácticas esenciales para asegurar la calidad del producto. Se caracterizan por hacer énfasis en la comunicación cara a cara, es decir, se basan en una fuerte y constante interacción, donde clientes y desarrolladores trabajan constantemente juntos, estableciéndose así una estrecha comunicación. Una de las cualidades más destacables en

una metodología ágil es su sencillez, tanto en su aprendizaje como en su aplicación, reduciéndose así los costos de implantación en un equipo de desarrollo. Siempre son grupos pequeños (menores de 10 integrantes) y trabajando en el mismo sitio, utilizan pocos artefactos, pocos roles, existe menos énfasis en la arquitectura del software. Se puede hacer mención dentro de las metodologías ágiles a: XP (por sus siglas en inglés *Extreme Programming*), *Scrum*, *OpenUp*. (Canós J, 2003)

- **Scrum**

Es una metodología para la autogestión de los equipos de desarrolladores; donde estos deciden cómo hacer sus tareas y cuánto van a tardar en ello. Ayuda a que todos trabajen juntos en la misma dirección, con un objetivo claro. Se realizan entregas parciales y regulares del resultado final del proyecto, priorizadas por el beneficio que aportan al usuario final.

Scrum está especialmente indicada para proyectos en entornos complejos, donde se necesita obtener resultados de forma rápida y los requisitos son cambiantes o poco definidos. Es recomendable cuando es necesario identificar y solucionar ineficiencias sistemáticamente, o se quiere trabajar utilizando un proceso especializado en el desarrollo de producto. Esta metodología tiene algunas desventajas, por ejemplo, genera muy poca documentación en comparación con otras metodologías, y en muchas ocasiones es necesario completarla con otros procesos de XP. Dispone de herramientas para la gestión de cada una de sus fases, y es ideal para equipos de diez o menos miembros; sin embargo no indica y/o provee de ninguna práctica concreta para el desarrollo de software. (Canós J, 2003)

- **OpenUP**

Open UP constituye un Proceso Unificado de desarrollo de corta duración, aplicado de manera iterativa e incremental dentro de un ciclo de vida estructurado en tres capas. Adopta una ágil filosofía centrada en el proceso colaborativo de desarrollo de software. Preserva la esencia del Proceso Unificado con un desarrollo iterativo e incremental, dirigido por Casos de Uso, centrado en la arquitectura. Está pensado para proyectos pequeños con ciclos de vida basados en valor y riesgo. La planificación se realiza a dos niveles y la integración es continua. Tiene una arquitectura y diseño evolutivo con una visión compartida. Open UP es un proceso completo, flexible y corto, fomenta el uso de técnicas ágiles y principios, mientras que tiene un ciclo de vida estructurado y probado que hace referencia en la continua entrega de software de calidad. (Yasniel, 2011)

- **Metodología Extreme Programming (XP)**

XP es 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 la 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. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. Kent Beck, el padre de XP, describe la filosofía de XP sin cubrir los detalles técnicos y de implantación de las prácticas. Posteriormente, otras publicaciones de experiencias se han encargado de dicha tarea. (Canós J, 2003)

A continuación se describen las cuatro fases: planificación, diseño, desarrollo y pruebas, para el proceso de esta investigación.

- **Planificación:** en esta fase del ciclo de desarrollo, el cliente define a grandes rasgos las Historias de usuario. Los programadores realizan una estimación del esfuerzo necesario de cada Historia de usuario y se establece la prioridad de cada una de ellas. Se realiza el plan de entregas en función de los parámetros: tiempo de desarrollo ideal y grado de importancia para el cliente. En esta etapa el equipo de desarrollo elabora el plan de iteraciones. Todo el trabajo de la iteración es expresado en tareas de ingeniería, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por la pareja de programadores. El equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. (Beck, 1999)
- **Diseño:** en esta fase se realizan las tarjetas Cargo o Clase, Responsabilidad y Colaboración (CRC) permitiendo desprenderse del método de trabajo basado en procedimientos y trabajar con una metodología basada en objetos. Las tarjetas CRC permiten que el equipo completo contribuya en la tarea del diseño. (Beck, 1999)
- **Desarrollo:** primeramente se define que el código debe ser desarrollado siguiendo los estándares de desarrollo para facilitar su lectura y modificación por cualquier miembro del equipo de desarrollo. El desarrollo es realizado por dos personas que trabajan de forma conjunta en una computadora. De esta manera, se incrementa la calidad del software desarrollado sin afectar al tiempo de entrega. Este equipo posee unos conocimientos similares en cuanto a la tarea que van a realizar, es decir, están aproximadamente al mismo nivel. Mientras uno de ellos se encarga de pensar la táctica

con la que se va a abordar el problema, el otro se encarga de pensar las estrategias que permiten llevar dichas tácticas a su máximo exponente. Ambos roles son intercambiables. Para una mayor calidad en el código es utilizada las pruebas unitarias. (Beck, 1999)

- o **Prueba:** uno de los pilares de XP es el proceso de pruebas, esta metodología anima a probar constantemente tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones (Pressman, 2002). En esta fase el equipo de desarrollo aplica las pruebas de aceptación.

Características de la Metodología XP. (Rodríguez M., 2007)

El desarrollo bajo XP tiene características que lo distinguen claramente de otras metodologías:

- Los diseñadores y programadores se comunican efectivamente con el cliente y entre ellos mismos.
- Los diseños del software se mantienen sencillos y libres de complejidad o pretensiones excesivas.
- Se obtiene retroalimentación de usuarios y clientes desde el primer día gracias a las baterías de pruebas.
- El software es liberado en entregas frecuentes tan pronto como sea posible.
- Los cambios se implementan rápidamente tal y como fueron sugeridos.
- Las metas en características, tiempos y costos son reajustadas permanentemente en función del avance real obtenido.
- Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- El manejo del cambio se convierte en parte sustantiva del proceso.
- El costo del cambio no depende de la fase o etapa.
- No introduce funcionalidades antes que sean necesarias.
- El cliente o el usuario se convierte en miembro del equipo.

A partir de lo antes analizado las autoras asumen la metodología ágil XP (Extreme Programming). Para la selección de las herramientas a utilizar en el desarrollo de la propuesta de solución, partiendo de que el sistema a desarrollar no es complejo y se cuenta con poco tiempo para la implementación del mismo, se establecieron los siguientes parámetros:

- Adecuada para realizar proyectos pequeños.
- Adecuada para realizar proyectos ligeros.
- Tecnología libre.
- Disponibilidad en múltiples plataformas.
- Extensa documentación.

1.6. Herramienta CASE

CASE¹ es una herramienta que presenta elementos indispensables para crear una aplicación informática. El uso de las mismas reduce los costos y el tiempo en el ciclo de desarrollo del software ya que permiten modelar fácilmente procesos y sistemas, con previo dominio de al menos un lenguaje de modelado soportado por la herramienta. (Sierra, 2010) Algunas de las herramientas CASE más utilizadas son:

Rational Rose

Es una herramienta basada en modelos. Se integra con las bases de datos y los ambientes integrados de desarrollo. Todos sus productos dan soporte al lenguaje UML². Esta herramienta constituye un entorno de modelado que permite generar código a partir de modelos para múltiples lenguajes (*Rational Rose Enterprise*). Ofrece un lenguaje de modelado común que agiliza la creación del software. Tiene como desventaja que no es un software libre, su entorno gráfico no es muy amigable para el usuario y necesita alta capacidad de procesamiento (Canales, 2008).

Visual Paradigm

Es una herramienta profesional de diseño que soporta el ciclo de vida completo del desarrollo de software. Considerada como muy completa y fácil de usar, con soporte multiplataforma y que proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. Soporta todos los diagramas UML, y los Diagramas Entidad–Relación. Tiene licencia gratuita y comercial. Es fácil de instalar y actualizar, además es compatible entre ediciones. (Sierra, 2010)

¹ Del inglés Computer Aided Software Engineering.

² Del inglés Unified Modeling Language.

La primordial razón por la que fue escogida Visual Paradigm 8.0 fue por la escasez del uso de tecnologías libres durante el desarrollo del sistema de riesgos, además de su excelente integración con MySQL a la hora de generar el modelo de datos. Por otro lado es trascendental destacar que se integra con el entorno de desarrollo *NetBeans* 8.0 que es el seleccionado por el equipo de trabajo para la sistematización del software.

1.7. Herramientas y tecnologías para el desarrollo de software

Por las características de la herramienta a desarrollar se optó por una aplicación *web*, ya que esta no necesita ser descargada, instalada y configurada en cada uno de los equipos, consume menos recursos del hardware, puede ser utilizada por múltiples usuarios al mismo tiempo y para actualizarla el procedimiento se realiza desde un único servidor para todos los usuarios. Para el diseño y la implementación se seleccionaron herramientas y tecnologías libres utilizadas para la programación *web* que permiten un desarrollo efectivo y robusto de las aplicaciones, por lo cual pueden ser ejecutadas en los principales sistemas operativos sin ninguna restricción.

1.7.1 Lenguajes de programación

Un lenguaje de programación puede usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos. (Louden, 2004.)

Existe un gran número de lenguajes de programación en la actualidad que dan soporte a las tendencias actuales para las aplicaciones *web*, entre los que se encuentran PHP, CSS, HTML, JavaScript, entre otros. Estos lenguajes pueden dividirse en dos grandes grupos, los lenguajes del lado del servidor y los lenguajes del lado del cliente.

Lenguaje del lado del servidor

Los lenguajes del lado del servidor son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. (servidor, 2005) Además se encargan de manipular la información que persiste en la base de datos, la seguridad del sistema y en la mayoría de los casos realizan la lógica del negocio de la aplicación.

Hypertext Preprocessor (PHP)

PHP es un lenguaje de script interpretado en el lado del servidor, utilizado para la generación de páginas *web* dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no

necesita ser compilado para ejecutarse. Tiene como ventaja que es fácil de aprender, tiene licencia de código abierto, es un lenguaje multiplataforma, tiene una alta variedad de funciones y no requiere conocimientos de desarrollo a bajo nivel. Tiene capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otros. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. (Sæther B, 2001)

Python

Al igual que los lenguajes anteriores, Python es un software libre que posee una licencia de código abierto denominada Python Software Foundation License. Se trata de un lenguaje de programación de alto nivel cuya filosofía se centra en una sintaxis muy limpia, que favorece un código legible. Esto significa que se caracteriza por expresar los algoritmos de una manera adecuada a la capacidad cognitiva humana, en lugar de centrarse en la capacidad ejecutora de las máquinas. Es también un lenguaje multiplataforma. Esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite el desarrollo de estilos diferentes. Por todo esto, Python es considerado una apuesta por la simplicidad y versatilidad. Es un lenguaje de escritura independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de funciones, desde aplicaciones Windows hasta servidores de red e incluso páginas *web*. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo.

Se decidió utilizar como lenguaje del lado del servidor la versión 5.4 de PHP porque existe abundante documentación sobre él, el equipo de desarrollo ha trabajado con él, es un lenguaje multiplataforma, es de fácil acceso y además es libre.

Lenguaje del lado del cliente

Los lenguajes del lado del cliente son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él (Alvarez, 2001). En este caso se encuentran, HTML5, CCS3 y JavaScript entre otros, los cuales serán brevemente caracterizados en esta investigación por considerarse necesarios para la implementación del sistema.

Lenguaje de marcas de hipertexto (HTML)

HTML5 es la actualización de HTML, el lenguaje en el que es creada la *web*. Definido formalmente por un cuerpo de normas internacionales conocido como W3C³, HTML5 consta de más de cien especificaciones que se relacionan con la nueva generación de tecnologías *web*. Es una plataforma de código abierto desarrollado en términos de derechos de licencia libre. (Mauri Pérez, 2012)

La utilización de HTML5 como uno de los lenguajes para el desarrollo de la herramienta, responde a las necesidades de manipulación y maquetación de los elementos visuales de la aplicación para definir las interfaces de los riesgos, utilizando para ello las etiquetas provistas en sus especificaciones que permiten a la aplicación una interacción más potente. Estas etiquetas semánticas permiten a los desarrolladores diseñar bajo estándares y buenas prácticas de programación *web*.

Hoja de estilo en cascada (CSS)

CSS3 es un lenguaje para definir el estilo o la apariencia de las páginas *web*, escritas con HTML o de los documentos XML⁴. CSS se creó para separar el contenido de la forma, a la vez que permite a los diseñadores mantener un control mucho más preciso sobre la apariencia de las páginas. La novedad más importante que aporta CSS3 consiste en la incorporación de nuevos mecanismos para mantener un mayor control sobre el estilo con el que se muestran los elementos de las páginas, sin tener que recurrir a trucos, que a menudo complicaban el código de las *web* (Alvarez, 2011). En la aplicación se utilizará para la creación de los estilos de los componentes y el posicionamiento de ellos dentro de la página.

JavaScript

JavaScript es un lenguaje de script multiplataforma, interpretado y que se utiliza para manipular los objetos del DOM⁵. Puede incluirse en cualquier documento y es compatible con HTML, permitiendo el desarrollo de interfaces de usuario mejoradas y páginas *web* dinámicas (Eguíluz Pérez, 2009). Se decide utilizar este lenguaje, ya que permite validar los campos de los formularios y las acciones de los usuarios antes de enviar las peticiones al servidor; este método no es totalmente seguro, pero agrega un valor de funcionalidad importante al software, evitando las peticiones innecesarias al servidor.

³ e Del inglés World Wide Web Consortium.

⁴ Del inglés eXtensibleMarkupLanguage.

⁵ Del inglés DocumentObjectModel.

1.7.2 Framework de desarrollo

En la actualidad es una tendencia en el desarrollo de software la utilización de *frameworks* (o marcos de trabajo). Utilizarlos acelera el proceso de desarrollo al reutilizar el código ya existente, se promueven buenas prácticas de desarrollo con el uso de patrones y mejora la seguridad de la aplicación. Un *framework* se puede considerar como una aplicación genérica incompleta y configurable (Eguiluz, 2011). Existen varios *framework* de desarrollo según el lenguaje que se desee utilizar, entre los que se encuentran, Java Server en Java, Ruby on Rails y *Symfony* en PHP, por todas las características que este tiene hacen más fácil y comprensible su utilización.

Ruby

Ruby on Rails (RoR) es un lenguaje de programación totalmente orientado a objetos multiplataforma (lenguaje interpretado y de scripts), en el que RoR fue basado para su creación. Es uno de los más reconocidos por ciertas y específicas ventajas al programar en él. Es un *Framework* del lado del servidor que facilita la construcción de grandes aplicaciones *web* y que a su vez estas son de código abierto. Rails maneja el paradigma del MVC (*Model-View-Controller*) mejor conocido como modelo vista controlador que permite dar ciertas configuraciones de manera libre en el controlador, para darle lógica a las vistas y permitir manejar información de una base de datos mediante consultas. (Brandon, 2013)

Symfony

Symfony 2.6 es un *framework* HTTP de tipo Petición/Respuesta para desarrollar aplicaciones PHP. Facilita el desarrollo de las aplicaciones *web*, aumenta exponencialmente la productividad y ayuda a mejorar la calidad. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación *web* compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Cuenta con miles de páginas de documentación distribuidas en varios libros gratuitos y decenas de tutoriales. (Potencier, 2010)

Se seleccionó para el desarrollo del sistema a *Symfony* porque es un *framework* multiplataforma, lo que posibilita que se pueda ejecutar en cualquier Sistema Operativo como Windows y Linux en todas sus versiones y se organiza según el patrón de arquitectura Modelo Vista Controlador (MVC). Libera a los desarrolladores de la tarea de crear funcionalidades menores y, en ocasiones, aburridas de implementar. Las aplicaciones desarrolladas con *Symfony* son compatibles con la mayoría de las plataformas, bibliotecas e infraestructuras que

existen. Se adaptan a entornos de negocio en cambio permanente, requiriendo menos esfuerzo para su mantenimiento. Promueve el uso de buenas prácticas de programación y genera código fácilmente comprensible por el desarrollador. Es fácil de instalar y configurar en la mayoría de las plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y Unix estándares).

Bootstrap

Bootstrap es un *framework* de código abierto cuyo objetivo es facilitar el desarrollo de aplicaciones o páginas *web* teniendo una colección de plantillas CSS, HTML y plugins JavaScript. Los diseños creados con Bootstrap son simples, limpios e intuitivos. Con sólo agregar algunas clases y las etiquetas correctas se pueden lograr casi sin esfuerzo grupos de botones, barras de navegación, dropdowns⁶, formularios, etc., todo sin tener que escribir una línea de código CSS. Las aplicaciones que utilizan bootstrap adaptan la interfaz dependiendo del tamaño del dispositivo en el que se visualice de forma nativa, esto se denomina diseño adaptativo o *ResponsiveDesign*. (Lerner, 2012)

En esta investigación es seleccionada la versión 3.3 principalmente para el maquetado de la aplicación aunque se utilizan las demás funcionalidades que el mismo proporciona, pensando también en la compatibilidad con los distintos navegadores y dispositivos en que pueda ser utilizado el sistema.

JQuery

JQuery es una biblioteca para el lenguaje JavaScript. Implementa una serie de clases que permite programar sin preocuparse del navegador, ya que funcionan de forma exacta en todas las plataformas más habituales. La biblioteca tiene licencia de código abierto que permite que siempre cuente con soporte constante y rápido, publicándose actualizaciones de manera periódica. La ventaja principal de jQuery es que es mucho más sencillo que sus competidores. (Alvarez, 2011)

Para el desarrollo de la solución es utilizada la versión 1.10 de jQuery porque tiene una gran cantidad de plugins que pueden ser adicionados fácilmente, traduciéndose esto en un ahorro sustancial de tiempo y esfuerzo. Otra ventaja de jQuery sobre sus competidores, como Flash, es su excelente integración con AJAX⁷ que permite cargar contenido sin tener que actualizar la página.

⁶ Dropdowns: permite a los usuarios seleccionar uno o varios elementos de una lista predefinida.

⁷ Del inglés Asynchronous JavaScript and XML.

Doctrine

Doctrine 2 es un ORM⁸ para PHP 5.3.0 que proporciona persistencia transparente de objetos PHP. Se sitúa en la parte superior de una poderosa capa de abstracción de base de datos DBAL⁹. La principal tarea de los ORM es la traducción transparente entre objetos (PHP) y las filas relacionales de la base de datos.

Una característica fundamental de Doctrine es la opción de escribir las consultas de base de datos en un dialecto SQL¹⁰ propio orientado a objetos llamado DQL¹¹, inspirado en Hibernate HQL. Además DQL difiere ligeramente de SQL en que abstrae considerablemente la asignación entre las filas de la base de datos y objetos, permitiendo a los desarrolladores escribir poderosas consultas de una manera sencilla y flexible (Labs, 2009). En el sistema este ORM es utilizado en la capa del modelo de la base de datos los objetos y las relaciones entre ellos.

PHPUnit

PHPUnit es un entorno para realizar pruebas unitarias en el lenguaje de programación PHP. Es un *framework* multiplataforma que se creó con la idea de corregir los errores en cuanto se detecten. Como todos los *frameworks* de pruebas unitarias, utiliza `assert`¹² para verificar que el comportamiento de una unidad de código es el esperado (Bergmann, 2009). En el desarrollo del sistema, fue utilizado PHPUnit en su versión 4.1.0 para realizar las pruebas unitarias, que tienen como objetivo aislar cada parte del programa y demostrar que las partes de forma individual son correctas.

1.7.3 Entorno de desarrollo integrado

El programa IDE¹³ está compuesto por una serie de herramientas, tales como editor de texto, compilador, intérprete, depurador, sistema de ayuda para la construcción de interfaces gráficas de usuario, etc., que utilizan los programadores para desarrollar código. Esta herramienta puede estar pensada para su uso con un único lenguaje de programación o bien puede dar cabida a varios de estos (Fernando García, 2013). Un buen IDE tiene varias características muchas de las cuales se presentan a continuación:

- Reconocimiento de sintaxis

⁸ Del inglés Object-Relational Mapping.

⁹ Del inglés DataBaseAbstractionLayer.

¹⁰ Del inglés Structured Query Language.

¹¹ Del inglés Doctrine QueryLanguage.

¹² `assert`: validaciones de PHPUnit.

¹³ Del inglés Integrated Development Environment.

- Extensiones y componentes para el IDE
- Integración con *framework* populares
- Multiplataforma
- Integración con sistemas de control de versiones
- Depurador
- Múltiples idiomas
- Manual de usuarios y ayuda
- Importar y Exportar proyectos
- Soporte para diversos lenguajes de programación

Netbeans

Netbeans es una herramienta escrita en Java, que puede ser utilizado desde cualquier sistema operativo que tenga la máquina virtual de Java. Cuenta con características visuales para el desarrollo *web*, además es un IDE multilenguaje completo y modular que presenta soporte para Java SE, Java EE, Java ME y con vinculación de gran cantidad de módulos (plugins). Cuenta con depurador, perfilador, herramientas para refactorizaciones, completamiento de código, editores y herramientas para los lenguajes XML, HTML, CSS, PHP, JavaScript, etc. (Oracle Corporation, 2000)

Zend Studio

Zend Studio es uno de los IDE que proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código. Este agrupa todos los componentes necesarios para el ciclo de desarrollo de aplicaciones PHP, consta de dos partes: las funcionalidades del cliente (contiene interfaz de edición y ayuda) y las del servidor (instala Apache y el módulo PHP o, en caso de que estén instalados, los configura para trabajar juntos en depuración). Incluye un conjunto de herramientas de edición, depurado, análisis y optimización, lo que permite simplificar los proyectos complejos (Crusat, 2004).

Se decidió seleccionar como IDE para el desarrollo del sistema, *Netbeans* en su versión 8.0 porque posee un gran soporte para la edición en HTML5, además tiene un excelente balance entre una interfaz con múltiples opciones y completamiento de código del framework *Symfony*. Facilita un rendimiento superior para los desarrolladores de PHP, proporcionando editores y

herramientas completas para sus tecnologías, además de que no se optó por el IDE Zend Studio porque el mismo tiene carácter privativo.

1.7.4 Sistema Gestor de Base de Datos

Un Sistema Gestor de Base de Datos (SGBD) es un conjunto de programas que permiten crear y mantener una Base de Datos, asegurando su integridad, confidencialidad y seguridad. Existen diferentes tipos de SGBD, pero los más conocidos y utilizados actualmente son PostgreSQL, Oracle y MySQL. (Connolly, 2005)

PostgreSQL

El sistema gestor de Base de Datos postgresQL cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación de confiabilidad, integridad de datos y corrección. Funciona en los principales sistemas operativos, incluyendo Linux, UNIX y Windows. Tiene soporte completo para claves foráneas, uniones, vistas, disparadores y procedimientos almacenados (en varios idiomas). Se incluye la mayoría de tipos de datos de SQL y es compatible con el almacenamiento de objetos binarios, incluyendo imágenes, sonidos y vídeos.

PostgreSQL tiene como desventaja que la velocidad de gestión de bases de datos es baja aunque se mantiene estable incluso si la base de datos aumenta mucho, no incluye herencia entre objetos ya que no existen y carece de un conjunto de herramientas que permitan una fácil gestión de los usuarios y de las bases de datos que contenga el sistema. (Group, 1999)

Oracle

Entre sus principales características se encuentran que el acceso a los datos es según los privilegios concedidos por el administrador y presenta sofisticados procedimientos para hacer copias de seguridad. Se puede asignar espacio en disco para almacenar datos y es posible acceder a estos usando software de otros fabricantes. El mayor inconveniente de Oracle es que si es mal configurado puede ser muy lento. Existen pocos libros buenos sobre asuntos técnicos referentes a la instalación y administración de Oracle (Oracle Corporation, 2000). Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que sólo se vea en empresas muy grandes y multinacionales, por norma general.

MySQL

MySQL es un sistema de gestión de bases de datos relacional, esta versión tiene licencia GPL¹⁴ de la GNU¹⁵, creada por la empresa sueca MySQL AB. Las principales características de este SGBD son las siguientes: aprovecha la potencia de sistemas multiprocesadores, gracias a su implementación multihilo, es rápido, fácil de instalar y configurar, cuenta con una infinidad de bibliotecas y herramientas que facilitan su uso, tiene capacidad de gestionar y almacenar grandes cantidades de datos, gran portabilidad entre sistemas y realiza gestión de usuarios y contraseñas, manteniendo un buen nivel de seguridad en los datos. (Microsystems, 2010)

El sistema gestor de bases de datos que las autoras decidieron utilizar, fue MySQL 5.6.12 para almacenar y gestionar los datos. Este puede ser ejecutado en una computadora con escasos recursos, es software libre y gestor rápido.

1.7.5 Servidor web

Un servidor *web* o servidor HTTP es un programa que procesa cualquier aplicación del lado del servidor realizando conexiones bidireccionales y/o unidireccionales síncronas y asincrónicas con el cliente generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. (Domínguez-Dorado, 2004). La *World Wide Web* se ha hecho muy extensa a lo largo del tiempo y con esto se han creado servidores de Internet con licencia libre y propietarios. Para la selección del servidor *web* a utilizar en la investigación se han valorado los tres más utilizados, estos son Apache, Nginx y el Internet *Information Server* (IIS).

IIS

El servidor IIS es un conjunto de servicios para servidores bajo *Microsoft Windows*, presenta alto rendimiento y fiabilidad pero debe ser usado bajo licencia. A partir de la versión 6 todos los procesos de IIS se ejecutan bajo una cuenta específica que tiene menos privilegios que una cuenta de administrador del sistema pero que aporta mayor seguridad. (Domínguez-Dorado, 2004)

Nginx

Nginx es un servidor *web* de código abierto que ha demostrado ser eficaz, ligero y muy potente. Hace uso de sockets asincrónicos. Un proceso por núcleo es suficiente para manejar miles de conexiones, lo que permite una carga de CPU¹⁶ y de memoria mucho más ligera. Archivos de configuración fáciles de leer y ajustar. (Nedelcu, 2013)

¹⁴ Del inglés General Public License.

¹⁵ Del inglés GNU is Not Unix.

¹⁶ Del inglés Central Processing Unit.

Apache

Apache es uno de los servidores *web* más reconocidos a nivel mundial. Su gran popularidad se debe a múltiples características que brindan una serie de ventajas como:

- Trabaja con gran cantidad de lenguajes como Perl, PHP y otros lenguajes de script, Java y páginas JSP, teniendo todo el soporte que se necesita para tener páginas dinámicas.
- Es un servidor altamente configurable de diseño modular.
- Por su diseño modular es muy sencillo ampliar las capacidades del servidor *web* Apache. En la actualidad hay casi infinitos módulos para Apache que se pueden instalar a medida que se necesitan.
- Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.
- Tiene una alta configurabilidad en la creación y gestión de logs. Apache permite la creación de ficheros logs que facilita tener un mayor control sobre lo que sucede en el servidor. (Ruiz, 2009)

Después de evaluar estos aspectos de los servidores *web*, teniendo en cuenta que la instalación es libre, por su capacidad de configuración, robustez y estabilidad, el lenguaje de programación será PHP y el SGBD MySQL, ha sido seleccionado Apache 2.4.4 como el servidor *web* soporte de la aplicación.

1.7.6 Resumen de metodología, herramientas y tecnologías de desarrollo a utilizar.

Para la selección de las herramientas a utilizar en el desarrollo de la investigación se tuvieron en cuenta aspectos como: la abundante documentación disponible, el reconocimiento con que cuentan a nivel internacional, su condición de software libre y la experiencia de los desarrolladores en el trabajo con ellas.

Tabla 1 Resumen de metodología, herramientas y tecnologías

Herramienta o tecnología	Nombre	Versión
Metodología de desarrollo	XP	-
Herramienta CASE	Visual Paradigm	8.0

Lenguaje de programación	PHP	5.4.12
<i>Framework</i> del lado del cliente	jQuery	1.10
	Bootstrap	3.3
Framework del lado del servidor	<i>Symfony</i>	2.6
	PHPUnit	4.1.0
Ambiente de desarrollo integrado	<i>Netbeans</i> IDE	8.0
Gestor de base de datos	MySQL	5.6.12
Servidor <i>web</i>	Apache	2.4.4

1.8. Sistemas similares

Se realizó un estudio de los trabajos más significativos afines a la gestión de riesgos que se han desarrollado en la UCI, investigando los procesos definidos, la representación de los riesgos y técnicas o métodos más idóneos para llevar a cabo los procesos. El tema de la gestión de riesgos se ha abordado a través de varios trabajos de diplomas con el objetivo de indagar en todos sus aspectos y factores, se analizaron 2 trabajos desarrollados en la institución, a saber: “Gestión de la Seguridad Informática. Sistema para la gestión de riesgos”, realizado en junio del 2008 y “Propuesta de una Guía para la gestión de riesgos en el Polo de Gestión Universitaria de la Facultad 1”, desarrollado en junio del 2009.

En el primer trabajo mencionado se procura lograr una mejor eficiencia en la gestión de la Seguridad Informática. El principal resultado de esta investigación es la elaboración de una propuesta de aplicación, que cumpla con las exigencias necesarias para la implementación de un sistema que permita analizar y gestionar los riesgos de un sistema de información. Para la implementación de este sistema se utilizó como lenguaje de programación PHP 5, como sistema gestor de base de datos PostgreSQL 8.0, como servidor *web* Apache, se utiliza UML (Lenguaje de Modelado Unificado) como lenguaje de modelado, y como Herramienta CASE: Visual Parading. (Rodríguez, 2008)

Sistema para la gestión de riesgos en la Facultad 4

Por su parte en la investigación “Propuesta de una Guía para la gestión de riesgos en el Polo de Gestión Universitaria de la Facultad 1” se tuvo como premisa establecer una guía para la gestión de riesgos en el Polo de gestión Universitaria de la Facultad 1, de manera que se minimice el impacto de los riesgos que afecten los objetivos de los proyectos del Polo. Se basa en gestionar un grupo de procesos entre los que se encuentran la Planificación, Análisis, Mitigación, Control y Seguimiento e Identificación. Este trabajo utilizó como herramienta CASE Visual Parading para el modelado de diagramas. (Iglesias, 2009)

Después de haber realizado un análisis de la información recopilada sobre los sistemas informáticos identificados, vinculados al campo de acción, se puede afirmar que estos sistemas no resuelven las necesidades planteadas por la Facultad 4, ya que no gestionan los riesgos de diversos tipos, no asignan roles, y no tienen la opción de revisar y aprobar los riesgos. Siendo necesaria la construcción de un sistema para la gestión de riesgos que ayude a minimizar los riesgos en cada área de la Facultad y propicie la continuidad de los procesos.

Conclusiones

En este capítulo se expone el marco teórico conceptual de la investigación que permitió declarar los principales conceptos, entre ellos: riesgo, gestión de riesgo y toma de decisiones. Se realizó un estudio de los sistemas similares que se han desarrollado en la UCI,

Las herramientas y tecnologías seleccionadas para el desarrollo del sistema son libres, de este modo se está cumpliendo con la política de la universidad y del país de potenciar el uso del software libre, por lo que se determinó asumir la metodología XP para guiar el proceso de desarrollo, como lenguaje de modelado fue elegido UML y como herramienta CASE para realizar el modelado del software se optó por Visual Paradigm for UML, como lenguaje de programación PHP, como *framework* del lado del cliente se utilizó JQuery y Bootstrap, como *framework* del lado del servidor *Symfony* y PHPUnit, como ambiente de desarrollo integrado se seleccionó *Netbeans* IDE, como gestor de base de datos MySQL y como servidor *web* el Apache.

Capítulo 2: Descripción de la propuesta de solución

2.1 Introducción

En el presente capítulo, se identifican los procesos del negocio relacionados con la gestión de riesgos, una vez descritos permiten identificar las áreas a informatizar. Para la modelación del sistema propuesto se identifican los requisitos funcionales y los requisitos no funcionales asociados a estas áreas, agrupándolos por Historias de usuarios a priorizar y descritos para obtener la documentación necesaria para realizar el diseño del sistema.

2.2 Modelo conceptual de negocio

A pesar de no exigirse en la metodología XP, es de gran ayuda para el equipo de desarrollo realizar un modelo conceptual de negocio que permita un mejor entendimiento de los conceptos asociados al negocio y sus relaciones.

Un modelo de negocio describe la lógica de un sistema de negocios. Se crea con el fin de dejar claro quiénes son los actores que se encuentran en un caso de negocio. Las relaciones en un modelo de negocio se formulan en términos de valores intercambiados entre los actores. (Microsoft, 2008)

Plan de riesgo de la Facultad	Documento que contiene los riesgos de la Facultad y es elaborado por la comisión de riesgo y aprobado por el consejo de dirección.
Riesgos	Son los riesgos positivos o negativos existentes en las áreas.
Responsable de área:	Persona encargado de elaborar el plan de riesgos de las áreas y el plan de respuesta de los riesgos.
Plan de riesgos de las áreas	Documento que contiene los riesgos de cada área y es elaborado por los responsables de área dígame departamentos, vicedecanatos, decanato y centro de desarrollo.
Decano	Persona encargada de aprobar todos los planes de riesgos de las áreas.
Vicedecano Administrativo	Persona encargada de realizar el plan de riesgos de su área y el plan de respuesta de los riesgos, además de revisar el plan de riesgo de la Facultad.

2.3 Especificación de los requisitos de software

Los requisitos se pueden clasificar en funcionales y no funcionales. Ellos muestran las capacidades o condiciones que el sistema debe cumplir y las propiedades o cualidades que el producto debe tener. En la fase de construcción deben ser verificables. De ahí la importancia de la identificación correcta de los requisitos a tener en cuenta para el éxito del sistema. El nivel de detalle de los requisitos es crucial para que el equipo de desarrollo pueda diseñar un componente que los satisfaga y los encargados de las pruebas puedan evaluarlos.

2.3.1 Requisitos Funcionales

RF.1: Insertar riesgo

Descripción: El usuario podrá insertar un riesgo a partir de:

Entrada:

- ✓ Los datos del riesgo que el usuario debe introducir como riesgo, actividad, afectación, impacto, detección, estado y frecuencia.

Salida:

- ✓ Un riesgo insertado en la Base de Datos.

RF.2: Insertar respuesta del riesgo

Descripción: El usuario podrá insertar una respuesta al riesgo a partir de:

Entrada:

- ✓ Los datos de la respuesta que el usuario debe introducir como medidas a aplicar, activación, responsable, ejecutante, posibles manifestaciones y riesgo.

Salida:

- ✓ La respuesta del riesgo insertado en la Base de Datos.

RF.3: Insertar área

Descripción: El usuario podrá insertar un área a partir de:

Entrada:

- ✓ Los datos del área que el usuario debe introducir como el nombre de la misma y si está activa o no.

Salida:

- ✓ El área insertada correctamente.

RF.4: Mostrar gráficas

Descripción: El usuario podrá visualizar, informaciones referentes a los riesgos en forma de gráficas, a partir de:

Salida:

- ✓ Visualización de las gráficas, mostrando la cantidad de riesgos con un mismo valor por frecuencia, detección e impacto.

RF.5: Consultar trazas

Descripción: El usuario podrá consultar las trazas, la cual mostrará la lista de todas las operaciones que se han realizado en el sistema, además cuenta con un filtro para hacer menos engorrosa la búsqueda de un elemento específico.

Salida:

- ✓ Visualización de las acciones que se han realizado, mostrando, elemento, id, acción, fecha, hora y usuario.

RF.6: Exportar a PDF

Descripción: Para exportar a PDF el usuario escoge un plan específico de la lista y hace clic en Exportar:

Salida:

- ✓ Se mostrará un PDF con todos los datos del plan, lista de riesgos con su lista de respuestas.

RF.7: Insertar usuario.

Descripción: Para insertar un usuario se mostrará una vista con todos los datos que deben ser introducidos.

Entrada:

- ✓ Los datos del usuario que deben introducir como, usuario, correo y rol.

Salida:

- ✓ Un usuario insertado correctamente.

RF.8: Buscar

Descripción: Se selecciona los aspectos que se desea, una vez seleccionados, se da clic en Buscar y mostrará los riesgos que coinciden con los aspectos seleccionados.

Entrada:

- ✓ Los elementos por los que se desea buscar, como: área, año, trimestre, con respuesta, estado, impacto, detección y frecuencia.

Salida:

- ✓ Lista de riesgos que coincida con las características especificadas.

RF.9: Insertar categoría.

Descripción: Para insertar una categoría se mostrará una vista con todos los datos que deben ser introducidos:

Entrada:

- ✓ Los datos de la categoría que deben introducir como, nombre, descripción y si está activa.

Salida:

- ✓ La categoría insertada correctamente.

RF.10: Mostrar reportes.

Descripción: Al dar clic en Reportes, se mostrará una lista de todos los riesgos del área y se dará la opción de buscar respecto a varios aspectos

Salida:

- ✓ Lista de riesgos con los datos iguales a los especificados.

RF.11: Listar riesgos.

Descripción: Al dar clic en Gestión de Riesgos se muestra el listado de riesgos creados en el sistema y la descripción de cada uno de ellos:

Salida:

- ✓ Lista de riesgos del área.

RF.12: Ver riesgo.

Descripción: Se mostrará los detalles del riesgo al dar clic en Ver:

RF.13: Editar riesgo.

Descripción: Al hacer clic en Editar se visualizarán los detalles del riesgo que pueden ser editados como riesgo, actividad, afectación, impacto, detección, estado y frecuencia:

Salida:

- ✓ Riesgo modificado correctamente.

RF.14: Incluir riesgo en el plan de la Facultad.

Descripción: Se selecciona un riesgo de determinada área y al dar clic en Incluir, automáticamente pasará al plan de la Facultad.

Salida:

- ✓ Nuevo riesgo insertado en el plan de la Facultad.

RF.15: Listar Plan.

Descripción: Al hacer clic en Planes se mostrará una lista con todos los planes creados en el sistema y la descripción de cada uno de ellos:

Salida:

- ✓ Lista de planes de determinada área.

RF.16: Mostrar riesgos de un plan.

Descripción: Se selecciona un plan determinado y al dar clic en Ver Riesgos se mostrará un listado con todos los riesgos asociados a ese plan:

Salida:

- ✓ Lista de riesgos con sus descripciones.

RF.17: Mostrar respuestas de un plan.

Descripción: Se selecciona un plan determinado y al dar clic en Ver Respuesta se mostrará un listado con todas las respuestas de los riesgos asociados a ese plan:

Salida:

- ✓ Lista de respuestas de los riesgos con sus descripciones.

RF.18: Modificar respuesta del riesgo.

Descripción: Se selecciona una respuesta determinada y al dar clic en Editar se visualizará los datos de la respuesta del riesgo:

Entrada:

- ✓ Los elementos de la respuesta que se puede modificar como: medidas a aplicar, activación, responsable, ejecutante, posibles manifestaciones y riesgo.

Salida:

- ✓ Respuesta modificada correctamente.

RF.19: Eliminar respuesta del riesgo.

Descripción: Se escoge una respuesta y al dar clic en Eliminar, automáticamente eliminará la respuesta.

Salida:

- ✓ Mensaje informando que la respuesta se ha eliminado correctamente.

RF.20: Mostrar área.

Descripción: Se selecciona un área determinada y al dar clic en Ver se mostrarán los detalles del área especificada:

Salida:

- ✓ Descripción del área.

RF.21: Listar área.

Descripción: Al dar clic en Área se mostrará la lista de todas las áreas insertadas en el sistema:

Salida:

- ✓ Lista de áreas con sus descripciones.

RF.22: Editar área.

Descripción: Se selecciona un área determinada y al dar clic en Editar, se mostrarán los elementos que pueden ser modificados:

Entrada:

- ✓ Los datos necesarios para la modificación como: nombre y si está activa.

Salida:

- ✓ El área editada correctamente.

RF.23: Eliminar área.

Descripción: Se selecciona un área determinada y al dar clic en Eliminar, automáticamente eliminará el área.

Salida:

- ✓ Mensaje informando que el área ha sido eliminada correctamente.

RF.24: Listar usuario.

Descripción: Al hacer clic en Usuario, mostrará una lista con todos los usuarios del sistema.

Salida:

- ✓ Lista de usuarios con sus descripciones: id, usuario y rol.

RF.25: Mostrar usuario.

Descripción: Se selecciona un usuario y al dar clic en Ver se mostrará los datos del usuario.

Salida:

- ✓ Descripción del usuario: id y usuario, además de brindar la opción de eliminarlo.

RF.26: Eliminar usuario.

Descripción: Se selecciona un usuario y al dar clic en Eliminar, automáticamente eliminará el usuario.

Salida:

- ✓ Mensaje informando que el usuario se ha eliminado correctamente.

RF.27: Modificar categoría.

Descripción: Se selecciona una categoría y al dar clic en Editar, se mostrará la categoría con su descripción:

Entrada:

- ✓ Datos necesarios para la modificación de la categoría: nombre, descripción y si está activa o no.

Salida:

- ✓ La categoría modificada correctamente.

RF.28: Mostrar categoría.

Descripción: Se selecciona una categoría y al dar clic en Ver, se mostrará la categoría con su descripción:

Salida:

Categoría seleccionada con su descripción: id, categoría, tipo y activa.

RF.29: Eliminar categoría.

Descripción: Se selecciona una categoría y al dar clic en Eliminar, automáticamente la eliminará de la lista.

Salida:

Mensaje informando que la categoría se ha eliminado correctamente.

2.3.2 Requisitos no funcionales

Algunos de los atributos propios de un sistema eficaz no se pueden describir en términos de funcionalidad. En la práctica, los requisitos no funcionales son primordiales para el éxito de los sistemas. A continuación se describen los requisitos no funcionales, en este caso se adoptan los que están estrechamente relacionados con la solución propuesta.

Usabilidad

RNF 1. Tipo de aplicación informática: aplicación *web*.

Requerimiento de software

RNF 2. Sistema operativo: Windows Server 2003.

Es necesario que estén instaladas las siguientes aplicaciones:

Servidor de aplicaciones *web* Apache 2.4.4

Base de datos MySQL 5.6.12

Netbeans IDE 8.0

RNF 3. El sistema deberá ser capaz de responder a cualquier petición antes de 5 segundos.

RNF 4. Idioma: se utilizó el idioma español para los mensajes y textos de la interfaz.

Restricciones de diseño

RNF 5. Lenguaje: Se utiliza para la construcción del sistema el lenguaje de programación PHP, herramienta que se distribuye bajo licencias libres.

Interfaz

RNF 6. Navegadores: Para acceder al Sistema de Gestión de Riesgos debe usarse como navegador el chrome en su versión 25.0.1364.172m . No se garantiza la correcta visualización en otros navegadores.

Interfaces de comunicación

RNF 7: El sistema debe brindar la posibilidad de autenticar los usuarios utilizando un servidor LDAP.

2.4 Roles del sistema

Los usuarios del sistema son roles predefinidos encargados de interactuar con el sistema. A continuación se describen los usuarios del sistema con sus responsabilidades.

Tabla 2 Roles del sistema

Roles	Responsabilidades
Responsable de área	Responsable de gestionar los riesgos, generar el plan de riesgos y el plan de respuesta de los riesgos de su área.
Vicedecano administrativo	Responsable de gestionar los riesgos, genera el plan de riesgos y el plan de respuesta de riesgos de su área y de la Facultad. Es el encargado de gestionar los usuarios y revisar los riesgos de cada área en estado de revisión y cambiarle el estado de aprobación.
Decano	Encargado de gestionar los riesgos, generar el plan de riesgos y el plan de respuesta de riesgos de la Facultad. Es el encargado de gestionar los usuarios. Tiene la responsabilidad de insertar, eliminar y editar las áreas y las categorías. Revisa los riesgos de cada área en estado de aprobación y los pone en estado de aprobado.
Invitado	Este rol solo tendrá permiso de lectura, es decir podrá ver los riesgos existente en las áreas y no podrá ni modificarlos ni eliminarlos.

2.5 Exploración

En esta fase, los clientes plantean a grandes rasgos las Historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología. (Anderson, 2011)

Las estimaciones en esta fase son primarias y podrían cambiar cuando se analice más en detalle cada iteración, obteniendo como resultado una visión general del sistema y un plazo total estimado. Se realizan tres operaciones esenciales: escribir las Historias de usuario, estimarlas y dividir las en unas más pequeñas o concentrarlas en otras más generales.

2.5.1 Historias de usuario

Las Historias de usuario (HU) son utilizadas en XP para especificar los requerimientos del software. Se trata de tarjetas en las cuales el cliente describe brevemente las características que el sistema debe poseer. El tratamiento de las Historias de usuario es muy dinámico y flexible, en cualquier momento las Historias de usuario pueden reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. (Beck, 1999)

XP no propone ningún formato específico para las Historias de usuario, deja los detalles a consideración del equipo de desarrollo. Teniendo en cuenta los datos necesarios para la planificación y estimación de las Historias de usuario y las plantillas utilizadas en desarrollos anteriores, se utilizó la siguiente plantilla:

Tabla 3 Modelo propuesto para una historia de usuario

Historia de usuario	
Número: Número asignado a la HU.	Nombre: Nombre de la HU.
Usuario: Rol o roles que realiza las acciones dentro de la HU.	
Prioridad: Nivel de prioridad de la HU en el negocio.(Alta,Media,Baja)	Riesgo: nivel de riesgo en caso de no realizarse la HU(Alto,Medio,Bajo)
Estimación: cuándo el valor es 1 equivale a una semana ideal de trabajo. En la metodología XP está definida una semana ideal como 5 días hábiles trabajando 40 horas, es decir, 8 horas diarias. Por lo que cuándo el valor de dicho atributo es de 0.5 equivale a 2 días y medio de trabajo, lo que se traduce en 20 horas.	Iteración: Iteración en la que se realizará
Descripción: Breve descripción de lo que realizará la HU.	

Observación: Información extra que se estime agregar para hacer más comprensible la HU.

Seguidamente se muestran las Historias de usuario confeccionadas para el desarrollo del sistema de gestión de riesgos.

Tabla 4 HU- Insertar riesgo

Historia de usuario	
Número: 1	Nombre: Insertar riesgo
Usuario: Responsable de área, Vicedecano administrativo y Decano.	
Prioridad: Alta	Riesgo: Alto
Estimación: 0.9	Iteración: 1
<p>Descripción: Al hacer clic en la opción insertar riesgos se muestra una vista en forma de formulario solicitando los datos necesarios para crear el riesgo:</p> <ul style="list-style-type: none"> - Riesgo - Afectación - Impacto - Detección - Frecuencia - Categoría 	
Observación:	

Tabla 5 HU- Insertar respuesta del riesgo

Historia de usuario	
Número: 2	Nombre: Insertar respuesta del riesgo
Usuario: Responsable de área, Vicedecano administrativo y Decano.	
Prioridad: Alta	Riesgo: Alto
Estimación: 0.8	Iteración: 1
<p>Descripción: Al hacer clic en la opción insertar respuesta del riesgo se muestra una vista en forma de formulario solicitando los datos necesarios para crear la respuesta:</p> <ul style="list-style-type: none"> - Medidas a aplicar - Activación - Responsable - Ejecutante - Posibles manifestaciones 	
Al concluir se muestra un mensaje confirmando que fue insertada la respuesta.	
Observación:	

Tabla 6 HU- Insertar área

Historia de usuario	
Número: 3	Nombre: Insertar área
Usuario: Decano.	
Prioridad: Alta	Riesgo: Alto
Estimación: 0.7	Iteración: 1
<p>Descripción: Al hacer clic en Insertar área, se abre una vista solicitando los datos necesarios para la creación de la misma:</p> <ul style="list-style-type: none"> - Nombre - Activa, si se marca la opción de activa, entonces el área podrá tener riesgos asociados a ella, si está no activa, esta área no se podrá utilizar, no se podrán insertar riesgos en esa área. Un área se puede desactivar para dejar de usarse pero los riesgos asociados a ella, permanecerán, no se eliminarán. <p>Se le da clic en la opción crear y al concluir se muestra un mensaje confirmando que fue creada correctamente.</p>	
Observación:	

Tabla 7 HU- Mostrar gráficas

Historia de usuario	
Número: 4	Nombre: Mostrar gráficas.
Usuario: Responsable de área, Vicedecano administrativo y Decano.	
Prioridad: Alta	Riesgo: Alto
Estimación: 0.8	Iteración: 1
<p>Descripción: Al hacer clic en Ver Gráficas de un plan específico, se visualizará gráficas de barras mostrando la cantidad de riesgos con valores iguales por frecuencia, detección e impacto, así como el año y el trimestre del plan al que pertenecen.</p>	
Observación:	

Tabla 8 HU- Consultar trazas

Historia de usuario	
Número: 5	Nombre: Consultar trazas
Usuario: Decano.	
Prioridad: Alta	Riesgo: Alto
Estimación: 0.8	Iteración: 1
<p>Descripción: Al hacer clic en Trazas se muestra la lista de todas las operaciones que se han realizado en el sistema. Además se puede filtrar para buscar las acciones que se desea visualizar.</p>	
Observación:	

Tabla 9 HU- Exportar en pdf

Historia de usuario	
Número: 6	Nombre: Exportar en pdf
Usuario: Responsable de área, Vicedecano administrativo y Decano.	
Prioridad: Alta	Riesgo: Alto
Estimación: 0.8	Iteración: 1
Descripción: Se escoge un plan de la lista y al dar clic en Exportar, se mostrará un pdf, con la información de todos los riesgos y sus respuestas.	
Observación:	

Tabla 10 HU- Insertar usuario

Historia de usuario	
Número: 7	Nombre: Insertar usuario
Usuario: Vicedecano administrativo y Decano.	
Prioridad: Alta	Riesgo: Alto
Estimación: 0.7	Iteración: 1
Descripción: Al dar clic en Insertar Usuario, se muestran los datos necesarios para la creación de un usuario: <ul style="list-style-type: none"> • Usuario • Correo • Rol • Área 	
Observación:	

Tabla 11 HU- Buscar

Historia de usuario	
Número: 8	Nombre: Buscar
Usuario: Responsable de área, Vicedecano administrativo y Decano.	
Prioridad: Alta	Riesgo: Alto
Estimación: 0.8	Iteración: 1
Descripción: Se selecciona los aspectos que se desea: <ul style="list-style-type: none"> - Área - Año - Trimestre - Con respuesta o sin respuesta - Estado 	

<ul style="list-style-type: none"> - Impacto - Detección - Frecuencia <p>Una vez seleccionados, se da clic en Buscar, buscará los riesgos que coinciden con los aspectos seleccionados y mostrará todos los que cumplan con lo especificado anteriormente.</p> <p>Observación:</p>
--

Tabla 12 HU- Insertar categoría

Historia de usuario	
Número: 9	Nombre: Insertar categoría
Usuario: Decano.	
Prioridad: Alta	Riesgo: Alto
Estimación: 0.4	Iteración: 1
<p>Descripción: Al dar clic en insertar categoría se muestra una vista solicitando los datos necesarios para añadirlo:</p> <ul style="list-style-type: none"> - Nombre - Descripción - Activa, si se marca la opción activa, entonces la categoría podrá tener riesgos asociados a ella, si está no activa, esta categoría no se podrá utilizar, no se podrá insertar riesgos con esa categoría. Una categoría se puede desactivar para dejar de usarse pero los riesgos asociados a ella, permanecerán, no se eliminarán. <p>Además brinda la opción de Eliminar. Al concluir se muestra un mensaje confirmando que fue creada correctamente.</p> <p>Observación:</p>	

Tabla 13 HU- Mostrar reportes

Historia de usuario	
Número: 10	Nombre: Mostrar reportes
Usuario: Responsable de área, Vicedecano administrativo y Decano.	
Prioridad: Alta	Riesgo: Alto
Estimación: 0.4	Iteración: 1
<p>Descripción: Al dar clic en Reportes, se mostrará una lista de todos los riesgos del área y se dará la opción de buscar respecto a varios aspectos:</p> <ul style="list-style-type: none"> - Área - Año - Trimestre 	

- Con respuesta o sin respuesta
- Estado
- Impacto
- Detección
- Frecuencia

Observación:

Fueron descritas por parte del cliente un total de 29 HU, en este epígrafe se muestran las principales, las restantes se encuentran en los anexos de la investigación.

2.6 Planificación

En esta fase del ciclo de desarrollo el cliente especifica a grandes rasgos las Historias de usuario. Se realiza el plan de entregas en función de los parámetros: tiempo de desarrollo ideal y grado de importancia para el cliente, además se elabora el plan de iteraciones por parte del equipo de desarrollo. Los programadores realizan una estimación del esfuerzo necesario de cada historia de usuario y se establece la prioridad de cada una de ellas. (Anderson, 2011) Todo el trabajo de la iteración es expresado en tareas de ingeniería, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por la pareja de programadores. El equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto.

2.6.1 Estimación de esfuerzos por historia de usuario

Las estimaciones de esfuerzos asociadas a la implementación de las HU son establecidas por los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de trabajo. Las HU deben ser programadas en un tiempo estimado de hasta tres semanas. Si la estimación supera las tres semanas la historia deberá ser dividida hasta que pueda ser desarrollada en un tiempo factible. (Joskowicz, 2008)

No	Historia de usuario	Estimación
1	Insertar riesgos	0.8
2	Insertar respuesta del riesgo	0.7
3	Insertar área	0.8
4	Mostrar gráficas	0.8

Sistema para la gestión de riesgos en la Facultad 4

5	Consultar trazas	0.7
6	Exportar en pdf	0.8
7	Insertar usuario	0.4
8	Buscar	0.4
9	Insertar categoría	0.3
10	Mostrar reportes	0.2
11	Listar riesgos	0.4
12	Ver riesgo	0.2
13	Editar riesgo	0.4
14	Incluir riesgo en el plan	0.3
15	Listar Plan	0.4
16	Mostrar riesgos de un plan	0.3
17	Mostrar respuestas de un plan	0.3
18	Modificar respuesta del riesgo	0.4
19	Eliminar respuesta del riesgo	0.1
20	Mostrar área	0.5
21	Listar área	0.5
22	Editar área	0.4
23	Eliminar área	0.4
24	Listar usuario	0.6
25	Mostrar usuario	0.6
26	Eliminar usuario	0.5
27	Modificar categoría	0.5
28	Mostrar categoría	0.2

29	Eliminar categoría	0.4
----	--------------------	-----

2.6.2 Plan de entregas

El plan de entregas es un documento que especifica exactamente que Historias de usuario serán implementadas en cada entrega del sistema y sus prioridades, de modo que también permita conocer con exactitud qué Historias de usuario serán implementadas en la próxima liberación. Debe ser negociado y elaborado en forma conjunta entre el cliente y el equipo desarrollador durante las reuniones de planificación de entregas, la idea es hacer entregas frecuentes para obtener una mayor retroalimentación. (Escribano, 2002) A partir de una reunión con el cliente, teniendo en cuenta las necesidades trazadas por el mismo y las evaluaciones realizadas por los desarrolladores, el equipo de desarrollo definió el siguiente Plan de Entregas:

Tabla 14 Plan de entregas

Entregable	1ra entrega (4ta semana de Febrero)	2da entrega (4ta semana de Marzo)	3ra entrega (3ra semana de Abril)
Sistema para la Gestión de riesgos de la Facultad 4	Versión 0.1	Versión 0.2	Versión 0.3

Historia de usuario	1ra entrega	2da entrega	3ra entrega
Insertar riesgos	x	x	x
Insertar respuesta del riesgo	x	x	x
Insertar área	x	x	x
Mostrar gráficas	x	x	x

Sistema para la gestión de riesgos en la Facultad 4

Consultar trazas	x	x	x
Exportar en pdf	x	x	x
Insertar usuario	x	x	x
Buscar	x	x	x
Insertar categoría	x	x	x
Mostrar reportes	x	x	x
Listar riesgos		x	x
Ver riesgo		x	x
Editar riesgo		x	x
Incluir riesgo en el plan		x	x
Listar Plan		x	x
Mostrar riesgos de un plan		x	x
Mostrar respuestas de un plan		x	x
Modificar respuesta del riesgo		x	x
Eliminar respuesta del riesgo		x	x
Mostrar área		x	x
Listar área		x	x
Editar área			x
Eliminar área			x
Listar usuario			x
Mostrar usuario			x
Eliminar usuario			x
Modificar			x

categoria			
Mostrar categoría			x
Eliminar categoría			x

2.6.3 Plan de iteraciones

Todo proyecto que siga la metodología X.P, se ha de dividir en iteraciones de aproximadamente 3 semanas de duración. Para cada una de las iteraciones el cliente establece un conjunto de HU que serán implementadas en cada iteración del sistema. Estas Historias de usuario son divididas en tareas de entre 1 y 5 días de duración que se asignarán a los programadores. (Escribano, 2002). Al final de cada iteración se realizan las pruebas de aceptación y la aplicación tendrá implementadas funcionalidades para dar cumplimiento a los objetivos propuestos.

Tabla 15 Plan de iteraciones

Iteración	Orden de las HU	Cantidad de tiempo de trabajo
1	Insertar riesgos Insertar respuesta del riesgo Insertar área Mostrar gráficas Consultar trazas Exportar en pdf Insertar usuario Buscar Insertar categoría Mostrar reportes	4 semanas y 2 días\medio
2	Listar riesgos Ver riesgo	3 semanas y 4 días\medio

	Editar riesgo Incluir riesgo en el plan Listar Plan Mostrar riesgos de un plan Mostrar respuestas de un plan Modificar respuesta del riesgo Eliminar respuesta del riesgo Mostrar área	
3	Listar área Editar área Eliminar área Listar usuario Mostrar usuario Eliminar usuario Modificar categoría Mostrar categoría Eliminar categoría	3 semanas y medio día

El plan de duración de las iteraciones se realizó luego de tener el estimado en días que demoraba implementar cada HU. Se tuvo en cuenta además la prioridad que el cliente le asignó.

Conclusiones

En el desarrollo de este capítulo se mostraron los artefactos generados por las dos primeras fases de la metodología XP, asegurando una implementación por etapas correctamente detalladas y especificadas. En estas fases se fundamentaron las HU necesarias para la implementación del sistema y se generó el Plan de entregas con las fechas en las que será liberada cada versión del sistema. En las etapas de Exploración y Planificación aparecen elementos fundamentales para el desarrollo del sistema, como la entrevista entre el cliente y los desarrolladores.

Capítulo 3 Validación de la propuesta

3.1 Fase de diseño

En la fase de diseño se especifica la arquitectura, los patrones de diseño y se realiza el modelo de datos para una mejor visión de la distribución del sistema. Además se confeccionan las tarjetas clase-responsabilidad-colaborador (CRC) para la descripción de cada una de las clases y entidades. (Anderson, 2011)

3.2 Arquitectura

En la propuesta del trabajo realizado fue escogido el patrón arquitectónico Modelo Vista Controlador (MVC), su buena organización en capas permite que cualquier cambio que ocurra en el código no afecte el diseño ni las demás funciones del sistema, se divide como bien define

el modelo (la lógica del negocio), la vista (las páginas de presentación) y el controlador (los métodos)

Los **patrones arquitectónicos**, o patrones de arquitectura, son patrones de software que ofrecen soluciones a problemas de arquitectura de software en ingeniería de software. Especifican un conjunto predefinido de subsistemas con sus responsabilidades y una serie de recomendaciones para organizar los distintos componentes. Resuelven problemas arquitectónicos, adaptabilidad a requerimientos cambiantes, performance, modularidad y acoplamiento, etc. La solución que plantea es la creación de patrones de llamadas entre objetos (similar a los patrones de diseño), decisiones y criterios arquitectónicos, empaquetado de funcionalidad. Este tipo de patrones se utilizan en la fase de desarrollo, en el diseño inicial. Los beneficios de la utilización de dichos patrones van desde la imposición de decisiones tempranas en el desarrollo hasta la reutilización. (Adriana, 2010)

Symfony2 la arquitectura que utiliza es Modelo-Vista-Controlador (MVC), esta es una de la más utilizada por los *frameworks* para PHP. *Symfony2* solo proporciona herramientas para la parte del Controlador y de la Vista. La parte del Modelo es responsabilidad del programador, aunque existen bibliotecas para integrar fácilmente los ORM más conocidos, como Doctrine y Propel. (Potencier, 2011)

Las capas en el sistema quedan de la siguiente forma:

El Modelo es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vista. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuándo cambia el Modelo.

La Vista es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al propio Modelo.

El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. (CodeIgniter, 2013)

Ventajas

Separación clara entre los componentes de un programa; permitiendo su implementación por separado.

Conexión entre el Modelo y sus Vistas dinámicas; se produce en tiempo de ejecución, no en tiempo de compilación.

Los sistemas además de estar creados bajo un patrón de arquitectura deben presentar también patrones de diseño y *Symfony* utiliza en su arquitectura la mayoría de los patrones de diseño que existen, a continuación se mencionan algunos de estos. (Fernández Y., 2012)

3.3 Patrón de diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular. Cada patrón describe un problema que ocurre una y otra vez en el entorno y describe también el núcleo de la solución al problema, de forma que puede utilizarse un millón de veces sin tener que hacer dos veces lo mismo. (63) A continuación se muestran los patrones de diseño que se definieron en el trabajo.

3.3.1 Patrones GRASP

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Los patrones GRASP básicos se refieren a cuestiones y aspectos fundamentales del diseño, ellos son:

Bajo acoplamiento: El bajo acoplamiento es un principio que se debe recordar durante las decisiones de diseño: es la meta principal que es preciso tener siempre. Es un patrón evaluativo que el diseñador aplica al juzgar sus decisiones de diseño. Asigna una responsabilidad a una clase para mantener bajo acoplamiento, o sea, disminuir la dependencia entre clases, evitando que una modificación en alguna de ellas repercuta en gran medida en el resto, posibilitando además una mayor reutilización. (Larman, 2004)

En el sistema las clases *AreaController* y *RiesgoController* heredan solamente de *Controller*, por lo que se garantiza que exista un grado moderado de acoplamiento entre las clases. Además, al no asociar las clases del modelo con las de la vista o el controlador, la dependencia entre las clases, en este caso, se mantiene baja.

Alta Cohesión: La información que almacena una clase debe ser coherente y debe estar relacionada con la clase, asignando responsabilidades con una alta cohesión (Larman, 2004). La alta cohesión dice que cada clase debe contener toda la información (atributos) y funcionalidades (métodos) asociadas al objeto que representan. Siempre se ve en concordancia con el bajo acoplamiento porque si las clases modelan toda la información asociada a un objeto, significa que hay poca dependencia de otras clases externas para su correcto funcionamiento. En *Symfony2* se pone de manifiesto en las clases entidad.

Creador: El patrón creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. Al escogerlo como creador, se da soporte al bajo acoplamiento. (Larman, 2004)

En *Symfony2* en la clase Controller se definen y ejecutan todas las acciones. En las acciones se crean los objetos de las clases que representan las entidades, evidenciando de este modo que la clase Controller es "creador" de dichas entidades.

En *Symfony2* todas las peticiones *web* son manejadas por el controlador frontal, que es el punto de entrada único de toda la aplicación en un entorno determinado.

3.3.2 Patrones GOF

Symfony utiliza casi todos los patrones de diseño que existen, esto lo convierte en un *framework* de elegancia, pero en este trabajo solo será descrito el **Decorador**.

Decorador: El propósito de este patrón es añadir responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la especialización mediante herencia, cuándo se trata de añadir funcionalidades (Larman, 2004). Este patrón añade funcionalidad a una clase de forma dinámica y transparente. En la aplicación el archivo *base.html.twig* o plantilla global almacena el código HTML que es común a todas las páginas del sistema, de ella hereda el archivo *layout.html.twig* que es el que contiene el diseño general del sistema para no tener que repetirlo en cada página.

3.4 Tarjetas CRC

Las tarjetas CRC (Class-Responsibility-Collaboration) es una técnica de diseño orientado a objetos, propuesta por Kent Beck y Ward Cunningham. Permiten la representación de un objeto mediante una clase. El objetivo de esta técnica es hacer, mediante tarjetas, un inventario de las clases que se van a necesitar para implementar el sistema y la forma en que van a interactuar, de esta forma se pretende facilitar el análisis y discusión de las tarjetas por parte de varios actores del equipo de proyecto con el objetivo de que el diseño sea lo más simple posible, verificando las especificaciones del sistema. (R, 2010)

Seguidamente se muestran las tarjetas CRC de las clases controladoras, confeccionadas para el desarrollo del sistema de gestión de riesgos de la Facultad 4. Las restantes se pueden encontrar en los anexos del documento.

Tabla 16 Tarjeta CRC ClaseCategoriaController

Nombre de la clase: CategoriaController	
Responsabilidades	Colaboraciones
indexAction createAction createCreateForm newAction showAction editAction createEditForm updateAction deleteAction createDeleteForm	Categoria, CategoriaType

Tabla 17 Tarjeta CRC ClaseAreaController

Nombre de la clase: AreaController	
Responsabilidades	Colaboraciones
indexAction createAction createCreateForm newAction showAction editAction createEditForm updateAction	Area, AreaType

deleteAction	
createDeleteForm	

Tabla 18 Tarjeta CRC ClasePlanController

Nombre de la clase: PlanController	
Responsabilidades	Colaboraciones
indexAction createAction createCreateForm newAction showAction editAction createEditForm updateAction deleteAction createDeleteForm	Plan, PlanType

Tabla 19 Tarjeta CRC ClaseRespuestaController

Nombre de la clase: RespuestaController	
Responsabilidades	Colaboraciones
indexAction createAction createCreateForm newAction	Respuesta, RespuestaType

showAction	
editAction	
createEditForm	
updateAction	
deleteAction	
createDeleteForm	

Tabla 20 Tarjeta CRC ClaseRiesgoController

Nombre de la clase: RiesgoController	
Responsabilidades	Colaboraciones
indexAction createAction createCreateForm newAction showAction editAction createEditForm updateAction deleteAction createDeleteForm	Riesgo, RiesgoType

Tabla 21 Tarjeta CRC ClaseTrazaController

Nombre de la clase: TrazaController	
Responsabilidades	Colaboraciones

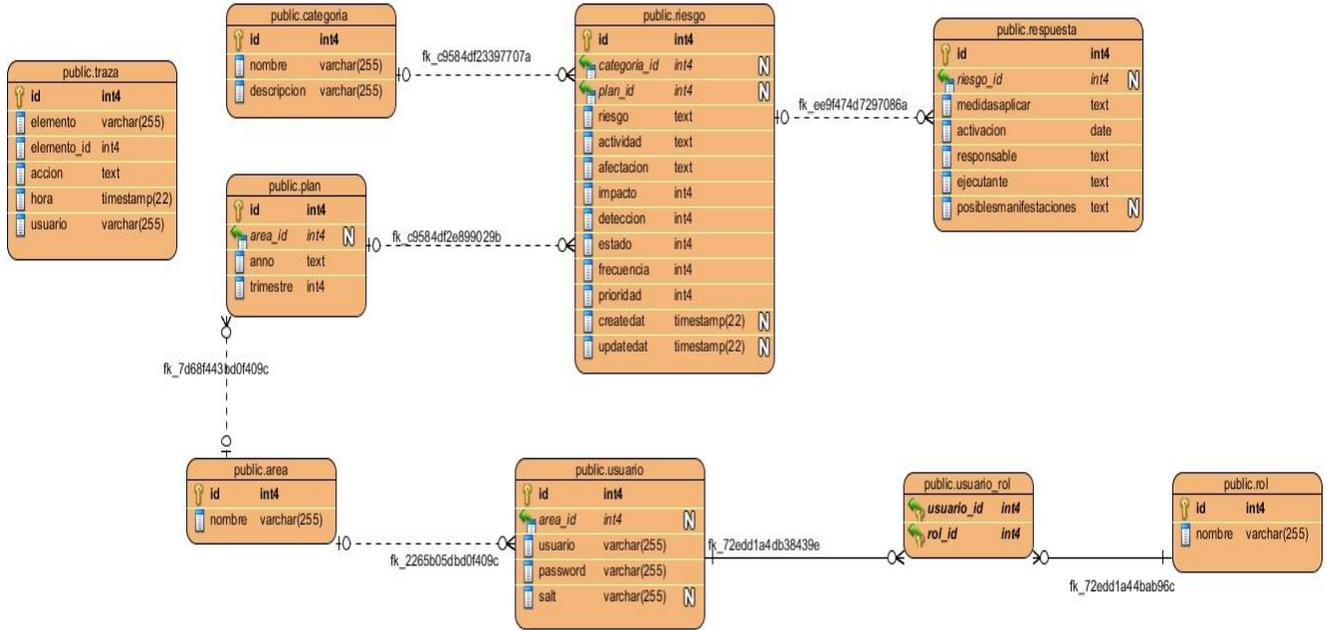
indexAction	Traza, TrazaType
createAction	
createCreateForm	
newAction	
showAction	
editAction	
createEditForm	
updateAction	
deleteAction	
createDeleteForm	

Tabla 22 Tarjeta CRC ClaseUsuarioController

Nombre de la clase: UsuarioController	
Responsabilidades	Colaboraciones
indexAction	Usuario, UsuarioType
createAction	
createCreateForm	
newAction	
showAction	
editAction	
createEditForm	
updateAction	
deleteAction	
createDeleteForm	

3.4.1 Diagrama Entidad Relación

El Diagrama de Entidad Relación es un modelo de datos basado en una percepción del mundo real que consiste en un conjunto de objetos básicos llamados entidades y relaciones entre estos objetos, implementándose en forma gráfica a través del Diagrama Entidad Relación.



3.5 Tareas de ingeniería

La metodología XP propone dividir cada Historia de usuarios en tareas de ingeniería con el objetivo de facilitar la implementación a los programadores, estas tareas deben tener una duración de entre uno y cinco días aproximadamente. Las tareas elaboradas por cada Historia de usuarios fueron:

Tabla 23 Tarea # 1 Tabla 33 Tarea # 1

Tarea	
Número: 1	Número de HU: 1
Nombre: Insertar riesgos	
Tipo de tarea: Desarrollo	Estimación: 4 días y medio
Fecha inicio: 16 de febrero del 2015	Fecha fin: 20 de febrero del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	
Descripción: Insertar riesgo	

Tabla 24: Tarea # 2

Tarea	
Número: 2	Número de HU: 2

Nombre: Insertar respuesta del riesgo	
Tipo de tarea: Desarrollo	Estimación: 4 días
Fecha inicio: 20 de febrero del 2015	Fecha fin: 26 de febrero del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	
Descripción: Insertar respuesta del riesgo	

Tabla 25: Tarea # 3

Tarea	
Número: 3	Número de HU: 3
Nombre: Insertar área	
Tipo de tarea: Desarrollo	Estimación: 3 días y medio
Fecha inicio: 26 de febrero del 2015	Fecha fin: 03 de marzo del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	
Descripción: Insertar área	

Tabla 26: Tarea # 4

Tarea	
Número: 4	Número de HU: 4
Nombre: Mostrar gráficas	
Tipo de tarea: Desarrollo	Estimación: 4 días
Fecha inicio: 04 de marzo del 2015	Fecha fin: 09 de marzo del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	
Descripción: Mostrar gráficas	

Tabla 27: Tarea # 5

Tarea	
Número: 5	Número de HU: 5
Nombre: Consultar trazas	
Tipo de tarea: Desarrollo	Estimación: 4 días
Fecha inicio: 10 de marzo del 2015	Fecha fin: 13 de marzo del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	
Descripción: Consultar trazas	

Tabla 28: Tarea # 6

Tarea	
Número: 6	Número de HU: 6
Nombre: Exportar en pdf	
Tipo de tarea: Desarrollo	Estimación: 4 días
Fecha inicio: 16 de marzo del 2015	Fecha fin: 19 de marzo del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	

Descripción: Exportar en pdf

Fueron realizadas un total de 29 Tareas de Ingeniería, en este epígrafe solamente se encuentran 6 las otras están en los anexos del trabajo.

3.6 Pruebas

Las pruebas son un instrumento adecuado para determinar el status de la calidad de un producto de software. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos. En las pruebas se usan casos de prueba, especificados de forma estructurada mediante Técnicas de prueba. El proceso de pruebas, los objetivos, los métodos y las técnicas usadas se describen en el plan de prueba. Las pruebas de software constituyen parte de una de las fases que propone XP. Las pruebas que propone XP dividen el software y el sistema en dos grupos, las pruebas unitarias y las pruebas de aceptación. (B., 2005)

3.6.1 Pruebas unitarias

Las pruebas unitarias o pruebas de caja blanca se basan en realizar un conjunto de pruebas al código del sistema. Estas pruebas aseguran que un determinado módulo o clase (en dependencia del lenguaje de programación que se esté utilizando) cumpla con el comportamiento esperado. Permiten al programador saber si una determinada funcionalidad se puede agregar al sistema existente sin alterar el funcionamiento actual del mismo. Se realizan a los principales algoritmos o procedimientos. (Pressman, 2002)

Las pruebas unitarias son una de las piedras angulares de XP. Todos los módulos del sistema deben pasar las pruebas unitarias antes de ser liberados o publicados. Por otra parte, las pruebas deben ser definidas antes de realizar el código (*Test-driven programming*). Que todo código liberado pase correctamente las pruebas unitarias es lo que habilita la propiedad colectiva del código. En este sentido, el sistema y el conjunto de pruebas debe ser guardado junto con el código, para que pueda ser utilizado por otros desarrolladores, en caso de tener que corregir, cambiar o recodificar parte del mismo. Cuando se encuentra un error (*bug*), este debe ser corregido inmediatamente, y se deben tener precauciones para que errores similares no vuelvan a ocurrir. Asimismo, se generan nuevas pruebas para verificar que el error haya sido resuelto. (Joskowicz, 2008)

Uno de los errores que se suele cometer es pensar que se puede dejar la construcción de las pruebas para los últimos meses en la realización de un proyecto. Descubrir todos los errores que pueden aparecer lleva tiempo, y más si se deja la depuración de todos para el final. Las unidades de prueba están directamente relacionadas con el concepto de posesión del código.

Symfony2 ha optado por utilizar el *framework PHPUnit*, que prácticamente se ha convertido en un estándar en el mundo *PHP*. De esta forma, las pruebas unitarias de *Symfony2* combinan la potencia de *PHPUnit* con las utilidades y facilidades proporcionadas por *Symfony2*. (Eguiluz, 2011)

En el sistema desarrollado como resultado de la presente investigación las pruebas unitarias se definen en la carpeta *Test* del *bundle*. Por convención de *Symfony2*, cada prueba unitaria se define en una clase cuyo nombre acaba en *Test*. Se utiliza la misma estructura de directorios del elemento que se quiere probar dentro de la carpeta *Test*.

A continuación se muestra un ejemplo del código de cómo fueron implementadas estas pruebas unitarias.

```
public function testValidarAccionNula() {
    $traza = new Traza();
    $traza->setFecha(new \DateTime());
    $traza->setAccion(null);
    $traza->setUsuario("user");

    $listaErrores = $this->validator->validate($traza);
    $this->assertEquals(1, $listaErrores->count(), 'Error validando la acción, no debe ser nula');
}
```

Figura 1 Implementación de la prueba unitaria del método testValidarAccionNula()

Al ejecutar las pruebas se muestra en consola los resultados de las mismas.

```

<[30;42mOK (21 tests, 21 assertions)<[0m
C:\wamp>php phunit.phar -c www\Tesis\Aplicacion\Hatbe\app www\Tesis\Aplicacion\Hatbe\src\Aplicacion\HatbeBundle\Tests\Entity
PHPUnit 4.1.0 by Sebastian Bergmann.

Configuration read from C:\wamp\www\Tesis\Aplicacion\Hatbe\app\phpunit.xml.dist

.<[41;37mF<[0m.....
Time: 310 ms, Memory: 9.25Mb

There was 1 failure:

1) Application\HatbeBundle\Tests\Entity\HintTest::testValidarHintNulo
Error validando el hint, no debe ser nulo
Failed asserting that 1 matches expected 0.

C:\wamp\www\Tesis\Aplicacion\Hatbe\src\Aplicacion\HatbeBundle\Tests\Entity\HintTest.php:37
<[37;41m                                     <[0m
<[37;41mFAILURES!                               <[0m
<[37;41mTests: 21, Assertions: 21, Failures: 1.<[0m
    
```

Figura 2 Resultado de ejecutar una prueba unitaria

3.6.2 Pruebas de aceptación:

Una prueba de aceptación tiene como propósito demostrar al cliente el cumplimiento de un requisito del software. Estas pruebas son hechas por el Cliente y el Probador. Su objetivo es verificar todo el sistema, o una gran parte de él. Estos tipos de pruebas comprueban la calidad de aspectos como: que la aplicación funcione correctamente, que cargue correctamente, la actuación del sistema, su compatibilidad, que no haya problemas con su instalación, que no existan agujeros en su seguridad, entre otros. Se le puede aplicar al sistema, lo mismo, una vez que todo el diseño esté terminado. (Patricio Letelier, 2007) A continuación aparecen los casos de pruebas de aceptación referentes a la propuesta de solución.

Tabla 29 Modelo propuesto para una prueba de aceptación

Caso de prueba	
Número: número que identifica el caso de prueba.	Número HU: número de historia de usuario correspondiente al caso de prueba.
Nombre: nombre de la HU a la que pertenece el caso de prueba.	
Condiciones de ejecución: condiciones necesarias para ejecutar la prueba.	
Pasos de ejecución/entradas: valores de entrada.	

Resultado esperado: salida de la ejecución.

Para la validación de la solución implementada se definen los siguientes casos de prueba de aceptación:

Tabla 30 Caso de Prueba # 1

Caso de prueba	
Número: 1	Número :Historia de usuario:1
Nombre: Insertar riesgo	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Responsable de área, Vice-Decano o Decano.	
Pasos de ejecución/entrada: Entrada: La entrada representa los datos del riesgo que el usuario debe introducir para ser insertado en la base de datos. Pasos de ejecución: Para insertar un riesgo se debe ir al menú y dar clic en la opción gestión de riesgo, seguidamente se mostrará una lista de riesgo con los riesgos ya insertados y se seleccionará la opción "Insertar riesgo". Se va a mostrar una ventana en la cual se debe introducir los datos del riesgo. Una vez introducidos los datos se da clic en el botón "Guardar" y seguidamente se le muestra el riesgo añadido en la lista de riesgo.	
Resultado esperado: el riesgo está insertado correctamente	

Tabla 31 Caso de Prueba # 2

Caso de prueba	
Número: 2	Número :Historia de usuario: 2
Nombre: Insertar respuesta del riesgo	

Condiciones de ejecución: El usuario debe estar autenticado con el rol Responsable de área, Vice-Decano o Decano.
<p>Pasos de ejecución/entrada.</p> <p>Pasos de ejecución: Para insertar respuesta de un riesgo se debe ir al menú y dar clic en la opción gestión de riesgo, seguidamente se mostrará una lista de riesgo con los riesgos ya insertados y se seleccionará la opción “Ver”. Se va a mostrar una ventana con los detalles del riesgo con una opción “insertar respuesta”. Una vez introducidos los datos se da clic en el botón “Guardar” y seguidamente se muestra el riesgo con su respuesta.</p>
Resultado esperado: La respuesta ha sido insertada correctamente

Tabla 32 Caso de Prueba # 3

Caso de prueba	
Número: 3	Número :Historia de usuario: 3
Nombre: Insertar área	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Responsable de área, Vice-Decano, o Decano.	
<p>Entrada/ Pasos de ejecución:</p> <p>Pasos de ejecución: Para añadir una nueva área se debe ir al menú principal y dar clic en la opción “Áreas” seguidamente se mostrará un ventana con un listado de las áreas insertadas, luego de estar en esa ventana dar clic en la opción “Insertar área”, y saldrá una vista para que se llenen los datos de la nueva área, y por último se le da clic en crear.</p>	
Resultado esperado: se añadió una nueva área	

Tabla 33 Caso de Prueba # 4

Caso de prueba	
Número: 4	Número :Historia de usuario: 4
Nombre: Mostrar gráficas	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Responsable de área, Vice-Decano o Decano.	
Entrada/ Pasos de ejecución: Pasos de ejecución: Para mostrar las gráficas se debe ir al menú principal y dar clic en la opción “planes” seguidamente se mostrará un ventana con un listado de los planes insertados, luego de estar en esa ventana se debe dar clic en la opción “ver gráficas” el sistema permitirá mostrar un gráfico con la información del estado de los riesgos a través de la clasificación de los mismos.	
Resultado esperado: El gráfico.	

Tabla 34 Caso de Prueba # 5

Caso de prueba	
Número: 5	Número :Historia de usuario: 5
Nombre: Consultar trazas	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Responsable de área, Vice-Decano o Decano.	
Entrada/ Pasos de ejecución: Pasos de ejecución: Para mostrar las trazas se debe ir al menú principal y dar clic en la opción “Trazas” seguidamente se mostrará una ventana con un listado de todas las cosas se ha hecho en el sistema.	
Resultado esperado: Las trazas.	

Tabla 35 Caso de Prueba # 6

Caso de prueba	
Número: 6	Número :Historia de usuario: 6
Nombre: Exportar a pdf	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Responsable de área, Vice-Decano o Decano.	
Entrada/ Pasos de ejecución: Pasos de ejecución: Para exportar a pdf los planes de riesgo se debe ir al menú principal y dar clic en la opción “planes” seguidamente se mostrará un ventana con un listado de los planes insertados, luego de estar en esa ventana se debe dar clic en la opción “Exportar ” el sistema permitirá exportar el plan seleccionado.	
Resultado esperado: Plan exportado	

Tabla 36 Caso de Prueba # 7

Caso de prueba	
Número: 7	Número :Historia de usuario: 7
Nombre: Insertar usuario	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Decano.	
Entrada/ Pasos de ejecución: La entrada consiste en los datos del usuario que se deben introducir para ser insertado en la base de datos. Pasos de ejecución: Para insertar los datos de un usuario debe ir al menú y seleccionar la opción “Usuarios”. En la ventana que se va a mostrar debe presionar en el botón “Nuevo”. Una vez introducidos los datos se presiona en el	

botón “Guardar” y seguido se le muestra el usuario añadido.
Resultado esperado: El usuario es adicionado correctamente

Tabla 37 Caso de Prueba # 8

Caso de prueba	
Número: 8	Número: Historia de Usuario: 8
Nombre: Buscar	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Responsable de área, Vice-Decano, o Decano.	
Entrada/ Pasos de ejecución: Pasos de ejecución: Para realizar la b debe ir búsqueda se debe ir al menú principal y dar clic en la opción “Reportes” seguidamente se mostrará una ventana con un conjunto de acciones que permitirán realizar la búsqueda.	
Resultado esperado: Una búsqueda	

Tabla 38 Caso de Prueba # 9

Caso de prueba	
Número: 9	Número: Historia de usuario: 9
Nombre: Insertar Categoría	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Responsable de área, Vice-Decano, o Decano.	
Entrada/ Pasos de ejecución: Pasos de ejecución: Para insertar una categoría se debe ir al menú principal y dar clic en la opción “Categoríareporte” seguidamente se mostrará una vista con las opciones para que sean llenadas y luego se le da clic en insertar y saldrá un	

mensaje diciendo que la categoría ha sido insertada correctamente.
Resultado esperado: Un listado de planes

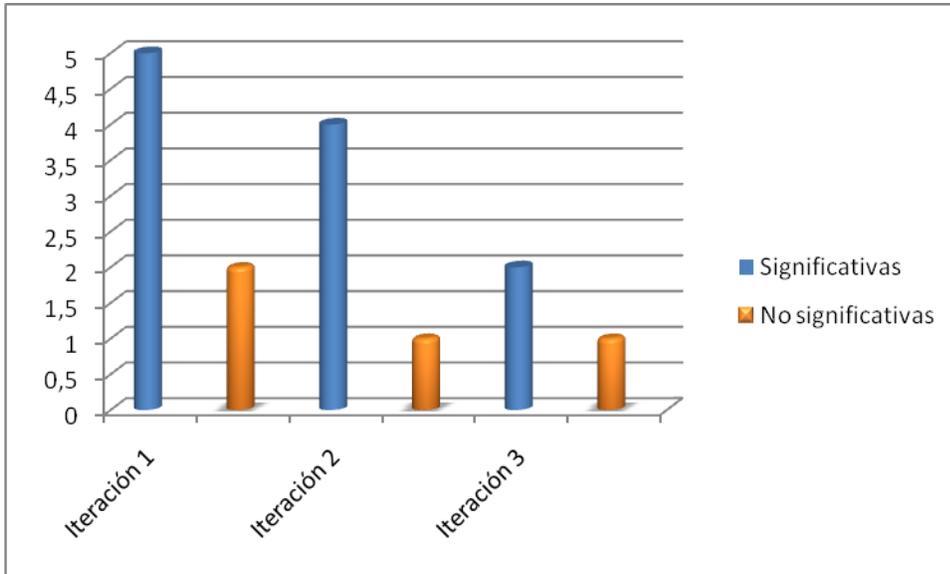
Tabla 39 Caso de Prueba # 10

Caso de prueba	
Número: 10	Número :Historia de usuario: 10
Nombre: Mostrar reportes	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Responsable de área, Vice-Decano, o Decano.	
Entrada/ Pasos de ejecución: Pasos de ejecución: para mostrar un reporte se debe ir al menú principal y dar clic en Reportes, saldrá una vista con las opciones que desee para que pueda realizar el reporte	
Resultado esperado: Un reporte	

Fueron realizadas un total de 29 Casos de pruebas, en este epígrafe solamente se encuentran 10 las otras están en los anexos del trabajo.

3.7 Análisis de los resultados de las pruebas

Al concluir cada una de las iteraciones planificadas para el desarrollo de la propuesta de solución, fueron realizadas las pruebas pertinentes para realizar las entregas pactadas con el cliente. Se detectaron quince no conformidades clasificadas en significativas y no significativas. Entre las no significativas se detectaron cuatro, de las cuales tres fueron errores ortográficos y uno de interfaz. Como parte de las no conformidades significativas se encontraron tres de funciones incorrectas y ocho errores de validación para un total de once. En el siguiente gráfico se muestra un resumen de lo explicado anteriormente.



Todas las no conformidades fueron resueltas lo cual valida en cierto grado la calidad de la propuesta de solución, logrando una mayor satisfacción por parte del cliente.

Conclusiones

En el desarrollo del capítulo se describe la arquitectura que presenta la propuesta de solución, que no es más que la MVC implementada por el *framework Symfony*. Al mismo tiempo se presenta un estudio de los patrones de diseño implementados por el *framework Symfony*, se presentan las tareas de ingenierías necesarias para desarrollar cada historia de usuario. Además se concibieron los artefactos para llevar a cabo la implementación de la propuesta de solución, tal es el caso de las tarjetas CRC y el Diagrama Entidad Relación. Se plantea la estrategia de pruebas seguida por el equipo de desarrollo y los resultados de las pruebas, las cuales acceden poner en manos del cliente un software con menor probabilidad de fallo.

Conclusiones Generales

La investigación realizada posibilita declarar los siguientes aspectos conclusivos:

- El estudio teórico realizado permitió profundizar en los fundamentos científicos del objeto de la investigación y asumir por las autoras el referente principal conceptual y teórico que sustenta la investigación.
- El sistema elaborado para la gestión de riesgos en el escenario de la Facultad 4 de la Universidad de las Ciencias Informáticas, constituye un aporte con valor agregado por su nivel de usabilidad ante la necesidad de esa área de agilizar el mencionado proceso.
- A través del estudio del estado del arte se especificaron las tecnologías que apuntaron la solución de la problemática de la investigación, tales fueron: como metodología de

desarrollo se aplicó XP, UML para el lenguaje de modelado y Visual Paradigm 8.0 como herramienta CASE, las herramientas que se utilizaron en el desarrollo del sistema son: PHP 5.4.12 como lenguajes de programación, *Netbeans* 8.0 como IDE de desarrollo, el *framework* de desarrollo del lado del servidor *Symfony* 2.6.5 y PHPUnit 4.1.0 , el sistema gestor de base de datos MySQL 5.6.12, como servidor *web* Apache 2.4.4.

- La etapa de validación permitió comprobar la eficacia del proceso de desarrollo desplegado a través del funcionamiento de la solución.

Recomendaciones

- Se recomienda que el sistema elaborado sea utilizado por otras áreas de la Universidad con características similares a la Facultad 4 o ser readecuado a la medida de los intereses y necesidades de otros clientes dentro de la propia institución.
- Es sugerente el estudio y uso del sistema que aporta la investigación por parte de la Dirección de Supervisión y Control de la Universidad, al efecto del significado que posee para la gestión del proceso que atiende esa estructura institucional.

Referencias Bibliográficas

Adriana, Venete. 2010. *Introduccion a los patrones de Arquitectura.* Ingeniería de software, Universidad Jaume. 2010.

Alberts, A. 2006. *advanced Risk Analysis for High-Performing Organizations.* SEI. . 2006.

Alvarez, Miguel Angel. 2011. *Manual de jQuery.* . 2011.

2011. *Manual de CSS 3.* 2011.

Alvarez, Rubén. 2001. Lenguaje del lado del servidor o cliente. [En línea] 01 de 01 de 2001. [Citado el: 04 de 03 de 2015.] <http://www.desarrolloweb.com/articulos/239.php>.

Anderson. 2011. *Modelos de Negocio: Propuesta de un Marco Conceptual para Centros de Productividad*. Facultad de Ciencias Económicas. Universidad nacional de Colombia : s.n., 2011.

B., Alexander Oré. 2005. *Patrones del "Gang of Four"*. Facultad de informática, Universidad Politécnica de Madrid. Madrid : s.n., 2005.

Báez, M. P. 2002. La industria del Software, una oportunidad para México. [En línea] 2002. http://www.chfcanada.coop/eng/pdf/FedConf_2008/3B%20Risk%20management%20model%20for%20your%20federation.pdf.

Beck, K. 1999. *Extreme Programming Explained*. s.l. : ISBN 0201616416 , 1999.

Beck, Kent. 1999. *Extreme Programming Explained*. Addison-Wesley : s.n., 1999. ISBN 978-0321278654.

Bergmann, Sebastian. 2009. Sitio oficial de PHPUnit. [En línea] 14 de 05 de 2009. [Citado el: 21 de 03 de 2015.] <http://www.phpunit.de>.

Bleger, Jose. 2012. [En línea] 2012. [Citado el: 24 de 02 de 2015.] <http://html.rincondelvago.com/la-entrevista.html>.

Brandon, Michael. 2013. Ruby on Rails desde cero:Primeros pasos. [En línea] 2013.

Canales, Roberto. 2008. *Arquitectura SOA e Integración de aplicaciones*. . 2008.

Cano, R. 2008. *Gestión de los riesgos en el Proyecto "A Jugar"*. Cuba : s.n., 2008.

Canós J, Letelier P. 2003. *Metodologías Ágiles en el Desarrollo de Software*. Alicante,España : s.n., 2003.

CodeIgniter. 2013. Guía del Usuario de CodeIgniter. [En línea] 2.1.4. , 2013. [Citado el: 20 de 03 de 2015.] [http://escodeigniter.com/guia_usuario/..](http://escodeigniter.com/guia_usuario/)

Connolly, Thomas M. y Begg, Carolyn E. 2005. *Sistemas de bases de datos. Un enfoque práctico*. 2005.

Crusat, Marc Vaquer. 2004. *Zend Studio: el entorno de desarrollo para PHP*. 2004.

2011. Desarrollo Sostenible. [En línea] 2011. [Citado el: 03 de 03 de 2015.] http://www.desenvolupamentsostenible.org/index.php?option=com_content&view=article&id=269&Itemid=302&lang=es.

Domínguez-Dorado, M. 2004. *Introducción a las aplicaciones Web con ASP e IIS*. España : Iberprensa (Madrid), 2004.

Eguíluz Pérez, Javier. 2009. *Introducción a JavaScript*. 2009.

Eguiluz, Javier. 2011. *Desarrollo web ágil con Symfony2*. 2011.

2011. *Desarrollo web ágil con Symfony2*. 2011.

Entornos de Desarrollo Integrado. . [En línea] [Citado el: 15 de 01 de 2015.] <http://petra.euitio.uniovi.es/~i1667065/HD/documentos/Entornos%20de%20Desarrollo%20Integrado.pd..>

Escribano, G. F. 2002. *eXtreme Programming / Programación Extrema*. 2002.

F. Barcenás, Anorvis Aleman. 2009. *La toma de decisiones*. Universidad de Córdoba. Colombia : s.n., 2009.

Fernández Y., Díaz Y. 2012. *Patrón Modelo-Vista-Controlador*. 2012. issn 1729 3804.

Fernando García. 2013. Entornos de Desarrollo Integrado. [En línea] 25 de 01 de 2013. [Citado el: 12 de 02 de 2015.] <http://petra.euitio.uniovi.es>.

Gaceta. 2011.. *Gaceta Oficial de la República de Cuba, Contraloría General de la República Resolución No. 60/11*. 2011.

García, W. 2006. *Gestión de Proyectos Informáticos. Administración de Proyectos*. . 2006.

Gestión de riesgos en ingeniería del software. Maguerit. 2008.. 2008.

Group, Postgres Global Development. 1999. Manual del usuario de PostgreSQL. 1999.

Iglesias, Yudalys Junco. 2009. *Propuesta de una guía para la gestión de riesgos en el polo de Gestión Universitaria de la Facultad 1*. La Habana: s.n., 2009.

José A García. 2003. La toma de decisiones. [En línea] 2003. [Citado el: 10 de 03 de 2015.] <https://toma%20de%20decisiones/TOMA%20DE%20DECISIONES.htm>.

Joskowicz, Jose. 2008. *Reglas y Prácticas en eXtreme Programming*. España: s.n., 2008.

2008. *Reglas y Prácticas en eXtreme Programming*. España: s.n., 2008.

Labs, Sension. 2009. *Doctrine ORM for PHP.* 2009.

Larman, Craig. 2004. *Applying UML and patterns: an introduction to object-oriented analysis and design and iterative development.* New Jersey : s.n., 2004.

2013. Lenguaje de programación. [En línea] 2013. [Citado el: 01 de 03 de 2015.] http://escritorioalumnos.educ.ar/datos/recursos/lenguajes_de_programacion_4.pdf.

Lerner, Reuven M. 2012. *Twitter bootstrap.* *Linux Journal.* s.l. : vol 2012, no 218, 2012.

Louden, Kenneth C. 2004.. *Lenguajes de programación: Principios y práctica.* . s.l. : S.I: Cengag e Learning Editores , 2004.

M. Corona. 2012. Toma de decisiones. [En línea] 24 de 12 de 2012. [Citado el: 30 de 03 de 2015.] <https://CONCEPTOS%20TOMA%20DE%20DECISIONES.htm>.

Martínez P. 2006. La Importancia de la toma de decisiones. 2006.

Mauri Pérez, Yolanda y de la Soledad García, Ernesto. 2012. *Desarrollo del sistema de gestión de contenidos de la colección El Navegante en su versión multiplataforma.* La Habana : s.n., 2012.

Mela, Marta. 2011. Que son las Tic y para que sirven. [En línea] 13 de 04 de 2011. [Citado el: 20 de 02 de 2015.] <http://noticias.iberestudios.com/%C2%BFque-son-las-tic-y-para-que-sirven>.

Microsoft. 2008. Rol usuario del sistema. [En línea] 2008. <https://technet.microsoft.com/es-es/library/ms155949%28v=sql.105%29.aspx>.

2013. Sitio oficial de Microsoft Office Online. [En línea] 2013. [Citado el: 04 de 03 de 2015.] <http://office.microsoft.com>.

Microsystems, Sun. 2010. MySQL 5.0 Reference Manual. 2010.

Nedelcu, Clément. 2013. *Nginx HTTP Server.* Francia : s.n., 2013.

Oracle Corporation, Sun Microsystem. 2000. Sitio oficial *NetBeans*. [En línea] 12 de 2000. [Citado el: 14 de 02 de 2015.] <https://netbeans.org/community/releases/74>.

Patricio Letelier. 2007. Pruebas de aceptación como conductor del proceso de software. [En línea] 05 de 05 de 2007. <http://in2test.lsi.uniovi.es/repris/actividades/TestingJTS2007.pdf>.

Potencier, Fabien. 2011. *What is Symfony2?* . 2011.

2010. Sitio oficial de Symfony. [En línea] 2010. [Citado el: 02 de 25 de 2015.] <http://www.symfony.es>.

Pressman, R. S. 2002. *Ingeniería del Software. Un enfoque práctico.* . s.l. : 5ta parte Vol. 1, 2002.

Pressman, R. S. 2002. *Ingeniería del Software. Un enfoque práctico.* 2002. Vol. 5ta edición. ISBN: 8448132149.

R, Alvarez. 2010. desarrollo-de-software-tarjetas-crc/. [En línea] 01 de 10 de 2010. [Citado el: 05 de 04 de 2015.] <https://jummp.wordpress.com>.

Rodríguez M., Ordóñez M. 2007. *La metodología XP aplicable al desarrollo del software educativo en Cuba.* 2007.

Rodríguez, Gerlin Vidal. 2008. *Gestión de Seguridad Informática, Sistema para la gestión de riesgos.* Universidad de las ciencias Informáticas. La Habana : s.n., 2008.

Ruiz, Victor M. 2009. La biblia del servidor Apache. 2009.

Sæther B, Stig. 2001. *Manual de PHP.* s.l. : Free Software Foundation, 2001.

Sampieri, R. 2006. *Metodología de la investigación.* s.l. : cuarta edición, 2006. ISBN 970-10-5753-8.

SEI. 2004. *Software Engineering Institute.* 2004.

servidor, Lenguajes del lado del. 2005. Lenguajes del lado del servidor. [En línea] 2005. [Citado el: 01 de 03 de 2015.] http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html.

Sierra, María. 2010. Trabajando con Visual Paradigm for UML. 2010.

Sitio oficial *NetBeans*. [En línea] [Citado el: 21 de 03 de 2015.] [https://netbeans.org/community/releases/74/..](https://netbeans.org/community/releases/74/)

Yasniel. 2011. [En línea] 2011. [Citado el: 25 de 03 de 2015.] <http://openupeaojmp.blogspot.com/2013/09/metodologia-open-up.html>.

Bibliografía

Alberts, A. J.D.A.C.J. advanced Risk Analysis for High-Performing Organizations. SEI. 2006. P.

SEI. 2004. Software Engineering Institute. 2004.

Alvarez, Miguel Angel .Manual de CSS 3. 2011.

Alvarez, Miguel Angel .Manual de jQuery.

Beck, Kent y Foweler, Martin. 2000. Planning Xtreme Programming. 1ra. Edición. s.l. : Addison Wesley, 2000.

Canós, Joseph; Letelier, Patricio y Penadés, M^a Carmen. Metodologías Ágiles en el desarrollo de Software. Universidad Politécnica de Valencia, Valencia, 2003.

Crusat, Marc Vaquer. Zend Studio: el entorno de desarrollo para PHP. 2004

Eguíluz Pérez, Javier. Introducción a JavaScript. 2009.

Eguiluz, Javier .Desarrollo *web* ágil con *Symfony2*. 2011.

Eguíluz Pérez, Javier. Introducción a XHTML. Extraído el, 2007, vol. 24.

Gaceta Oficial de la República de Cuba, Contraloría General de la República Resolución No. 60/11,2011.

Group, Postgres Global Development .Manual del usuario de PostgreSQL. 1999.

Joskowicz, José. Reglas y Prácticas en eXtreme Programming. 2008.

J. Kabir, Mohamemed. La biblia del Servidor Apache.

Lounden, Kenneth C .Lenguajes de programación: Principios y práctica. . S.l: Cengag e Learning Editores, 2004.

Larman, Craig. UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado. Pearson Educación, 2003. ISBN 9788420534381.

Pressman, R. S. Ingeniería del Software. Un enfoque práctico. 2002. 5ta parte Vol.

Sampieri, R. Metodología de la investigación, cuarta edición, 2006, ISBN 970-10-5753-8.

Sun Microsystems .MySQL 5.0 Reference Manual. 2010. Sun Microsystems .MySQL 5.0 Reference Manual. 2010.

Anexos

Anexo 1

Entrevista a responsables de áreas de la Facultad 4.

Objetivo: Conocer cómo se garantiza la seguridad de la información y la elaboración del plan de riesgo de la facultad.

Preguntas:

1. ¿Cómo usted garantiza la seguridad de la información en su área? ¿Qué personas tienen acceso a esta información?
2. ¿Cómo se elabora el plan de gestión de riesgos del área?
3. ¿Podrá un software contribuir a garantizar la seguridad de la información y consecuentemente a mejorar la elaboración del plan de riesgos antes mencionado? Argumente su consideración.

Anexo 2. Diferencias entre las metodologías ágiles y tradicionales.

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Tabla 40 Diferencias entre las metodologías ágiles y tradicionales.

Anexo 3. Historias de usuario

Tabla 41 HU- Listar riesgo

Historia de usuario	
Número: 11	Nombre: Listar riesgos
Usuario: Responsable de área, Vicedecano administrativo y Decano.	
Prioridad: Media	Riesgo: Medio
Estimación: 0.3	Iteración: 2

Descripción: Al hacer clic en la opción Gestión de Riesgos se muestra el listado de riesgos creados en el sistema y la descripción de cada uno.

Observación:

Tabla 42 HU- Ver riesgo

Historia de usuario	
Número: 12	Nombre: Ver riesgo
Usuario: Responsable de área, Vicedecano administrativo y Decano.	
Prioridad: Media	Riesgo: Medio
Estimación: 0.2	Iteración: 2
Descripción: Se selecciona un riesgo de la lista y se muestra una vista con su descripción.	
Observación:	

Tabla 43 HU- Editar riesgo

Historia de usuario	
Número: 13	Nombre: Editar riesgo
Usuario: Responsable de área, Vicedecano administrativo y Decano.	
Prioridad: Media	Riesgo: Medio
Estimación: 0.4	Iteración: 2
Descripción: Se selecciona de la lista, el riesgo que se desea editar y se muestran los datos del riesgo: <ul style="list-style-type: none"> - Riesgo - Afectación - Impacto - Detección - Frecuencia - Categoría 	
Observación:	

Tabla 44 HU- Incluir riesgo en el plan de la Facultad

Historia de usuario	
Número: 14	Nombre: Incluir riesgo en el plan de la Facultad
Usuario: Vicedecano administrativo y Decano.	
Prioridad: Media	Riesgo: Medio
Estimación: 0.2	Iteración: 2
Descripción: Al hacer clic en la opción incluir riesgo automáticamente lo incluirá en el plan de la Facultad.	
Observación:	

Tabla 45 HU- Listar plan

Historia de usuario	
Número: 15	Nombre: Listar Plan
Usuario: Responsable de área ,Vicedecano administrativo y Decano,Invitado	
Prioridad: Media	Riesgo: Alto
Estimación: 0.4	Iteración: 2
Descripción: Al hacer clic en Planes, el usuario visualizará la lista de todos los planes de su área.	
Observación:	

Tabla 46 HU- Mostrar riesgos de un plan

Historia de usuario	
Número: 16	Nombre: Mostrar riesgos de un plan
Usuario: Responsable de área, Vicedecano administrativo y Decano.	
Prioridad: Media	Riesgo: Alto
Estimación: 0.3	Iteración: 2
Descripción: Al hacer clic en Ver riesgos, se mostrarán los riesgos del plan escogido.	
Observación:	

Tabla 47 HU- Mostrar respuestas de un plan

Historia de usuario	
Número: 17	Nombre: Mostrar respuestas de un plan
Usuario: Responsable de área, Vicedecano administrativo y Decano.	
Prioridad: Media	Riesgo: Alto
Estimación: 0.3	Iteración: 1
Descripción: Al hacer clic en Ver respuesta, se mostrarán todas las respuestas de los riesgos del plan escogido.	
Observación:	

Tabla 48 HU- Modificar respuesta del riesgo

Historia de usuario	
Número: 18	Nombre: Modificar respuesta del riesgo
Usuario: Responsable de área, Vicedecano administrativo y Decano.	
Prioridad: Media	Riesgo: Medio
Estimación: 0.4	Iteración: 2
Descripción: Al hacer clic en modificar, se selecciona la respuesta que se desea modificar y aparecen los datos para ser editados:	

<ul style="list-style-type: none"> - Medidas a aplicar - Activación - Responsable - Ejecutante - Posibles manifestaciones
Al concluir se muestra un mensaje confirmando que fue modificado correctamente.
Observación:

Tabla 49 HU- Eliminar respuesta del riesgo

Historia de usuario	
Número: 19	Nombre: Eliminar respuesta del riesgo
Usuario: Responsable de área, Vicedecano administrativo y Decano.	
Prioridad: Media	Riesgo: Medio
Estimación: 0.1	Iteración: 2
Descripción: Al dar clic en eliminar respuesta se muestra un mensaje informando que la respuesta se ha eliminado correctamente.	
Observación:	

Tabla 50 HU- Mostrar área

Historia de usuario	
Número: 20	Nombre: Mostrar área
Usuario: Decano.	
Prioridad: Media	Riesgo: Media
Estimación: 0.5	Iteración: 2
Descripción: Cuando se quiere mostrar un área se debe dar clic en el menú principal Áreas, se abre una vista con los datos del área	
<ul style="list-style-type: none"> - nombre - id - si esta activa 	
Observación:	

Tabla 51 HU- Listar área

Historia de usuario	
Número: 21	Nombre: Listar área
Usuario: Decano.	
Prioridad: Media	Riesgo: Alto
Estimación: 0.5	Iteración: 2

Descripción: Al hacer clic en Áreas, se abre una vista con un listado de todas áreas y su descripción.
Observación:

Tabla 52 HU- Editar área

Historia de usuario	
Número: 22	Nombre: Editar área
Usuario: Decano.	
Prioridad: Media	Riesgo: Alto
Estimación: 0.4	Iteración: 3
Descripción: Al dar clic en Editar, saldrá una ventana con los datos del área: <ul style="list-style-type: none"> - - Nombre - - Activa 	
Al ser editados los datos se dará clic en Actualizar y quedarán modificados los datos.	
Observación:	

Tabla 53 HU- Eliminar área

Historia de usuario	
Número: 23	Nombre: Eliminar área
Usuario: Decano	
Prioridad: Media	Riesgo: Media
Estimación: 0.4	Estimación: 3
Descripción: Al dar clic en Eliminar, se mostrará un mensaje informando que el área se ha eliminado correctamente.	
Observación:	

Tabla 54 HU- Listar usuario

Historia de usuario	
Número: 24	Nombre: Listar usuario
Usuario: Vicedecano administrativo y Decano.	
Prioridad: Media	Riesgo: Alto
Estimación: 0.6	Iteración: 3
Descripción: Al dar clic en Usuarios, se abre una vista con un listado de todos los usuarios que han sido adicionados.	
Observación:	

Tabla 55 HU- Mostrar usuario

Historia de usuario	
Número: 25	Nombre: Mostrar usuario

Usuario: Vicedecano administrativo y Decano.	
Prioridad: Media	Riesgo: Alto
Estimación: 0.6	Iteración: 3
Descripción: Al dar clic en Ver, se muestran los datos del usuario: <ul style="list-style-type: none"> - Id - Usuario - Correo - Área 	
Además de mostrar la opción de Eliminar.	
Observación:	

Tabla 56 HU- Eliminar usuario

Historia de usuario	
Número: 26	Nombre: Eliminar usuario
Usuario: Vicedecano administrativo y Decano.	
Prioridad: Media	Riesgo: Alto
Estimación: 0.5	Iteración: 3
Descripción: Al dar clic en Eliminar, automáticamente se elimina el usuario.	
Observación:	

Tabla 57 HU- Modificar categoría

Historia de usuario	
Número: 27	Nombre: Modificar categoría
Usuario: Decano.	
Prioridad: Media	Riesgo: Alto
Estimación: 0.5	Iteración: 3
Descripción: Al dar clic en Editar, se muestra una vista solicitando los datos necesarios para la modificación: <ul style="list-style-type: none"> - Nombre - Descripción - Activa 	
Al concluir se muestra un mensaje confirmando que fue modificado.	
Observación:	

Tabla 58 HU- Mostrar categoría

Historia de usuario

Número: 28	Nombre: Mostrar categoría
Usurious: Decano.	
Prioridad: Media	Riesgo: Alto
Estimación: 0.4	Iteración: 3
Descripción: Al dar clic en Ver, se visualizará la categoría con su descripción y brindará la opción de eliminarla.	
Observación:	

Tabla 59 HU- Eliminar categoría

Historia de usuario	
Número: 29	Nombre: Eliminar categoría
Usurious: Decano.	
Prioridad: Media	Riesgo: Alto
Estimación: 0.4	Iteración: 3
Descripción: Al dar clic en eliminar clasificación se muestra un mensaje informando que la categoría se ha eliminado correctamente.	
Observación:	

Anexo 4. Tarjetas CRC

Tabla 60 tarjeta CRC Clase Área

Nombre de la clase: Área	
Responsabilidades	Colaboraciones
getId setNombre getNombre	

Tabla 61

Tarjeta CRC

Clase

Categoría

Nombre de la clase: Categoría	
Responsabilidades	Colaboraciones
getId setNombre getNombre setDescripcion getDescripcion	

Tabla 62

Tarjeta CRC Clase Plan

Nombre de la clase: Plan	
Responsabilidades	Colaboraciones
__construct addRiesgo removeRiesgo getId setArea getArea setRiesgos getRiesgos setAnno getAnno setTrimestre getTrimestre setPrevencion getPrevencion	Riesgo, Área, PlanRepository

Tabla 63 Tarjeta CRC Clase PlanRepository

Nombre de la clase: PlanRepository	
Responsabilidades	Colaboraciones
findConRespuestasCategorías	

Tabla 64 Tarjeta CRC Clase Respuesta

Nombre de la clase: Respuesta	
Responsabilidades	Colaboraciones
getId setMedidasAplicar getMedidasAplicar setActivacion getActivacion setResponsable getResponsable setEjecutante getEjecutante setPosiblesManifestaciones getPosiblesManifestaciones setRiesgo getRiesgo	RespuestaRepository, Riesgo

Tabla 65 Tarjeta CRC Clase RespuestaRepository

Nombre de la clase: RespuestaRepository	
Responsabilidades	Colaboraciones
findbyPlan	

Tabla 66 Tarjeta CRC Clase Riesgo

Nombre de la clase: Riesgo	
Responsabilidades	Colaboraciones
getId setRiesgo getRiesgo setActividad getActividad setAfectacion getAfectacion setImpacto getImpacto setDeteccion getDeteccion setEstado getEstado setFrecuencia getFrecuencia setPrioridad getPrioridad setCategoria getCategoria setPlan getPlan __toString setRespuesta getRespuesta setCreatedAt getCreatedAt setUpdatedAt getUpdatedAt __construct	Categoría, Plan, RiesgoRepository, Respuesta

Tabla 67 Tarjeta CRC Clase RiesgoRepository

Nombre de la clase: RiesgoRepository	
Responsabilidades	Colaboraciones
findCantidadConRespuesta findCantidad findPorEstado findConRespuestasCategorias	

Tabla 68 Tarjeta CRC Clase Rol

Nombre de la clase: Rol	
Responsabilidades	Colaboraciones
getId setNombre getNombre getRole __toString	

Tabla 69 Tarjeta CRC Clase Traza

Nombre de la clase: Traza	
Responsabilidades	Colaboraciones
getId setElemento getElemento setElementold getElementold setAccion getAccion setHora getHora setUsuario getUsuario	

Tabla 70 Tarjeta CRC Clase Usuario

Nombre de la clase: Usuario	
Responsabilidades	Colaboraciones
__construct getId setUsuario getUsuario setPassword getPassword setSalt getSalt addUsuarioRol removeUsuarioRol getUsuarioRol eraseCredentials getRoles getUsername serialize unserialize __toString setArea getArea	Rol, Área

Tabla 71 Tarjeta CRC Clase AreaType

Nombre de la clase: ÁreaType	
Responsabilidades	Colaboraciones
buildForm setDefaultOptions getName	Área

Tabla 72 Tarjeta CRC Clase CategoriaType

Nombre de la clase: CategoríaType	
Responsabilidades	Colaboraciones
buildForm setDefaultOptions getName	Categoría

Tabla 73 Tarjeta CRC Clase PlanType

Nombre de la clase: PlanType

	Colaboraciones
Responsabilidades	
buildForm setDefaultOptions getName	Plan

Tabla 74 Tarjeta CRC Clase RespuestaType

Nombre de la clase: RespuestaType	
Responsabilidades	Colaboraciones
buildForm setDefaultOptions getName	Respuesta

Tabla 75 Tarjeta CRC Clase RiesgoType

Nombre de la clase: RiesgoType	
Responsabilidades	Colaboraciones
buildForm setDefaultOptions getName	Riesgo

Tabla 76 Tarjeta CRC Clase TrazaType

Nombre de la clase: TrazaType	
Responsabilidades	Colaboraciones
buildForm setDefaultOptions getName	Traza

Tabla 77 Tarjeta CRC Clase UsuarioType

Nombre de la clase: UsuarioType	
Responsabilidades	Colaboraciones
buildForm setDefaultOptions getName	Área, Usuario

Tabla 78 Tarjeta CRC Clase RiesgoVoter

Nombre de la clase: RiesgoVoter	
Responsabilidades	Colaboraciones
__construct getSupportedClasses getSupportedAttributes isGranted	Riesgo

Tabla 79 Tarjeta CRC Clase ÁreaControllerTest

Nombre de la clase: ÁreaControllerTest	
Responsabilidades	Colaboraciones
testCompleteScenario	

Tabla 80 Tarjeta CRC Clase CategoriaControllerTest

Nombre de la clase: CategoriaControllerTest	
Responsabilidades	Colaboraciones
testCompleteScenario	

Tabla 81 Tarjeta CRC Clase DefaulControllerTest

Nombre de la clase: DefaulControllerTest	
Responsabilidades	Colaboraciones
testIndex	

Tabla 82 tarjeta CRC Clase PlanControllerTest

Nombre de la clase: PlanControllerTest	
Responsabilidades	Colaboraciones
testCompleteScenario	

Tabla 83 Tarjeta CRC Clase RespuestaControllerTest

Nombre de la clase: RespuestaControllerTest	
Responsabilidades	Colaboraciones
testCompleteScenario	

Tabla 84 Tarjeta CRC Clase RiesgoControllerTest

Nombre de la clase: RiesgoControllerTest	
Responsabilidades	Colaboraciones
testCompleteScenario	

Tabla 85 Tarjeta CRC Clase TrazaControllerTest

Nombre de la clase: TrazaControllerTest	
Responsabilidades	Colaboraciones
testCompleteScenario	

Tabla 86 Tarjeta CRC Clase UsuarioControllerTest

Nombre de la clase: UsuarioControllerTest	
Responsabilidades	Colaboraciones
testCompleteScenario	

Tabla 87 Tarjeta CRC Clase General

Nombre de la clase: General	
Responsabilidades	Colaboraciones
__construct guardarLog	Usuario, Traza

Tabla 88 Tarjeta CRC Clase trimestreExtension

Nombre de la clase: trimestreExtension	
Responsabilidades	Colaboraciones
getName getFilters trimestreString estadoString	

Tabla 89 Tarjeta CRC Clase Util

Nombre de la clase: Util	
Responsabilidades	Colaboraciones
cleanString getSlug getPeriod	

Tabla 90 Tarjeta CRC Clase Configuration

Nombre de la clase: Configuration	
Responsabilidades	Colaboraciones
getConfigTreeBuilder	

Tabla 91 Tarjeta CRC Clase sgrGestionExtension

Nombre de la clase: sgrGestionExtension	
Responsabilidades	Colaboraciones
load	

Anexo 5. Tareas de Ingeniería

Tabla 92: Tarea # 7

Tarea	
Número: 7	Número de HU: 7
Nombre: Insertar usuario	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha inicio: 20 de marzo del 2015	Fecha fin: 25 de marzo del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	
Descripción: Insertar usuario	

Tabla 93: Tarea #8

Tarea	
Número: 8	Número de HU: 8
Nombre: Buscar	
Tipo de tarea: Desarrollo	Estimación: 4 días
Fecha inicio: 25 de marzo del 2015	Fecha fin: 31 de marzo del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	

Descripción: Buscar

Tabla 94: Tarea # 9

Tarea	
Número: 9	Número de HU: 9
Nombre: Insertar categoría	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha inicio: 01 de abril del 2015	Fecha fin: 02 de abril del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	
Descripción: Insertar categoría	

Tabla 95: Tarea # 10

Tarea	
Número: 10	Número de HU: 10
Nombre: Mostrar reportes	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha inicio: 03 de abril del 2015	Fecha fin: 06 de abril del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	
Descripción: Mostrar reportes	

Tabla 96: Tarea # 11

Tarea	
Número: 11	Número de HU: 11
Nombre: Listar riesgos	
Tipo de tarea: Desarrollo	Estimación: 1 día y medio
Fecha inicio: 7 de abril del 2015	Fecha fin: 8 de abril del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	
Descripción: Listar riesgos	

Tabla 97: Tarea # 12

Tarea	
Número: 12	Número de HU: 12
Nombre: Ver riesgo	
Tipo de tarea: Desarrollo	Estimación: 1 día
Fecha inicio: 08 de abril del 2015	Fecha fin: 09 de abril del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	
Descripción: Ver riesgo	

Tabla 98: Tarea # 13

Tarea	
-------	--

Número: 13	Número de HU: 13
Nombre: Editar riesgo	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha inicio: 09 de abril del 2015	Fecha fin: 13 de abril del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	
Descripción: Editar riesgo	

Tabla 99: Tarea # 14

Tarea	
Número: 14	Número de HU: 14
Nombre: Editar riesgo.	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha inicio: 09 de abril del 2015	Fecha fin: 13 de abril del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	
Descripción: Editar riesgo.	

Tabla 100: Tarea # 15

Tarea	
Número: 15	Número de HU: 15
Nombre: Incluir riesgos en el plan	
Tipo de tarea: Desarrollo	Estimación: 1 día
Fecha inicio: 13 de abril del 2015	Fecha fin: 14 de abril del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	
Descripción: Incluir riesgos en el plan	

Tabla 101: Tarea # 16

Tarea	
Número: 16	Número de HU: 16
Nombre: Listar plan	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha inicio: 14 de abril del 2015	Fecha fin: 20 de abril del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	
Descripción: Listar plan	

Tabla 102: Tarea # 17

Tarea	
Número: 17	Número de HU: 17
Nombre: Mostrar riesgos de un plan	
Tipo de tarea: Desarrollo	Estimación: 1 día y medio

Fecha inicio: 20 de abril del 2015	Fecha fin: 21 de abril del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	
Descripción: Mostrar riesgos de un plan	

Tabla 103: Tarea # 18

Tarea	
Número: 18	Número de HU: 18
Nombre: Mostrar respuesta de un plan	
Tipo de tarea: Desarrollo	Estimación: 1 día y medio
Fecha inicio: 20 de abril del 2015	Fecha fin: 21 de abril del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	
Descripción: Mostrar respuesta de un plan	

Tabla 104: Tarea # 19

Tarea	
Número: 19	Número de HU: 19
Nombre: Modificar respuesta del riesgo	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha inicio: 25 de abril del 2015	Fecha fin: 28 de abril del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	
Descripción: Modificar respuesta del riesgo	

Tabla 105: Tarea # 20

Tarea	
Número: 20	Número de HU: 20
Nombre: Eliminar respuesta del riesgo	
Tipo de tarea: Desarrollo	Estimación: medio día
Fecha inicio: 28 de abril del 2015	Fecha fin: 28 de abril del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	
Descripción: Eliminar respuesta del riesgo	

Tabla 106: Tarea # 21

Tarea	
Número: 21	Número de HU: 21
Nombre: Mostrar área	
Tipo de tarea: Desarrollo	Estimación: 2 días y medio
Fecha inicio: 29 de abril del 2015	Fecha fin: 01 de mayo del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	
Descripción: Mostrar área	

Tabla 107: Tarea # 22

Tarea	
Número: 22	Número de HU: 22
Nombre: Listar área	
Tipo de tarea: Desarrollo	Estimación: 2 días y medio
Fecha inicio: 01 de mayo del 2015	Fecha fin: 05 de mayo del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	
Descripción: Listar área	

Tabla 108: Tarea # 23

Número: 23	Número de HU: 23
Nombre: Editar área	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha inicio: 06 de mayo del 2015	Fecha fin: 07 de mayo del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	
Descripción: Editar área	

Tabla 109: Tarea # 24

Número: 24	Número de HU: 24
Nombre: Eliminar área	
Tipo de tarea: Desarrollo	Estimación: 4 días
Fecha inicio: 08 de mayo del 2015	Fecha fin: 11 de mayo del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	
Descripción: Eliminar área	

Tabla 110: Tarea # 25

Número: 25	Número de HU: 25
Nombre: Listar usuario	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha inicio: 12 de mayo del 2015	Fecha fin: 14 de mayo del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	
Descripción: Listar usuario	

Tabla 111: Tarea # 26

Número: 26	Número de HU: 26
Nombre: Mostrar usuario	
Tipo de tarea: Desarrollo	Estimación: 3 días

Fecha inicio: 15 de mayo del 2015	Fecha fin: 19 de mayo del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	
Descripción: Mostrar usuario	

Tabla 112: Tarea # 27

Tarea # 27	
Número: 27	Número de HU: 27
Nombre: Eliminar usuario	
Tipo de tarea: Desarrollo	Estimación: 2 días y medio
Fecha inicio: 21 de mayo del 2015	Fecha fin: 25 de mayo del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	
Descripción: Eliminar usuario	

Tabla 113: Tarea # 28

Tarea # 28	
Número: 28	Número de HU: 28
Nombre: Modificar categoría	
Tipo de tarea: Desarrollo	Estimación: 2 días y medio
Fecha inicio: 25 de mayo del 2015	Fecha fin: 27 de mayo del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	
Descripción: Modificar categoría	

Tabla 114: Tarea # 29

Tarea # 29	
Número: 28	Número de HU: 29
Nombre: Mostrar categoría	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha inicio: 28 de mayo del 2015	Fecha fin: 28 de mayo del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	
Descripción: Mostrar categoría	

Tabla 115: Tarea # 30

Tarea	
Número: 30	Número de HU: 30
Nombre: Eliminar categoría	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha inicio: 20 de mayo del 2015	Fecha fin: 22 de mayo del 2015
Programador responsable: Mislaidy Padín Gutierrez, Ailen Lores Delgado	
Descripción: Eliminar categoría	

Anexo 6.Casos de Pruebas

Tabla 116 Caso de Prueba # 11

Caso de prueba	
Número: 11	Número :Historia de usuario: 11
Nombre: Listar riesgos	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Responsable de área, Vice-Decano , Decano o invitado y además debe haber algún riesgo insertado en la base de datos	
Entrada/ Pasos de ejecución: Pasos de ejecución: Para listar los riesgos debe ir al menú y seleccionar la opción “Gestión de riesgo”, seguidamente se mostrará una ventana con una lista de todos los riesgos insertados.	
Resultado esperado: se mostrará un listado de todos los riegos insertados en el sistema.	

Tabla 117 Caso de Prueba # 12

Caso de prueba	
Número: 12	Número :Historia de usuario: 12
Nombre: Ver riesgo	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Responsable de área, Vice-Decano , Decano o invitado y además debe haber algún riesgo insertado en la base de datos	
Entrada/ Pasos de ejecución: Pasos de ejecución: Para ver un riesgo se debe ir al menú y seleccionar la opción “Gestión de riesgo”, seguidamente se mostrará una ventana con una lista de todos los riesgos insertados se da clic en ver y saldrá una vista con la descripción del riesgo.	

Resultado esperado: La descripción del riesgo.

Tabla 118 Caso de Prueba # 13

Caso de prueba	
Número: 13	Número :Historia de Usuario: 13
Nombre: Editar riesgo	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Responsable de área, Decano o Vicedecano y además debe haber algún riesgo insertado en la base de datos	
Entrada/ Pasos de ejecución: Entrada: Los datos del riesgo que el usuario quiere modificar. Pasos de ejecución: Para editar un riesgo se debe ir al menú y seleccionar la opción “Gestión de riesgo”, se da clic en esa opción y saldrá una ventana con una lista de todos los riesgos insertados en la Base de datos, se da clic en “ver” y se mostrará una vista con la descripción del riesgo, se da clic en la opción “editar” y saldrá una ventana donde se les podrá realizar los cambios al riesgo.	
Resultado esperado: El riesgo es modificado correctamente.	

Tabla 119 Caso de Prueba # 14

Caso de prueba	
Número: 14	Número: Historia de usuario: 14
Nombre: Incluir riesgo en el plan	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Responsable de área, Vice-Decano o Decano.	
Pasos de ejecución/entrada.	

Pasos de ejecución: Para incluir un riesgo en el plan se le da clic en la opción "Gestión de riesgos", seguidamente saldrá una vista con un listado de riesgos adicionados y el usuario debe dar clic en la opción "incluir"
Resultado esperado: El riesgo ha sido incluido correctamente

Tabla 120 Caso de Prueba # 15

Caso de prueba	
Número: 15	Número: Historia de usuario: 15
Nombre: Listar plan	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Responsable de área, Vice-Decano, o Decano.	
Entrada/ Pasos de ejecución: Pasos de ejecución: Para Listar plan de riesgo se debe ir al menú principal y dar clic en la opción "planes" seguidamente se mostrará una ventana con un listado de los planes insertados en la Base de datos.	
Resultado esperado: Un listado de planes	

Tabla 121 Caso de Prueba # 16

Caso de prueba	
Número: 16	Número: Historia de usuario: 16
Nombre: Mostrar riesgo de un plan	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Responsable de área, Vice-Decano, o Decano.	
Entrada/ Pasos de ejecución:	

<p>Pasos de ejecución: Para mostrar los riesgo de un plan se debe ir al menú principal y dar clic en la opción “planes” seguidamente se mostrará una ventana con un listado de los planes insertados en la Base de datos, se da clic en mostrar y saldrá una vista con los riesgos de ese plan .</p>
<p>Resultado esperado: Riesgos de un plan</p>

Tabla 122 Caso de Prueba # 17

Caso de prueba	
Número: 17	Número: Historia de usuario: 17
Nombre: Mostrar respuesta de un plan	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Responsable de área, Vice-Decano, o Decano.	
<p>Entrada/ Pasos de ejecución:</p> <p>Pasos de ejecución: Para mostrar las respuesta de un plan se debe ir al menú principal y dar clic en la opción “planes” seguidamente se mostrará una ventana con un listado de los planes insertados en la Base de datos, se da clic en mostrar y saldrá una vista con los riesgos de ese plan,luego se da clic en mostrar respuesta y sale la respuesta del plan.</p>	
Resultado esperado: Respuesta de un plan	

Tabla 123 Caso de Prueba # 18

Caso de prueba	
Número: 18	Número :Historia de usuario: 18
Nombre: Modificar respuesta del riesgo	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Responsable de área, Vice-Decano, o Decano.	

<p>Entrada/ Pasos de ejecución:</p> <p>Pasos de ejecución: Cuando se quiere modificar una respuesta de un riesgo, se va al menú principal y en la opción, planes ,se da clic, seguidamente saldrá una lista de todos los planes adicionados ,luego se dará clic en editar y saldrá una vista con todos los datos de la respuesta para que sean editados</p>
<p>Resultado esperado: La respuesta ha sido modificada.</p>

Tabla 124 Caso de Prueba # 19

Caso de prueba	
Número: 19	Número: Historia de usuario: 19
Nombre: Eliminar respuesta del riesgo	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Responsable de área, Vice-Decano, o Decano.	
<p>Entrada/ Pasos de ejecución:</p> <p>Pasos de ejecución: Para eliminar la respuesta del riesgo se debe ir al menú principal y dar clic en la opción “planes” seguidamente se mostrará una ventana con un listado de los planes insertados en la Base de datos, se da clic eliminar y saldrá un mensaje de que se eliminó el plan correctamente.</p>	
Resultado esperado: La respuesta del riesgo eliminada correctamente	

Tabla 125 Caso de Prueba # 20

Caso de prueba	
Número: 20	Número :Historia de usuario: 20
Nombre: Mostrar área	
Condiciones de ejecución: El usuario debe estar autenticado con el rol	

Responsable de área, Vicedecano, o Decano.
<p>Entrada/ Pasos de ejecución:</p> <p>Pasos de ejecución: Para mostrar un área se debe ir al menú principal y dar clic en la opción “áreas” seguidamente se mostrará un ventana con un listado de las áreas insertadas, luego de estar en esa ventana dar clic en la opción “Ver”, y saldrá una vista con la descripción del área.</p>
Resultado esperado: La descripción de un área.

Tabla 126 Caso de Prueba # 21

Caso de prueba	
Número: 21	Número :Historia de usuario: 21
Nombre: Listar área	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Responsable de área, Vicedecano, o Decano.	
<p>Entrada/ Pasos de ejecución:</p> <p>Pasos de ejecución: Para listar áreas se debe ir al menú principal y dar clic en la opción “Áreas” seguidamente se mostrará un ventana con un listado de las áreas insertadas.</p>	
Resultado esperado: Lista de áreas	

Tabla 127 Caso de Prueba # 22

Caso de prueba	
Número: 22	Número :Historia de usuario: 22
Nombre: Editar área	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Responsable de área, Vicedecano, o Decano.	

<p>Entrada/ Pasos de ejecución:</p> <p>Pasos de ejecución: Para editar un área se debe ir al menú principal y dar clic en la opción “Áreas” seguidamente se mostrará una ventana con un listado de las áreas insertadas, luego de estar en esa ventana dar clic en la opción “editar”, y saldrá una vista para que se cambien los datos del área, y por último se le da clic en Actualizar.</p>
<p>Resultado esperado: se editó un área</p>

Tabla 128 Caso de Prueba # 23

Caso de prueba	
Número: 23	Número :Historia de usuario: 23
Nombre: Eliminar área	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Responsable de área, Vicedecano, o Decano.	
<p>Entrada/ Pasos de ejecución:</p> <p>Pasos de ejecución: Para eliminar un área se debe ir al menú principal y dar clic en la opción “Áreas” seguidamente se mostrará una ventana con un listado de las áreas insertadas, luego de estar en esa ventana dar clic en la opción “Ver”, y saldrá una vista con la descripción del área, ahí en esa vista se encuentra la opción eliminar, se da clic en la opción y saldrá un cartel de que el área ha sido eliminada correctamente.</p>	
Resultado esperado: Se eliminó un área	

Tabla 129 Caso de Prueba # 24

Caso de prueba	
Número: 24	Número :Historia de usuario: 24

Nombre: Listar usuario
Condiciones de ejecución: El usuario debe estar autenticado con el rol Decano.
Entrada/ Pasos de ejecución: La entrada consiste en los datos del usuario que se deben introducir para ser insertado en la base de datos. Pasos de ejecución: Para listar los datos de un usuario debe ir al menú y seleccionar la opción “Usuarios”. En la ventana que se va a mostrar una lista de los usuarios que se encuentran añadidos en la Base de Datos.
Resultado esperado: Una lista de usuarios.

Tabla 130 Caso de Prueba # 25

Caso de prueba	
Número: 25	Número :Historia de usuario: 25
Nombre: Mostrar usuario	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Decano.	
Entrada/ Pasos de ejecución: Pasos de ejecución: Para mostrar un usuario debe ir al menú y seleccionar la opción “Usuarios”. En la ventana que se va a mostrar una lista de los usuarios que se encuentran añadidos en la Base de Datos, seguidamente se debe dar clic en la opción “Ver”, saldrá una vista con los datos del usuario introducido en la Base de datos.	
Resultado esperado: Una lista de usuarios.	

Tabla 131 Caso de Prueba # 26

Caso de prueba

Número: 26	Número :Historia de usuario: 26
Nombre: Eliminar usuario	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Decano.	
<p>Entrada/ Pasos de ejecución:</p> <p>La entrada consiste en los datos del usuario que se deben introducir para ser insertado en la base de datos.</p> <p>Pasos de ejecución: Para eliminar un usuario se debe ir al menú y seleccionar la opción "Usuarios". En la ventana que se va a mostrar debe presionar en el botón "Ver". Una vez presionado el botón saldrá una descripción de un usuario, se dará clic en la opción eliminar y seguido se le muestra un mensaje de que el usuario se ha eliminado.</p>	
Resultado esperado: El usuario es eliminado.	

Tabla 132 Caso de Prueba # 27

Caso de prueba	
Número: 27	Número :Historia de usuario: 27
Nombre: Modificar categoría	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Decano.	
<p>Entrada/ Pasos de ejecución:</p> <p>La entrada consiste en los datos del usuario que se deben introducir para ser insertado en la base de datos.</p> <p>Pasos de ejecución: Para modificar una categoría se debe ir al menú y seleccionar la opción "Categoría". En la ventana que se va a mostrar debe presionar en el botón "Editar". Una vez presionado el botón saldrá una vista con los datos de la categoría que se quiere modificar, se dará clic en la opción</p>	

modificar y seguido se le muestra un mensaje de que se ha modificado la categoría correctamente.
Resultado esperado: Una categoría modificada

Tabla 133 Caso de Prueba # 28

Caso de prueba	
Número: 28	Número :Historia de usuario: 28
Nombre: Mostrar categoría	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Decano.	
<p>Entrada/ Pasos de ejecución:</p> <p>La entrada consiste en los datos del usuario que se deben introducir para ser insertado en la base de datos.</p> <p>Pasos de ejecución: Para mostrar una categoría se debe ir al menú y seleccionar la opción “Categoría”. En la ventana que se va a mostrar debe presionar en el botón “Ver”. Una vez presionado el botón saldrá una vista con los datos de la categoría seleccionada.</p>	
Resultado esperado: Una categoría mostrada.	

Tabla 134 Caso de Prueba # 29

Caso de prueba	
Número: 29	Número :Historia de usuario: 29
Nombre: Eliminar categoría	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Decano.	
Entrada/ Pasos de ejecución:	

La entrada consiste en los datos del usuario que se deben introducir para ser insertado en la base de datos.

Pasos de ejecución: Para eliminar una categoría se debe ir al menú y seleccionar la opción "Categoría". En la ventana que se va a mostrar debe presionar en el botón "Ver". Una vez presionado el botón saldrá una vista con los datos de la categoría seleccionada, luego se da clic en la opción eliminar y saldrá un mensaje de que la categoría ha sido eliminada.

Resultado esperado: Una categoría eliminada.