Universidad de las Ciencias Informáticas

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

Sistema informático para la gestión de riesgos en la auditoría de la calidad del software

Autores

Osmel Antonio Larduet Hechavarría

Heriberto Gordillo Hernández

Tutores

MSc. Dialexis Acosta Molina

Ing. Yusdel Meriño Almaguer

La Habana, Junio de 2013

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del trabajo titulado: Sistema informático para la gestión de riesgos en la auditoría de la calidad del software y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

_____ _____

Osmel Antonio Larduet Hechavarría

Heriberto Gordillo Hernández

_____ _____

MSc.Dialexis Acosta Molina

Ing. Yusdel Meriño Almaguer

El hoy no es más que el recuerdo
del ayer, y el mañana el sueño de hoy.

Khalil Gibran

Osmel:

Es importante para mí, agradecer desde lo más profundo de mi corazón, a todas aquellas personas que de una forma u otra, estuvieron a mi lado y confiaron en mí.

A mi familia por apoyarme y darme fuerzas para continuar. A Marta y Mario que fueron como unos padres para mí. Gracias, muchísimas gracias por todo.

A mi GOOGLE, mi FLAK, mi CURRUCUCUCU por haberme apoyado y ayudado en todas las cosas. Por ser lo mejor que me pudo haber pasado en esta universidad. Millones de gracias para ti. Eres y serás muy especial en mi vida.

A todos mis hermanitos (Javier, Yunier, Marlon, Pepe, Heriberto, Sol) por estar ahí en todos los momentos difíciles y alegres.

Heriberto

Agradezco a todos los que han contribuido con la realización de este trabajo y con mi formación profesional, especialmente:

A mis padres, por ser mis mayores consejeros y el motor impulsor de todos mis logros, por estar presentes en cada momento de mi vida, además de apoyarme aunque no tenga la razón. Muchas gracias por todo lo que me han dado en este mundo, Cary y Heriberto.

A mis tíos y primos que me han apoyado mucho también incondicionalmente, además de ser un ejemplo para los que están comenzando a formarse como profesionales. A mi tío Mario, que desde que nací me ha dado su apoyo, pues sin su ayuda y confianza no hubiera podido llegar hasta aquí.

A mis abuelos, ya que cada uno ha aportado su granito de arena en este sueño, incluso, lo es para ellos también.

A mi suegry Deily que me ha apoyado y ha depositado toda su confianza en mí.

A mi compañero de Tesis, Osmel, por formar parte de la obtención de este logro, que es la formación como profesional, además de ser un buen amigo.

A toda mi familia, que de una forma u otra se ha preocupado y me ha apoyado a lo largo de la carrera.

Agradecimientos comunes

A las personas que me ayudaron sin protestar para la confección de esta tesis: Eduardo, Adrián, a la maestra Yisel y a Edelso por ganar el concurso. A los que se me quedan, gracias por ayudarnos en la realización de la tesis.

A nuestro equipo LA FURIA, somos los mejores del mundo, después del Barza y el Real Madrid, claro, sin ustedes el fútbol no sería nada.

A nuestros compañeros de aula que vienen junto a nosotros desde primer año, pasando juntos duros momentos de estudio y apoyándonos mutuamente en todo momento. Gracias por ser el grupo más unido que hemos tenido como estudiantes.

A nuestros tutores Dialexis y Yusdel por haber tenido tanta paciencia con nosotros y por ayudarnos muchísimo en el desarrollo de la tesis. Gracias por la confianza depositada en nosotros.

A todos los profesores que nos impartieron clases, de los cuales aprendimos muchísimo.

A los miembros del tribunal de tesis y oponente que siempre fueron muy críticos con la tesis, contribuyendo a su calidad.

Heriberto

A mi Mamá por ser tan especial conmigo y darme esa confianza que se necesita para lograr algo en la vida, por quererme y entenderme, por ayudarme, por malcriarme y darme todo el amor que necesitaba para lograr mi sueño. Para ti, que siempre voy a ser tu niño chiquito. Mamá, te quiero mucho.

A mi Papá que me quiere y que ayuda en lo que haga falta.

A mi tío Mario que siempre me ha brindado su apoyo incondicional y su aliento a seguir luchando y ayudarme a vencer los obstáculos y hacer realidad mi sueño. Por ser tan especial, pues desde la cuna he sido como un hijo para él y como un padre padre para mí. A mi abuelo el Niño, que aunque ya no está me inculcó los deseos de aprender y lo extraño mucho.

Osmel

A mi Madre por ser fuente inagotable de luz, vida, amor, ternura, cariño, enseñanza, sacrificio. Siento orgullo de tener la madre más bella, luchadora y sacrificada del mundo.

A mi Padre (El Salvaje) por ser un ejemplo de hombre, de amigo. Por enseñarme y educarme. Espero que estés orgulloso de mí. Gracias PÁ.

A mi hermanita Arianne, por ser mi ejemplo, mi ídolo. La vida no nos ha dado tiempo para convivir juntos, pero los que hemos tenido han sido muy lindos. Gracias Ari, te quiero muchísimo.

La realización de una auditoría siempre va a estar expuesta a amenazas, riesgos y vulnerabilidades potenciales. Una adecuada gestión de riesgos permite identificar, analizar, evaluar y prevenir estos con el objetivo de disminuir o potenciar su impacto en cualquier actividad. En el Centro Nacional de Calidad del Software a pesar que existe un proceso para auditar la actividad productiva, no se gestionan adecuadamente los riesgos que pueden afectar este proceso. La presente investigación tiene como objetivo diseñar un proceso de gestión de riesgos en auditorías de la calidad del software, respaldado por una aplicación informática que ayudará a perfeccionar este proceso. Se realiza el análisis de los modelos, procesos, políticas y normas referentes a la gestión de riesgos, las actividades principales y los artefactos que intervienen. La metodología que condujo el proceso de desarrollo fue *eXtreme Programming*. La investigación realizada ayudó a la obtención de un procedimiento que describe el proceso de gestión de riesgos en las auditorías de la calidad del software, así como un listado de riesgos que pueden afectarla. La informatización de dicho proceso agiliza el trabajo del personal auditor, evitando que se convierta en una actividad tediosa que dificultar la calidad de la misma y ayudando a la toma de decisiones con mayor rapidez.

Palabras clave: auditoría de la calidad, calidad, gestión de riesgos, modelos, normas, proceso, riesgo, software.

Índice

Introducción

En la actualidad, el vertiginoso avance de las Tecnologías de la Información y las Comunicaciones (TICs), ha propiciado que empresas, entidades y organismos enfaticen en la necesidad de crear productos con calidad, para satisfacer las necesidades y expectativas de los clientes. Lo que ha abierto las puertas a la investigación e innovación, impulsada por la curiosidad, creatividad y empeño del hombre. La producción de software se ha beneficiado con el desarrollo de las tecnologías, como consecuencia de estos avances tecnológicos se han creado novedosos programas, sistemas informáticos y nuevas formas de comunicación.

Cuba trabaja por insertarse en el mercado del desarrollo de software. En la Universidad de las Ciencias Informáticas (UCI) una de las actividades fundamentales que se llevan a cabo, es la producción de software. El objetivo primordial es la obtención de productos de calidad que puedan ser exportados a empresas nacionales como internacionales.

El Centro Nacional de Calidad del Software (CaliSoft), brinda servicios de consultorías asociadas a la ingeniería y calidad de las aplicaciones desarrolladas en Cuba. De igual forma controla y audita el uso de normativas técnicas, procedimientos, documentos estandarizados y buenas prácticas para la calidad del desarrollo de software en el país. Dicho centro brinda servicios de formación en los temas de Calidad e Ingeniería basados en los requisitos de la norma ISO 9001:2008: Sistema de Gestión de la Calidad-Requisitos con tendencia a la mejora continua del sistema, impartidos por un personal de alta calificación.

En las auditorías ejecutadas por este centro a la actividad productiva, no se le otorga un tratamiento a profundidad a los riesgos que pueden afectar el proceso, de modo que se pueda prever con anterioridad su ejecución exitosa. El mismo no es guiado por una metodología formal de gestión de riesgos. Además, no se tiene una medida aproximada del nivel de riesgos a la que pueda estar expuesta una auditoría, lo que pudiera ocasionar atrasos en su ejecución e impedir el logro exitoso de los objetivos trazados.

La gestión de los riesgos de una auditoría constituye un medidor importante en el logro de los objetivos propuestos para dicha actividad. Administrar los posibles riesgos a los que puede exponerse este proceso con antelación, previene a desperdiciar esfuerzos y recursos que conlleven al fracaso del proceso de auditorías de la calidad para proyectos de desarrollos de software de la UCI.

La realización manual del proceso de gestión de riesgos puede atentar contra la seguridad y calidad de la información que se genera, puede convertirse en un proceso engorroso que consumiría mayor cantidad de recursos y tiempo en la toma de decisiones, constituye un alto costo de explotación del personal, un alto costo de operaciones y el rendimiento continuado es inestable.

Después de la problemática antes expuesta se define el *problema de investigación* en la siguiente interrogante:

¿Cómo prever la ejecución exitosa del proceso de auditoría de la calidad del software?

El *objetivo general* del presente trabajo es desarrollar una aplicación informática para la gestión de riesgos del proceso de auditorías de la calidad del software.

El *objeto de estudio* se enmarca en el proceso de gestión de riesgos y el *campo de acción* en el proceso de gestión de riesgos en la auditoría de la calidad del software.

Del objetivo general se derivan los siguientes *objetivos específicos:*

- ✓ Analizar los principales conceptos sobre la gestión de riesgos en la auditoría de la calidad del software y las aplicaciones informáticas existentes dedicadas a la gestión de riesgos.
- ✓ Analizar y seleccionar las tecnologías y herramientas existentes para desarrollar aplicaciones web.
- ✓ Describir la propuesta del proceso de gestión de riesgos en la auditoría de la calidad del software.
- ✓ Implementar un sistema para la gestión de riesgos del proceso de auditoría de la calidad del software.
- ✓ Validar la propuesta del proceso de gestión de riesgos y el sistema desarrollado.

Se espera obtener un procedimiento que guíe la gestión de riesgos en la auditoría de la calidad del software, con las principales actividades que dicho proceso involucra y una breve descripción de las mismas; apoyada en una aplicación informática para gestionar los riesgos en las auditorías.

Para dar cumplimiento a los objetivos del trabajo se planificaron las siguientes tareas:

- ✓ Análisis documental sobre los conceptos de riesgos, las auditorías de la calidad del software y el proceso de gestión de riesgos, sus técnicas, modelos, metodologías y herramientas.
- ✓ Análisis y comparación de las aplicaciones informáticas existentes que son utilizadas en el proceso de gestión de riesgos.

- ✓ Análisis del estado del arte de las normas, reglamentos, modelos y métodos de gestión de riesgos de una auditoría de la calidad del software.
- ✓ Selección de las herramientas, metodología y lenguajes de programación de desarrollo de software que serán utilizados para la implementación de la aplicación informática de gestión de riesgos para auditorías de la calidad del software.
- ✓ Construcción del diseño de una aplicación informática de gestión de riesgos de la auditoría de la calidad para proyectos de desarrollo de software.

A partir de lo expuesto anteriormente se propone la siguiente *idea a defender*: si se diseña e informatiza un procedimiento para la gestión de riesgos en el proceso de auditoría de la calidad del software, se podrá pronosticar exitosamente su ejecución.

Para la realización del presente trabajo se utilizaron varios métodos científicos que guiaron la investigación. Entre los métodos utilizados se encuentra el histórico-lógico para la revisión crítica de la bibliografía existente sobre la gestión de riesgos y para realizar un análisis del comportamiento del objeto de estudio en el transcurso del tiempo. También se empleó el método hipotético-deductivo, partiendo de la suposición de que la definición y la elaboración de un procedimiento de gestión de riesgos para las auditorías de la calidad de software, puede influir positivamente en el desarrollo exitoso del proceso de auditorías. Se utilizó el método de encuesta para obtener el criterio de los especialistas sobre la propuesta del proceso de solución.

Para facilitar su comprensión, el documento estará compuesto por la siguiente estructura capitular:

Capítulo 1: incluye todos los aspectos teóricos de la gestión de riesgos que dan soporte a la investigación, principales conceptos relacionados con el objeto de estudio. Breve referencia de los antecedentes nacionales e internacionales sobre el tema, así como la información más actualizada acerca de los modelos, normas, procesos y métodos utilizados en la gestión de riesgos.

Capítulo 2: descripción del procedimiento propuesto para la gestión de riesgos como factores de éxito del proceso de auditoría de la calidad del software. Actividades, artefactos de entrada y salida, además de los roles que intervienen.

Capítulo 3: se documentan las fases de Exploración y Planificación de la metodología de desarrollo de software XP. Se identifican las Historias de Usuario (HU), se confecciona el Plan de Iteraciones y el Plan de Entrega.

Capítulo 4: documentación de las fases de Diseño, Implementación y Pruebas según la metodología XP, se precisan las tareas que se desarrollan para cada Historia de Usuario. Se realiza la Fase de Pruebas y se muestra el resultado de las pruebas de aceptación realizadas a la aplicación.

1 Fundamentación teórica

En el presente capítulo se realiza el estado del arte de los conceptos relacionados con el riesgo, la gestión de riesgos, el aseguramiento y las auditorías de la calidad. Se analizan las actividades principales de la gestión de riesgos, donde se hace énfasis en las actividades fundamentales para asegurar la ejecución exitosa de la auditoría de la calidad del software.

1.1 Argumentos teóricos de la Gestión de riesgos

Para abordar el tema de la gestión de riesgos es preciso analizar la definición de riesgo. El riesgo es uno de los conceptos más discutidos en los círculos académicos y profesionales. La Real Academia Española define el riesgo como una contingencia o proximidad de un daño (1).

El riesgo no implica necesariamente daño, ni otorga certezas, ni relaciones directas y deterministas, sino una probabilidad de ocurrencia de éste. La idea central del enfoque de riesgo es poder anticiparse al daño, y centrarse en la prevención, por lo que es indispensable que se realice una buena distinción entre riesgo y daño. El riesgo se haya de forma implícita asociado a toda actividad y a su vez implica elección e incertidumbre (2). El SEI [1] define al riesgo como “la posibilidad de sufrir una pérdida” (3).

Por su parte PMI[2] define el riesgo como un evento o condición incierto, que de producirse, tiene un efecto negativo o positivo sobre los objetivos trazados, como tiempo, coste, alcance o calidad (4).

Para los autores del presente trabajo el riesgo es aquel evento que trae consigo consecuencias adversas para el desarrollo normal de una actividad. Es el efecto acumulativo de acontecimientos desfavorables sobre los objetivos de la actividad planificada. Es la probabilidad que presenta un nivel de consecuencias adversas tanto económicas, sociales o ambientales en un sitio en particular y durante un período de tiempo definido.

El análisis de los conceptos de riesgo, anteriormente mencionados, apoyaron a los autores de la presente investigación a concluir que es necesario dar un enfoque global a los riesgos, para emplearse en las disímiles áreas del conocimiento, debe ser guiado, visualizado y empleado con el fin de alcanzar una meta.

[1] Software Engineering Institute (Instituto de Ingeniería de Software)

[2] Project Management Institute

La prevención se realiza mediante una secuencia de actividades que habrán de realizarse para lograr objetivos.

La práctica de la gestión de riesgo es hacer viable la realización de toda actividad. Es “la práctica compuesta de procesos, métodos y herramientas que posibilita dar tratamiento a los riesgos. Provee de un entorno disciplinado para la toma de decisiones pre activa en base a determinar constantemente que puede ir mal (riesgos), identificar cuáles son los riesgos más importantes en los cuales enfocarse e implementar estrategias para gestionarlos” (3).

La gestión del riesgo consiste en una serie de pasos que ayudan al equipo a comprender y a gestionar la incertidumbre, además de contribuir a la toma de decisiones (5). Un riesgo es un problema potencial (puede ocurrir o no), pero sin tener en cuenta el resultado, es necesario identificarlo, evaluar su probabilidad de aparición, estimar su impacto y establecer un plan de contingencia por si ocurre el problema.

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.

La práctica de una adecuada gestión de riesgos permite la protección de lo que tanto trabajo cuesta reunir. Debe permitirle a la organización tomar los riesgos correctos, proporcionando el conocimiento y la comprensión de los mismos, identificando los recursos y esfuerzos necesarios para alcanzar los resultados deseados, movilizando las energías necesarias para ello y evaluando los resultados contra las expectativas presupuestas; igualmente provee los medios para la temprana detección y corrección de decisiones erradas o inadecuadas.

Como consecuencia de que la gestión de riesgo ha sido objeto de estudio en investigaciones durante todo este tiempo, se han desarrollado distintos modelos, normas y procesos sobre la gestión de riesgo, en los que se proponen diferentes factores que se consideran determinantes en dicho proceso.

1.2 Modelos, normas y procesos sobre la gestión de riesgos

Tradicionalmente, los riesgos más tratados habían sido los riesgos financieros. En la última década, hay mayor preocupación respecto a otros tipos de riesgos que pueden conllevar al fracaso de las organizaciones. La administración del riesgo otorga seguridad y solvencia. Además, ofrece conocimiento y experiencia.

Según el estudio exploratorio sobre los métodos de gestión de proyectos de alto riesgo, Marcelo, Rodenes y Torralba (2009), plantean que los modelos de gestión de riesgos han evolucionado de forma sucesiva a través de diferentes etapas o generaciones (6).

Primera generación (Casuística): se definen los riesgos tecnológicos y las listas de comprobación de riesgos, y se limitan las tareas a la identificación de riesgos en los proyectos.

Segunda generación (Taxonómica): se analizan los riesgos al inicio del proyecto. Cocho, Adam y Torralba definen esta visión como "preventiva", "teorizante" y de medidas "curativa". También califican los modelos "meramente reactivos, con unas relaciones de causa-efecto basadas solo en una confianza que parte de experiencias poco validadas". Dentro de esta etapa se pueden incluir los siguientes modelos:

- ✓ Modelo de Boehm
- ✓ Modelo de Riesgos del SEI
- ✓ Modelo de Hall y su relación con el de madurez de SEI-CMM[3]
- ✓ Modelo SPR[4] de mejora de capacidad en la gestión del riesgo

Tercera generación (Causal): se refiere en particular a proyectos informáticos. Surge de forma simultánea en Europa y en Estados Unidos partiendo de la preocupación por proyectos de tanto riesgo como la adquisición o el desarrollo de software. Esta es la generación actualmente emergente. Los principales modelos de gestión de riesgos propuestos por esta generación son:

- ✓ Modelo MAGERIT[5] de Gestión de riesgos en Sistemas adaptado a Proyectos (Transición)
- ✓ Eurométodo
- ✓ Modelo McFarlan (Transición)
- ✓ Modelo RiskMan[6] e iniciativa RiskDriver
- ✓ Modelo PRisk

[3] Capability maturity model integration (Modelo Integrado de Madurez y las Capacidades)

[4] Source-Pathway-Receptor (Fuente-Vías-Receptor)

[5] Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información

[6] Risk Management (Administración de riesgos)

Es importante destacar que estos modelos están destinados a la administración de riesgos en proyectos informáticos dedicados a la producción de software. Al analizarlos se pudo apreciar que la mayoría de ellos guardan relación con el proceso a diseñar. De ellos fueron extraídos todos aquellos aspectos comunes y las características más significativas que puedan aportar a la confección del proceso de gestión de riesgo para una auditoría de la calidad de software.

Modelo Espiral Boehm

Se caracteriza fundamentalmente por un enfoque cíclico para el crecimiento incremental del grado de definición e implementación de un sistema mientras que disminuye su grado de riesgo, y por un conjunto de puntos de fijación para asegurar el compromiso del usuario con soluciones de sistema que sean factibles y mutuamente satisfactorias (7).

Modelo de Hall

Este modelo está incluido en la segunda generación de análisis de riesgos y en él están definidas dos actividades principales, la evaluación y el control del riesgo. La gestión del riesgo en este modelo genera una estrategia para decidir qué hacer en cada momento. El Modelo 6-D de las 6 disciplinas PPMMDD (Planear, Producir, Medir, Mejorar, Diseñar, Descubrir) soporta la mejora continua del proceso SEI (Humphrey) modelo de madurez de procesos en el desarrollo de software CMM, que es un método para definir y gestionar los procesos a realizar por una organización (6).

Gestión de riesgos según PMI

La Guía del PMBOK7 es un estándar en la gestión de proyectos desarrollado por PMI8. PMBOK es un término integral que describe la suma de conocimiento dentro de la profesión de gestión de proyectos incluyendo construcción, ingeniería y software. Esta norma establece una referencia para todo aquel interesado en la profesión de gestión de proyectos.

Según PMI, la gestión de riesgos en los proyectos incluye los procesos relacionados con la planificación de la gestión de riesgos, la identificación y el análisis de riesgos (cualitativo y cuantitativo), las respuestas a los riesgos, y el seguimiento y control de riesgos de un proyecto; la mayoría de estos procesos se actualizan

[7] Project Management Body of Knowledge (Proyecto Organismo de Gestión Del Conocimiento)

[8] Project Management Institute

durante el proyecto. Los objetivos de la Gestión de los Riesgos del Proyecto son aumentar la probabilidad de los eventos positivos, y disminuir la probabilidad y el impacto de los eventos adversos para el proyecto (8).

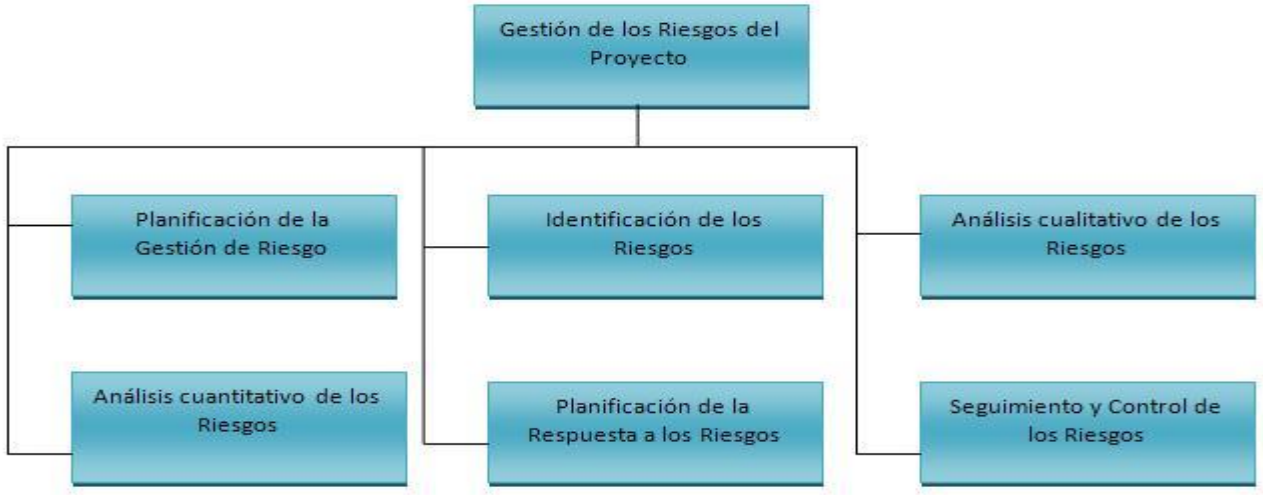

Figura 1. Gestión de riesgos según PMI (Fuente de elaboración: Infante **(7)**)

Gestión de riesgos según SEI. (SEI – CRM[9])

El modelo Continuous Risk Management (SEI-CRM), desarrollado por el SEI, es un método en el ámbito de la ingeniería del software cuyos conceptos, procesos y herramientas permiten gestionar de manera continua los riesgos de un proyecto, proporcionando un entorno disciplinado para la toma proactiva de decisiones a lo largo de todas las fases del proyecto: análisis de los problemas en potencia (riesgos), determinación de los riesgos importantes para elaborar estrategias y planes para gestionarlos. Estos riesgos son controlados hasta que se resuelven o se convierten en problemas menores, y son tratados como tales. Este modelo gestiona los riesgos como un ciclo básico, identifica, analiza, planifica, controla y comunica los riesgos a lo largo de todo el ciclo de vida del proyecto como se muestra a continuación (3).

[9] Continuos Risk Management (Gestión de Riesgos Continua)

Figura 2. Modelo de administración de riesgos según SEI. (Fuente de elaboración: SEI (3))

Gestión de riesgos en Cuba

El proceso de gestión de riesgos en la UCI ha sido desarrollado por varios estudiantes que han realizado sus trabajos de diploma sobre este tema. Constituyen un punto de partida para lograr que la gestión de riesgos se desarrolle satisfactoriamente mediante un estudio más exhaustivo y poniéndose en práctica los resultados obtenidos en los proyectos productivos donde se realizan estas investigaciones.

Entre las investigaciones desarrolladas está el Modelo de Gestión de Riesgos para Proyectos de Desarrollo de Software en la UCI (MoGeRi). Este modelo tiene como objetivo estandarizar el proceso de Gestión de riesgos para su utilización en los proyectos de la Universidad. Dicho modelo cuenta con 6 procesos, como se muestra en la figura 3.

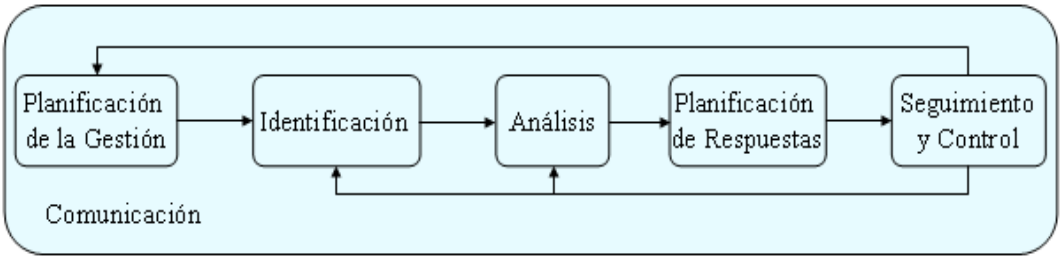

Figura 3. Procesos del MoGeRi. (Fuente de elaboración: Zulueta **(6)**)

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

Res 60 Contraloría General de la República

El componente Gestión y Prevención de Riesgos establece las bases para la identificación y análisis de los riesgos que enfrentan las entidades para alcanzar sus objetivos. Una vez clasificados los riesgos en internos y externos, por procesos, actividades y operaciones, y evaluadas las principales vulnerabilidades, se determinan los objetivos de control y se conforma el Plan de Prevención de Riesgos para definir el modo en que habrán de gestionarse. Existen riesgos que están regulados por disposiciones legales de los organismos rectores, los que se gestionan según los modelos de administración previstos. El componente se estructura en las siguientes normas (9):

- ✓ Identificación de riesgos y detección de cambio: se tipifican los riesgos que pueden afectar los objetivos propuestos, se nutre de experiencias derivadas de ocurrencias anteriores así como los que pueden preverse en el futuro. Es un proceso que se realiza de forma permanente, en el contexto externo pueden presentarse modificaciones en las disposiciones legales que conduzcan a cambios. 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.
- ✓ Determinación de los objetivos de control: el diagnóstico se realiza en reuniones por colectivos de áreas, direcciones o departamentos según corresponda, las cuales son presididas por la máxima autoridad del lugar, el dirigente sindical y los representantes de las organizaciones políticas; debe estar presente al menos uno de los integrantes del grupo que realizó la identificación y análisis de riesgos a nivel de la organización, con la información y antecedentes específicos del área. En estas reuniones se realiza entre todos un diagnóstico con los objetivos de control a considerar y se definen las medidas o procedimientos de control a aplicar, las mismas serán antecedidas de un trabajo de información y preparación de los trabajadores en asamblea de afiliados donde se les explica el procedimiento a seguir para su desarrollo.
- ✓ Prevención de riesgos de indisciplinas e ilegalidades, que continuados y en un clima de impunidad, provocan manifestaciones de corrupción administrativa o la ocurrencia de presuntos hechos delictivos.: constituye un conjunto de acciones o procedimientos de carácter ético-moral, técnico-organizativos y de control, dirigidas de modo consciente a eliminar o reducir al mínimo posible las causas y condiciones que propician los riesgos internos y externos, así como los hechos. 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. El Plan de Prevención de Riesgos constituye un instrumento de trabajo de la dirección para dar seguimiento sistemático a los objetivos de control determinados, se actualiza y analiza periódicamente con la activa participación de los trabajadores y ante la presencia de hechos que así lo requieran.

Como se puede apreciar la gestión de riesgos se pone de manifiesto en distintas esferas de la sociedad en riesgos empresariales, ante desastres naturales, en el desarrollo de software. Para cada una de estas áreas existen varios modelos de gestión de riesgos. Sin embargo, teniendo en cuenta el alcance de la presente investigación se profundizó en el estudio de los modelos de gestión de riesgos en el desarrollo de software. A partir de los modelos estudiados los autores consideran que la *Resolución 60 de la Contraloría General de la República* resume lo importante de cada uno de los modelos anteriores, y constituye un punto de partida para la elaboración de la propuesta de solución a partir de las actividades que propone por cada uno de los procesos.

La puesta en práctica de la gestión de riesgos trae consigo una serie de beneficios que ayudan a prever los eventos que pueden afectar el transcurso de cualquier actividad (10):

- ✓ Mayor posibilidad de alcanzar los objetivos.
- ✓ Incrementa el entendimiento de riesgos claves y sus más amplias implicaciones.
- ✓ Menos sorpresas y crisis.
- ✓ Mayor enfoque interno en hacer lo correcto de la forma correcta.
- ✓ Incrementa la posibilidad de qué cambios en iniciativas puedan ser logrados.
- ✓ Capacidad de tomar mayor riesgo por mayores recompensas.
- ✓ Más información sobre riesgos tomados y decisiones realizadas.

Estos beneficios permiten a las entidades crear productos o brindar servicios de excelencia. Alcanzar estas metas es el paradigma de la mayoría de las empresas en la actualidad, no obstante el logro de este objetivo está respaldado por una serie de actividades que encaminan a la organización en esa dirección. Las auditorías de la calidad también pueden tener en cuenta los beneficios de la gestión de riesgo para anticiparse a su ejecución exitosa o no.

El término Auditoría se ha empleado incorrectamente con frecuencia, ya que se ha considerado como una evaluación cuyo único fin es detectar errores y señalar fallas. La auditoría es un examen crítico que se realiza con el fin de evaluar la eficacia y eficiencia de una sección, organismo o entidad (11).

Una auditoría es como un examen en donde los procesos son calificados, es una evaluación, en donde se emite un dictamen que califica cómo se están realizando los procesos, incluyendo los aciertos, los errores, las acciones y omisiones (12).

La norma cubana 19011: 2012 define la auditoría de la calidad como “el proceso sistemático, independiente y documentado para obtener evidencias de la auditoría (registros, declaraciones de hechos o cualquier otra información) y evaluarlas de manera objetiva con el fin de determinar la extensión en que se cumplen los criterios de auditoría (conjunto de políticas, procedimientos o requisitos utilizados como referencia)” (13). Se trata de un examen metódico que se realiza para determinar si las actividades y resultados relativos a la calidad satisfacen las disposiciones previamente establecidas y que realmente se llevan a cabo, además de comprobar que son adecuadas para alcanzar los objetivos propuestos.

Juran (1987), plantea que la auditoría de la calidad es una herramienta de aseguramiento empleada para verificar y evaluar las actividades relacionadas con la calidad en el seno de una organización, dispone a elevar los niveles de actuación de la organización en todos sus contenidos y ámbitos, para que produzca bienes y servicios altamente competitivos (14).

El proceso de auditoría se encuentra dentro del conjunto de actividades y procedimientos entrelazados de forma tal que funcionen armónicamente, con el objetivo de asegurar la calidad en los productos y servicios brindados.

Es un proceso de vital importancia en el control y supervisión de la calidad. Pero también está expuesta a riesgos que pueden contribuir a su fracaso. Es necesario tomar medidas que permitan la realización exitosa de la misma. En la ejecución de la auditoría, la gestión de riesgos valdría de apoyo para saber qué hay en juego, saber a qué está expuesto el sistema y valorar la eficacia y eficiencia de las salvaguardas (15). En este sentido juega un papel fundamental la administración o gestión de riesgos que permitirá identificar, analizar y evaluar los riesgos para evitar pérdidas y aprovechar eficientemente los recursos disponibles.

1.3 Aplicaciones dedicadas a la gestión y administración de riesgos

A nivel mundial existen varias aplicaciones para la administración de riesgos orientadas a la minimización o evasión de riesgos, mediante la generación de principios y buenas prácticas de aplicación realista. Actualmente se utilizan diferentes herramientas para automatizar y mejorar los procesos de administración, gestión, identificación y análisis de los riesgos en auditorías. Algunas de estas herramientas son: Enterprise RiskAssessor[10] (ERA), AUDITA, MejoraRiesgos, Enterprise Risk Management (ERM Suite), Audicontrol, RiskAdvisor.

ERA es una herramienta de software para la implementación de sistemas de administración de riesgos y auditorías. Garantiza adquirir una solución ya madura, estable y totalmente flexible e implementar la auditoría integrada, evitando los riesgos y ganando tiempo y costos en un corto plazo. Se rige por una serie de estándares y normas mundiales y colombianas, como: Norma ISO 31.000, NTC[11]-5254, SARO[12]. También la herramienta MejoraRiesgos se basa en la Norma ISO 31.000, es una metodología para la mitigación y eliminación de riesgos. El software contiene varias etapas, tales como: contexto estratégico, identificación de riesgos, análisis, valoración, ponderación, determinación de políticas, mapa y plan de acción.

AUDITA permite realizar las auditorías con un enfoque orientado a riesgos y en consonancia con las normas referidas a riesgos operacionales. Algunos puntos a mencionar son la identificación de riesgos, los factores de riesgos, el impacto, la probabilidad, la evaluación de los controles aplicados mediante calificación de control interno, el riesgo residual, el riesgo aceptable, los riesgos transferidos. En todos los casos, la compañía decide la cantidad y los nombres de niveles con los que trabajará (16).

Estas herramientas son descartadas de la presente investigación, por ser estas aplicaciones de origen privativo, además no son multiplataforma. Por tanto, se hace necesaria la realización de un software que sea totalmente gratuito y que sea compatible con varios sistemas operativos. Para la construcción del sistema se hace indispensable un entorno de desarrollo acorde a las necesidades de los desarrolladores.

[10] Asesor de Riesgo Empresarial

[11] Norma Técnica Colombiana

[12] Sistema de Administración de Riesgos Operativos

El objetivo es tener las herramientas necesarias para elaborar una aplicación que cumpla con los requisitos del cliente y sea aplicable a cualquier proceso de gestión de riesgos.

1.4 Entorno de desarrollo

Los entornos de desarrollo de software son herramientas que relacionadas apoyan a los programadores a desarrollar aplicaciones. Constituye el conjunto de herramientas que el programador se auxilia para obtener productos con menor esfuerzo. Recursos como editores, compiladores y herramientas de análisis. A continuación se presenta un análisis de estas herramientas y tecnologías con sus características.

Se realiza el siguiente estudio de las herramientas, lenguajes, tecnologías y metodología teniendo en cuenta varios criterios de selección como fueron: las que cuenten con un alto reconocimiento a nivel mundial, que sean de código abierto o software libre, las que refieran una mayor documentación en la red y las que mayor dominio tenga el equipo de desarrollo y estén más acorde a las necesidades de desarrollo del mismo.

1.4.1 Metodología de desarrollo

Para lograr el éxito del proceso de desarrollo de software y la obtención de una documentación consistente durante la implementación de la solución, es recomendable el uso de una metodología de desarrollo. El estudio y análisis de la bibliografía consultada muestra la existencia de dos tipos de metodologías: tradicionales y ágiles. Entre las metodologías tradicionales o pesadas se encuentran RUP[13], MSF[14] entre otras como Win Win Spiral Model e Iconix. Estas son empleadas principalmente para grandes proyectos. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada.

Tabla 1. Comparación entre metodologías ágiles y tradicionales (Fuente de elaboración: Canós (17))

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.

[13] Proceso Unificado de Rational (Rational Unified Process)

[14] Microsoft Solution Framework

Especialmente preparados para cambios durante el proyecto.	Cierta resistencia a los cambios.
Impuestas internamente (por el equipo de desarrollo).	Impuestas externamente.
Proceso menos controlado, con pocos principios.	Proceso muchos 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.

Las tradicionales no se adaptan adecuadamente a los cambios, por lo que no cumplen con las expectativas de la presente investigación, ya que el desarrollo de la aplicación estará expuesto a constantes cambios. Además, el cliente estará presente en todo momento como un miembro más del equipo de desarrollo, elemento no contemplado dentro de las metodologías tradicionales. Por estas razones se propone el uso de una metodología ágil. Las metodologías ágiles más conocidas y utilizadas a nivel mundial y académico son: XP, Scrum y Crystal.

Programación Extrema o eXtreme Programming (XP)

Constituye la metodología más utilizada dentro del grupo de las ágiles. Su objetivo principal es asegurar la producción de software con calidad y cubrir las necesidades y requerimientos del usuario. Se centra en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de proyecto, potenciando el trabajo en equipo. Se basa en la simplicidad, la comunicación, la retroalimentación continua entre cliente y equipo de desarrollo (17).

XP consta de 4 fases: Planificación, Diseño, Desarrollo y Pruebas. Dentro de estas fases se desarrollan actividades que dan lugar a la creación a los artefactos que esta metodología propone, como se muestra en la figura 4:

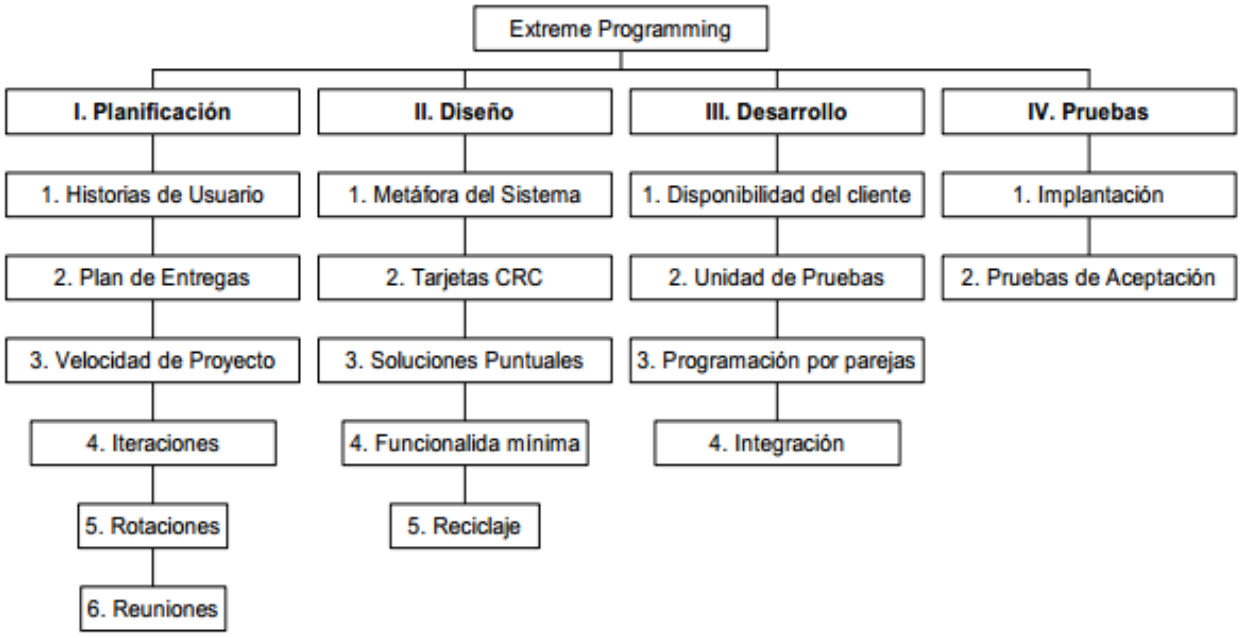

Figura 4. Fases y artefactos de XP (Fuente de elaboración: Escribano (18))

Características de XP:

- ✓ Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- ✓ Programación en parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto, de esta manera el código es revisado y discutido mientras se escribe.
- ✓ Frecuente integración del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.

- ✓ Corrección de los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes (19).

XP ha demostrado ser la metodología ágil que más estudios dispone y con mayor número de artículos escritos sobre ella, con conferencias internacionales de alto nivel y específicas sobre el tema.

1.4.2 Lenguajes de programación

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos, respectivamente (20).

- ✓ **Lenguajes del lado del cliente**

CSS3[15] y HTML5[16]: CSS es un lenguaje utilizado en la presentación de documentos HTML. Un documento HTML viene siendo coloquialmente "una página web". El CSS es utilizado para organizar la presentación y aspecto de una página web. Este lenguaje es principalmente utilizado para la presentación de colores, tipos y tamaños de letra. Se basa en intentar separar la estructura del documento HTML de su presentación. La página web sería lo que hay debajo (el contenido) y CSS sería un cristal de color que hace que el contenido se vea de una forma u otra (21).

La unión de estas dos tecnologías brindan muchas características y beneficios como: efectos visuales, mejores estilos para elementos, mejor visibilidad, fuentes personalizadas, nuevos formularios y validación, mejor acceso sin conexión, fáciles transformaciones y animaciones, sin plugins y lienzo. Estas son solo algunas de las ventajas que trae la utilización de HTML5 y CSS3, las cuales se pueden potenciar exponencialmente si las combinas con otras tecnologías como Javascript y PHP (22).

JavaScript: técnicamente, es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. Los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios (21). Se define como un lenguaje basado en objetos, no emplea clases ni herencia, típicas de la Programación Orientada a Objetos (POO). Implementado como parte de un navegador web, permitiendo mejoras en la interfaz de usuario y páginas web dinámicas en bases de datos locales al navegador. Se diseñó con una sintaxis similar al lenguaje C,

[15] Cascading Style Sheets (Hojas de estilo en cascada)
[16] HyperText Markup Language (lenguaje de marcado hipertextual)

aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes. Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web (23).

- ✓ **Lenguaje del lado del servidor**

PHP[17]*:* es un lenguaje interpretado del lado del servidor, utilizado generalmente para la generación de páginas web dinámica. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir a los desarrolladores, la generación de páginas web dinámicas rápidamente.

Al ser un lenguaje libre dispone de características que lo convierten en la herramienta ideal para la creación de páginas web dinámicas:

- ✓ Soporte para bases de datos como: MySQL y PostgreSQL.
- ✓ Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- ✓ Ofrece una solución simple y universal para las paginaciones dinámicas de la web de fácil programación.
- ✓ Soportado por una comunidad de desarrolladores, como producto de código abierto, goza de la ayuda de un grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- ✓ El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP (24).

Cuenta con una amplia documentación en internet. La UCI cuenta con una comunidad que le brinda nuevas funcionalidades para darle solución a diversos problemas. Se rige por las buenas prácticas de la programación orientada a objetos, lo cual hace que sea fácil y sencilla la interacción con dicho lenguaje.

[17] Hypertext Preprocessor

1.4.3 Marco de desarrollo

Symfony 2.0: es un marco de desarrollo (framework), pensado y diseñado para optimizar el desarrollo de aplicaciones web. A continuación, algunas premisas de Symfony 2.0: es fácil de instalar, interactuar, entender, además de ser independiente del sistema gestor de base de datos.

También, este framework proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y fácil de mantener. Por último, facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas.

Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Está desarrollado completamente con PHP 5. Se utiliza en sitios web de comercio electrónico de primer nivel. Compatible con la mayoría de los gestores de bases de datos, como: MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas Unix, Linux, como en plataformas Windows (23). Symfony 2.0 ha sido probado en numerosos proyectos reales como: Yahoo, Dailymotion, Opensky, Drupal 8, Behat, Doctrine, Propel, PHP Unit, Jackalope, Silex, PPI 2, Easybook, Midgard CMS, Zikula (antes PostNuke), phpBB, sigue la mayoría de las mejores prácticas y patrones de diseño para la web. Permite su integración con librerías desarrolladas por terceros. Además, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web (25).

1.4.4 Entorno de desarrollo integrado (IDE)

Entorno de Desarrollo Integrado o llamado IDE[18], es un programa informático compuesto por un conjunto de herramientas de programación. Es un entorno de programación que ha sido empaquetado como un programa de aplicación; consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes (26). A continuación se exponen algunas de las características y ventajas del uso del entorno de desarrollo integrado Netbeans.

[18] Integrated Development Environment (Entorno de desarrollo integrado)

Netbeans: es un producto libre y gratuito sin restricciones de uso. Además, es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe un número importante de módulos para extenderlo. Permite a los desarrolladores crear rápidamente aplicaciones utilizando lenguajes como PHP, C / C++ y Javascript.

La integración con Symfony es sencilla para su desarrollo, esto es posible porque desde el mismo IDE se pueden ejecutar todas las tareas de Symfony, incluyendo los comandos para crear proyectos y aplicaciones (27).

1.4.5 Sistemas Gestores de Base de Datos (SGBD)

MySQL: su diseño multihilo le permite soportar carga de forma eficiente. Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su rapidez y facilidad de uso. Esta aceptación es debida, en parte, a la infinidad de librerías y herramientas que permiten su uso a través de diferentes lenguajes de programación, además de su fácil instalación y configuración. Es sencillo y utilizado por muchos programadores; es soportado por varios lenguajes de programación entre ellos PHP. MySQL, junto con Apache y PHP forman un buen equipo para la creación de páginas web con contenido dinámico (28). En general, para la realización de la aplicación informática se tomó en cuenta que la velocidad y el número de acceso concurrente sea primordial.

1.4.6 Servidor Web Apache

El diseño modular de Apache permite a los administradores de sitios web elegir qué características se van a incluir en el servidor al seleccionar los módulos que se van a cargar, ya sea al compilar o al ejecutar el servidor. A continuación se muestran algunas características y ventajas de Apache:

Es una tecnología gratuita de código fuente abierta. Es un servidor configurable de diseño modular. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible su configuración para que ejecute un determinado script cuando ocurra un error en concreto. Tiene una alta configurabilidad en la creación y gestión de logs. Permite la creación de ficheros de log, de este modo se puede tener un mayor control sobre lo que sucede en el servidor (21).

Como servidor web se seleccionó Apache por ser gratuito, es flexible y funciona en distintas plataformas y entornos. Posee una estrecha relación con la mayoría de los lenguajes de desarrollo por parte del servidor, gestores de base de datos. Aparece entre los servidores web más utilizados por la sociedad.

1.5 Conclusiones parciales

El estudio del estado del arte sobre los conceptos más importantes relacionados con la calidad de software, la gestión de riesgos y las auditorías a la calidad de software, fomentaron las bases teóricas de la presente investigación.

1. La gestión de riesgos es el conjunto de actividades y métodos que se realizan, con el objetivo de anticiparse a los acontecimientos desfavorables que pueden afectar la realización de una actividad o evento.
2. La auditoría de la calidad como parte de toda disciplina, puede utilizar la gestión de riesgo para impedir el fracaso de su ejecución.
3. Para la gestión de riesgo en la presente investigación se seleccionaron las siguientes actividades a partir del estudio realizado:
    - ✓ Identificación de riesgos.
    - ✓ Análisis y evaluación de los riesgos.
    - ✓ Prevención de los riesgos.
    - ✓ Seguimiento y control.
4. Para el desarrollo e implementación de la propuesta de solución se decide utilizar XP como metodología de desarrollo de software, PHP 5.3.8 como lenguaje de programación, Symfony 2.0 como framework de desarrollo, MySQL 5.5 como gestor de base datos. Además, se seleccionaron como IDE, Netbeans en su versión 7.2.1, como lenguajes de programación del lado del cliente HTML5, CSS3 y JavaScript en su versión 1.9.

2 Propuesta del proceso de gestión de riesgos

En el presente capítulo se pretende describir detalladamente la propuesta de proceso de gestión de riesgos en el proceso de auditoría de la calidad de software. Sus principales actividades, los artefactos de entradas y salidas, además de los roles que intervienen.

2.1 Descripción del proceso de gestión de riesgos en la auditoría de la calidad del software

El siguiente proceso está basado en el análisis de los modelos descritos en el capítulo anterior. Es documentado mediante un procedimiento descrito textualmente en el *anexo # 1*, y gráficamente a continuación, siguiendo ambos, las normas de formato establecido en CaliSoft mediante el documento: IPP-3500:2008—Libro de proceso para definir procesos.

Nombre del proceso: Gestión de riesgos en la auditoría de la calidad del software.

Objetivo: establecer una guía para la administración de los riesgos que pueden afectar el proceso de auditoría de la calidad que lleva a cabo CaliSoft a los proyectos de software.

Alcance: el proceso es aplicable al proceso de auditoría de CaliSoft y a otro proceso de auditoría de la calidad del software.

Responsables:

- ✓ **Ejecuta:** Auditor líder de la auditoría de la calidad del software.
- ✓ **Responsable de su ejecución:** Coordinador de auditoría.

- ✓ **Revisa y actualiza este procedimiento:** Coordinador de auditoría.
- ✓ **Fiscaliza su cumplimiento:** Jefe del Dpto. de Consultoría y evaluaciones a procesos de CaliSoft.

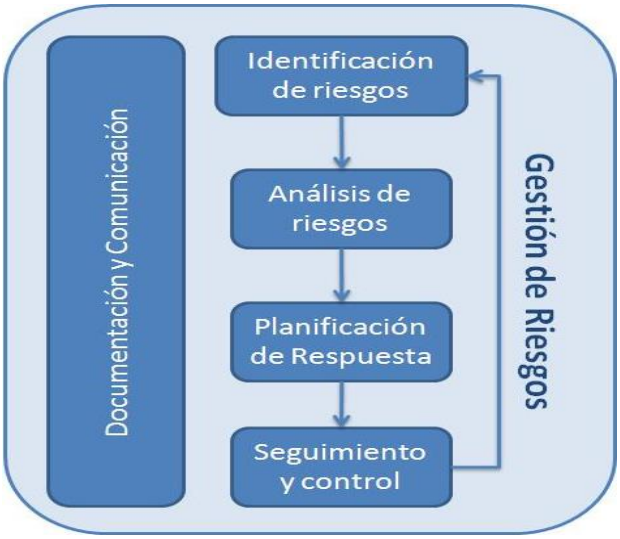

Figura 5 Representación gráfica del proceso de gestión de riegos en la auditoría de la calidad del software (Fuente de elaboración: propia)

El proceso está estructurado por 4 fases o etapas principales: identificación de riesgos, análisis y evaluación de los riesgos, prevención de los riesgos identificados y el seguimiento y control. Se obtiene como consecuencia la documentación que respalda la gestión de los riesgos en el proceso de auditoría y es anexada al expediente de la auditoría. Para cada etapa se representan las actividades, además de las entradas y salidas del procedimiento. Cada actividad tiene posibles técnicas para su ejecución que se explican de forma tal que oriente al interesado para su aplicación. La siguiente imagen describe los pasos y actividades que se desarrollan en este subproceso de la gestión de riesgos. Refleja los roles que intervienen y los artefactos que se generan. A continuación la figura 6 muestra gráficamente las actividades que se realizan en ella, además, se exponen aspectos importantes a tener en cuenta de cada uno de los subprocesos.

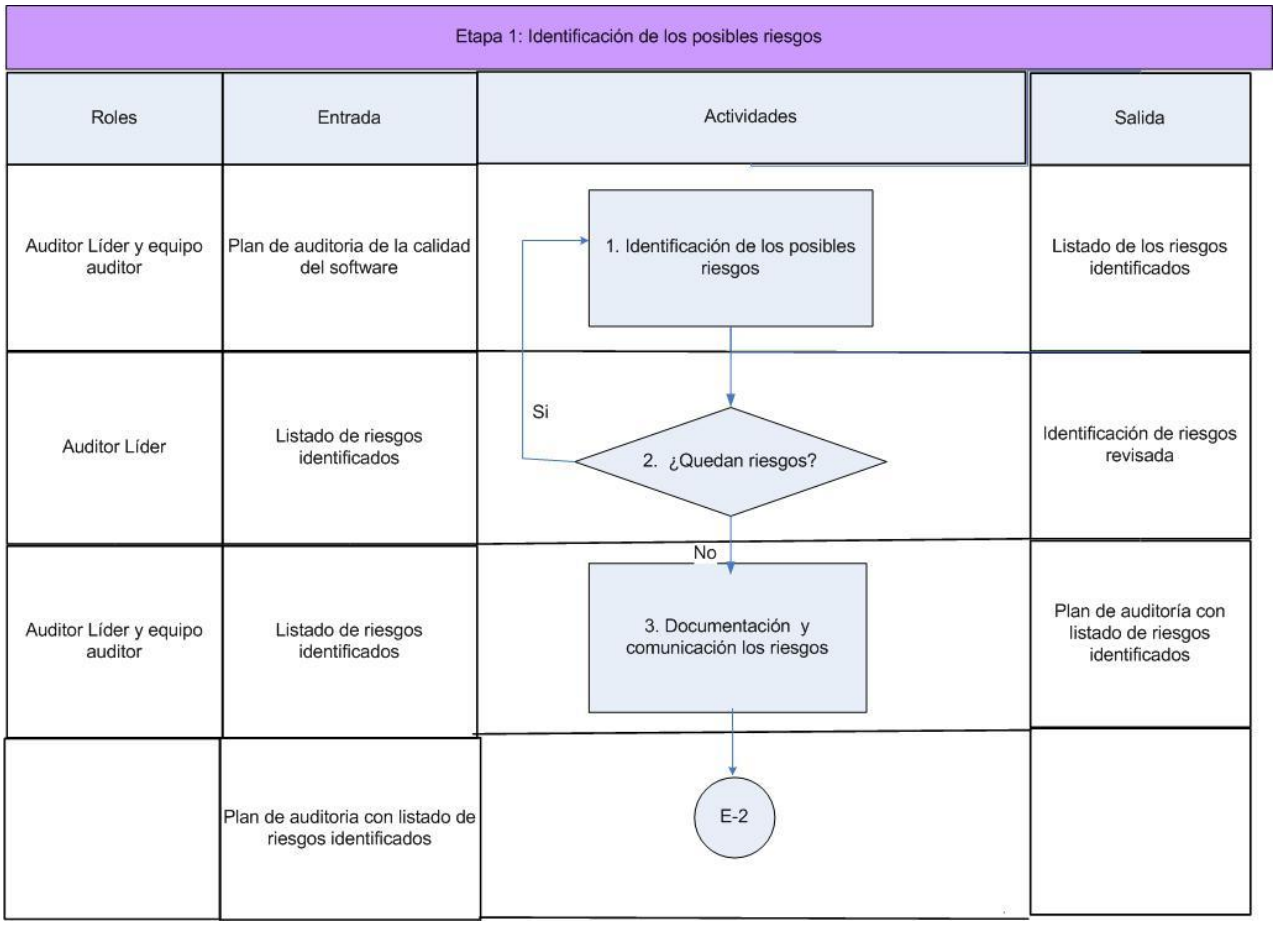

Figura 6. Descripción gráfica de la etapa I: Identificación de riesgos. (Fuente de elaboración: propia)

En esta etapa el auditor líder y el equipo auditor identifican los riesgos que pueden afectar el desarrollo de la auditoría. Para vaticinar estos y otros riesgos el equipo auditor identifica todos aquellos que pueden afectar el desarrollo exitoso de la auditoría (no importa lo improbable que sean). A través de una tormenta de ideas el equipo de auditores podrá realizar esta actividad, también pueden apoyarse en el uso de técnicas de identificación de riesgos como:

- ✓ Revisión de la documentación existente.
- ✓ Revisión, planificación y estimaciones.
- ✓ Diagrama de Ishikawa.
- ✓ Cuestionarios.

En este subproceso el principal aporte es la elaboración de un listado con los riesgos potenciales que pueden impedir el éxito de la auditoría, como se muestra en la figura # 7:

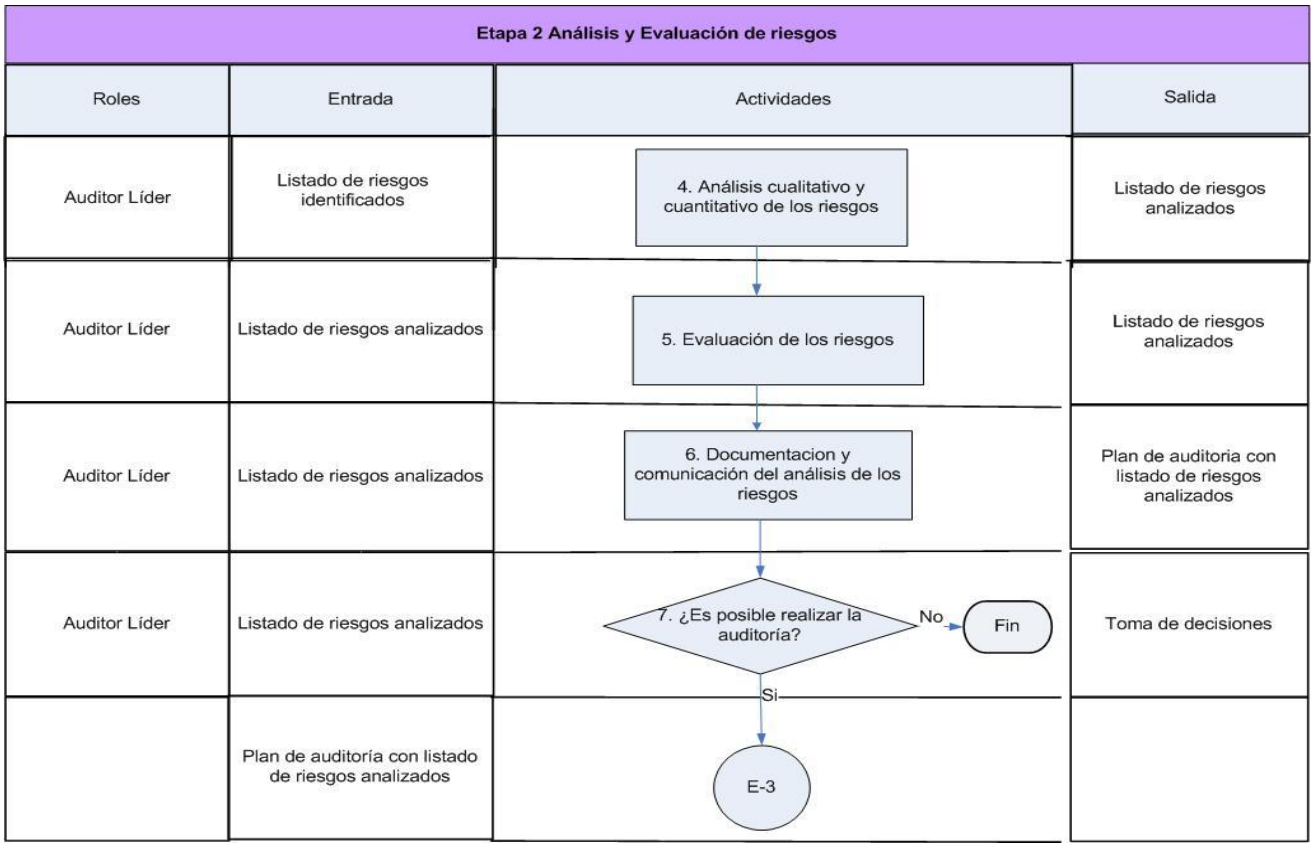

Figura 7. Descripción gráfica de la etapa 2: Análisis y evaluación de Riesgos. (Fuente de elaboración: propia)

El objetivo del análisis cualitativo y cuantitativo de los riesgos es cuantificar de forma precisa el impacto y probabilidad de ocurrencia de este. Para determinar la probabilidad de que ocurra un riesgo, los auditores realizan estimaciones individuales, estas son promediadas para después alcanzar un valor de consenso. En el análisis cualitativo la probabilidad de ocurrencia toma los siguientes valores: *Baja, Media y Alta*.

Probabilidad de ocurrencia baja: riesgo cuya aparición sea considerada mínima.

Probabilidad de ocurrencia media: riesgo cuya aparición sea considerada probable.

Probabilidad de ocurrencia alta: riesgo cuya aparición sea considerada frecuente.

A continuación se valora el impacto de cada riesgo. El impacto de un riesgo puede ser: *Bajo, Medio y Alto*.

Impacto Bajo: si el impacto de ocurrencia es tan insignificante que se logra realizar la auditoría sin dificultades, lográndose percibir la dificultad a simple vista y sea fácil darle solución.

Impacto Medio: si el impacto de ocurrencia del riesgo afecta el desempeño de la auditoría pero mantiene un ambiente de orden y planificación favorable, incidiendo en menor medida o casi nulo contra el tiempo, recursos y desempeño planificado.

Impacto Alto: si el impacto de ocurrencia del riesgo afecta la auditoría en la medida de que no se logre la ejecución del proceso y si se ejecuta, atenta negativamente contra el tiempo, recursos y desempeño planificado.

Entre estos parámetros se establece una correlación como muestra la Tabla 1.

Tabla 2. Matriz de correlación entre el impacto y la probabilidad. Fuente de elaboración (Carmona (29))

PROBABILIDAD DE OCURRENCIA	*IMPACTO DEL RIESGO*		
	BAJO	*MEDIO*	*ALTO*
	NIVEL DE ACEPTACIÓN DEL RIESGO		
ALTO	*MEDIO*	*BAJO*	*BAJO*
MEDIO	*MEDIO*	*MEDIO*	*BAJO*
BAJO	*ALTO*	*MEDIO*	*MEDIO*

Riesgo alto ⟶ Nivel de aceptación bajo

Riesgo medio ⟶ Nivel de aceptación medio

Riesgo bajo ⟶ Nivel de aceptación alto

Para la toma de decisiones en cuanto a la realización de la auditoría se determinaron intervalos probabilísticos que ayudarán al líder auditor a ejecutar, aplazar o abortar la auditoría. Se elaboró una métrica en sesiones de trabajo con expertos matemáticos, mediante resultados de tendencia sobre el comportamiento de los riesgos en auditorías de la calidad del desarrollo de software, realizadas entre el 2008 – 2009, donde se calcula la probabilidad para cada Nivel de Aceptación del Riesgo (NAR).

TR= Cantidad total de riesgos identificados.

$CantNAR_{alto}$ = Cantidad de riesgos con Nivel de aceptación del riesgo de tipo alto

$CantNAR_{medio}$ = Cantidad de riesgos con Nivel de aceptación del riesgo de tipo medio

$CantNAR_{bajo}$ = Cantidad de riesgos con Nivel de aceptación del riesgo de tipo bajo

P(X) = Probabilidad del riesgo

$P(CantNAR_{alto})$ = $CantNAR_{alto}$ / TR

$P(CantNAR_{medio})$ = $CantNAR_{medio}$ / TR

$P(CantNAR_{bajo})$ = $CantNAR_{bajo}$ / TR

En las tablas 3 y 4 se muestran los intervalos probabilísticos para evaluar la ejecución de la auditoría definidos por cada nivel de aceptación del riesgo. El Nivel de Aceptación del Riesgo de tipo *Alta* no se tomó en cuenta porque se da prioridad a los niveles de aceptación del riesgo bajo y medio. Además, la aceptabilidad *Alta* es el complemento de la suma de las tres probabilidades que en cualquiera de los casos no influiría en la decisión final.

Tabla 3. Nivel de Aceptación del Riesgo (*Baja*) (Fuente de elaboración: propia)

Intervalos	Evaluación
$0 \leq P(CantNAR_{bajo}) \leq 0.2$	Aceptar
$0.2 < P(CantNAR_{bajo}) \leq 0.39$	Retrasar
$0.39 < P(CantNAR_{bajo}) \leq 1.0$	Rechazar

Tabla 4. Nivel de Aceptación del Riesgo (*Media*) (Fuente de elaboración: propia)

Intervalos	Evaluación
$0 \leq P(CantNAR_{medio}) \leq 0.3$	Aceptar
$0.3 < P(CantNAR_{medio}) \leq 0.6$	Retrasar
$06 < P(CantNAR_{medio}) \leq 1.0$	Rechazar

La tabla 5 muestra el análisis final según los resultados de la evaluación de las probabilidades de los riesgos con nivel de aceptación de tipo bajo y medio.

Tabla 5. Evaluación Final (Fuente de elaboración: propia)

No	Probabilidad Baja	Probabilidad Media	Evaluación Final
1	*Aceptar*	*Aceptar*	*Ejecutar*
2	*Aceptar*	*Retrasar*	*Ejecutar*
3	*Aceptar*	*Rechazar*	*Aplazar*
4	*Retrasar*	*Aceptar*	*Aplazar*
5	*Retrasar*	*Retrasar*	*Abortar*
6	*Para cualquier otra combinación la auditoría es abortada*.		

En caso de que la evaluación final determina que la auditoría debe abortarse, el auditor líder comunica al Coordinador de auditoría la decisión y las razones pertinentes junto con el listado de riesgos identificados. El Coordinador tiene la responsabilidad de comunicar a la entidad el veredicto final.

Una vez tomada la decisión de ejecutar o aplazar la auditoría, los riesgos analizados son ordenados en un listado en orden descendente atendiendo a la probabilidad de ocurrencia y al impacto asociado a cada uno de ellos. Donde, los riesgos de alta probabilidad de materialización y alto impacto pasan a lo alto de la lista y los riesgos de baja probabilidad y menor impacto caen a la parte baja de la lista.

Este listado proporcionará el orden de prioridad en que deben ser tratados los riesgos. A los riesgos de probabilidades altas de ocurrencia y poco impacto se le prestará atención. Esto consigue una priorización del riesgo de primer orden.

El auditor líder estudia la lista ordenada resultante y define una línea de corte. La línea de corte implica que a los riesgos que quedan por encima de la línea se les prestará mayor atención en lo adelante. Los riesgos que queden por debajo de la línea son reevaluados para conseguir una priorización de segundo orden.

El impacto y la probabilidad del riesgo influyen de forma diferente en la gestión. Un factor de riesgo que tenga un impacto alto pero escasa probabilidad de que ocurra, no debería absorber una cantidad significativa de tiempo de gestión. Sin embargo, los riesgos de alto impacto con una probabilidad de ocurrencia probable o frecuente deberán ser atendidos lo antes posible. Los riesgos de bajo impacto y alta probabilidad deben tenerse en cuenta.

Para que sea útil la evaluación se deberá prestar especial atención a la ocurrencia de combinaciones de los riesgos. La ocurrencia simultánea de riesgos de impacto alto puede crear problemas para el logro de los objetivos. Prever la aparición de estas situaciones ayuda a contar con un mecanismo que disminuya el impacto de estas combinaciones sobre la auditoría.

Esta fase se repetirá a lo largo del proceso ya que el auditor líder debería volver a la lista de riesgo periódicamente para volver a evaluar cada riesgo, determinar qué nuevas circunstancias hayan podido cambiar su impacto o probabilidad. Como consecuencia de esta actividad, puede ser necesario añadir nuevos riesgos a la lista, quitar algunos que ya no sean relevantes y cambiar la posición relativa de otros. La elaboración de una métrica que facilita la toma de decisiones al personal auditor es el principal aporte en esta fase.

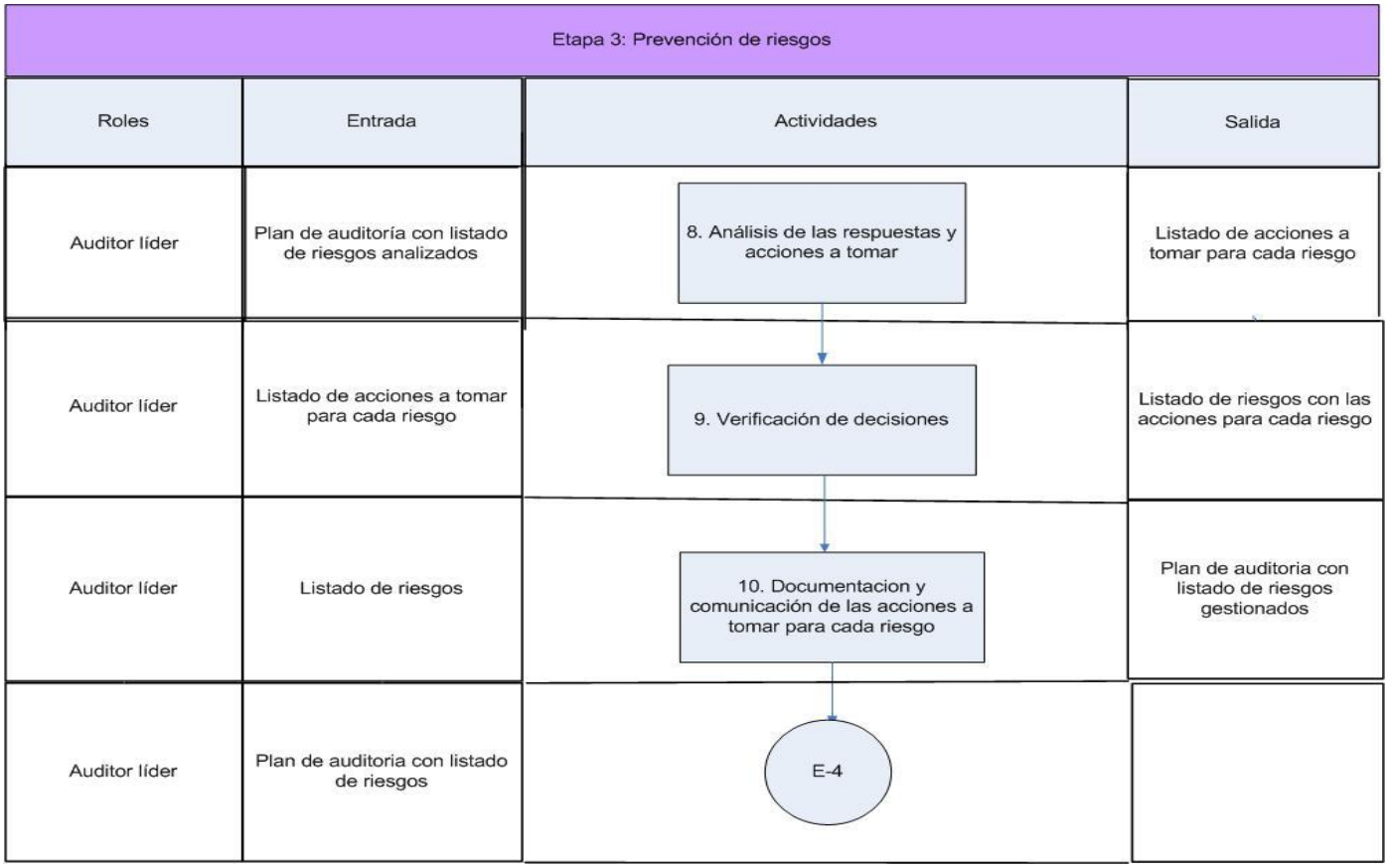

Figura 8. Descripción gráfica de la etapa 3: Prevención de riesgos (Fuente de elaboración: propia)

La figura 8 muestra gráficamente el subproceso de Prevención de Riesgos, que tiene como objetivo planificar los recursos y el cronograma para el cumplimiento de cada respuesta. La prevención de los

riesgos consiste en determinar qué acción se ejecutará frente a un riesgo determinado. La lista de riesgos analizados ofrece el orden en que deben ser tratados según esa prioridad. Los encargados del cumplimiento de la planificación de las respuestas establecidas ><para cada riesgo, son el Coordinador del proceso y el Auditor líder.

El auditor líder crea un listado detallado para controlar los riesgos más importantes identificados durante el análisis de riesgos y los efectos que estos provocan, así como las acciones que se deben tener en cuenta para contrarrestar el impacto que los riesgos provocan sobre el proceso de auditoría.

Para contrarrestar los riesgos se adoptan una serie de acciones que se ocupan de evitar o disminuir las consecuencias que pueden tener impactos negativos sobre los objetivos del proceso, en caso de ocurrir.

Para dar tratamiento a los riesgos con un nivel de aceptación alto se tomará la decisión de aceptarlos. En cambio, se tomará la decisión de mitigar o transferir a los riesgos cuyo nivel de aceptabilidad sea medio y evitar para los riesgos bajo nivel de aceptación. Cada una de estas acciones se explica a continuación:

- ✓ *Evitar:* implica cambiar el plan de auditoría para eliminar la amenaza que representa un riesgo adverso, aislar los objetivos de la auditoría del impacto del riesgo o relajar el objetivo que está en peligro, por ejemplo, ampliando el cronograma o reduciendo el alcance. Algunos riesgos que surgen en las etapas tempranas del proyecto pueden ser evitados aclarando los requisitos, obteniendo información, mejorando la comunicación o adquiriendo experiencia.
- ✓ *Transferir*: requiere trasladar el impacto negativo de una amenaza, junto con la propiedad de la respuesta, a un tercero. Transferir el riesgo simplemente da a otra parte la responsabilidad de su gestión; no lo elimina.
- ✓ *Mitigar*: implica reducir la probabilidad y/o el impacto de un evento de riesgo adverso a un umbral aceptable. Adoptar acciones tempranas para reducir la probabilidad de la ocurrencia de un riesgo y/o su impacto sobre el proceso a menudo es más efectivo que tratar de reparar el daño una vez ocurrido el riesgo.
- ✓ *Aceptar*: estrategia que se adopta debido a que rara vez es posible eliminar todo el riesgo. Esta estrategia indica que el equipo del proceso ha decidido no cambiar el plan de auditoría para hacer frente a un riesgo, o no ha podido identificar ninguna otra estrategia de respuesta adecuada.
- ✓ *Explotar*: elimina la incertidumbre asociada con un riesgo positivo en particular haciendo que la oportunidad definitivamente se concrete.

- ✓ *Compartir*: implica asignar la propiedad a terceros que están capacitados para capturar la oportunidad para beneficio del proceso.
- ✓ *Mejorar***:** busca fortalecer o facilitar la causa de la oportunidad, y dirigirse de forma proactiva a las condiciones que la disparan.

El líder de proyecto realiza un análisis de la factibilidad de cada respuesta planificada en cuanto a recursos asignados para su desarrollo y cumplimiento. En la misma se valora la posibilidad de cumplimiento de las respuestas planificadas apoyándose en los recursos que ha asignado el proceso para realizar la gestión de riesgos.

Conjuntamente se analizan las causas que provocaron la aparición del riesgo. Esta actividad es importante porque dotará de experiencia a los integrantes del equipo para futuros proyectos. También vale para calcular el efecto que este puede provocar.

El listado de las acciones o tareas a acometer para impedir o minimizar el impacto que trae aparejado la materialización de un riesgo, aporta conocimiento y experiencia. También se define el responsable de ejecutar las respectivas tareas y el tiempo de ejecución para ellas.

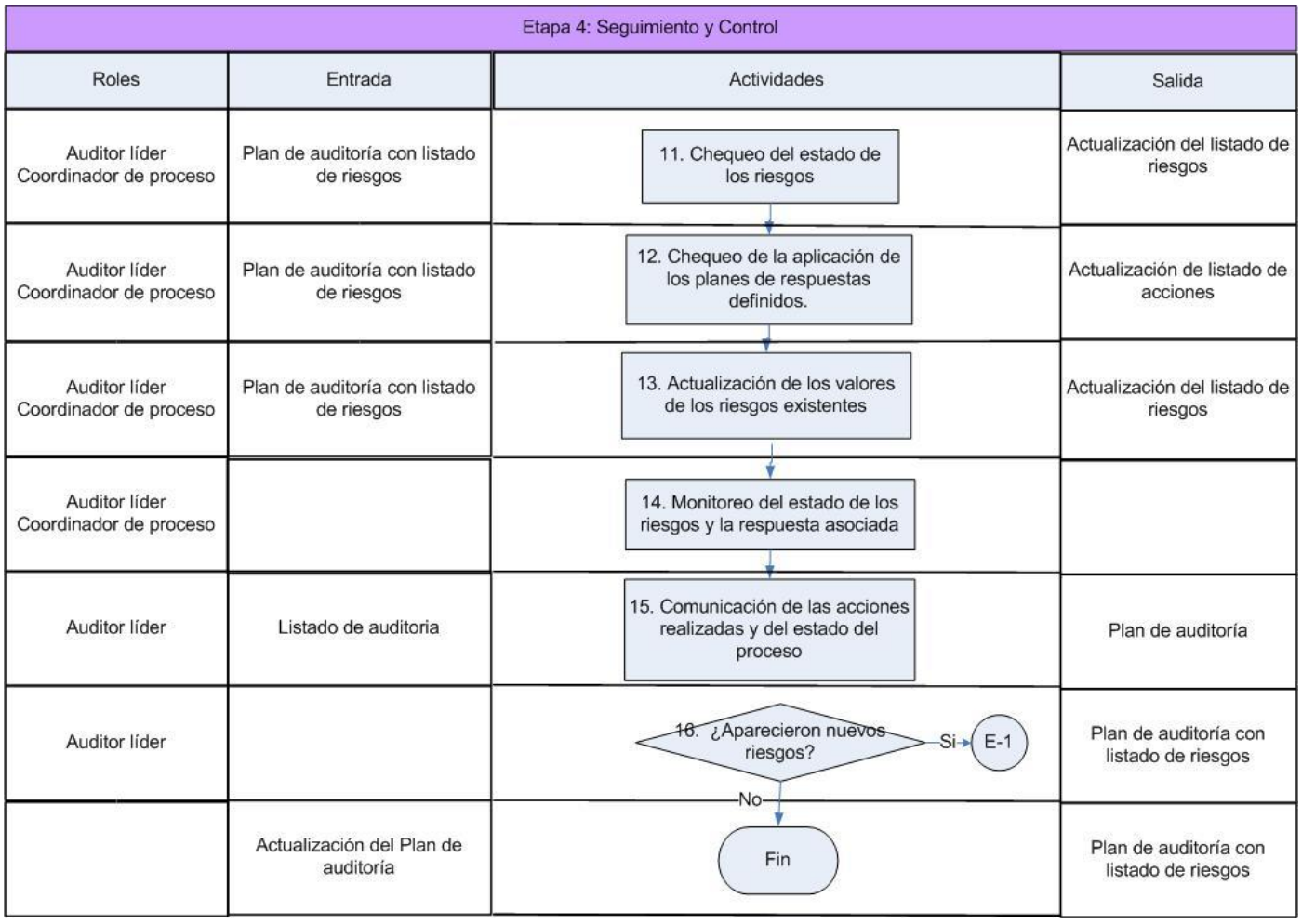

Figura 9. Descripción gráfica de la etapa 4: Seguimiento y control (Fuente de elaboración: propia)

La figura # 9 describe el flujo de actividades desarrolladas en esta etapa, que se ejecuta una vez iniciada la auditoría y prevalece durante todo el proceso. El Seguimiento y Control de los riesgos garantiza que las tareas que implementan medidas preventivas o planes de contingencia se realicen en tiempo y forma. Permite asegurar que las acciones definidas para mejorar las oportunidades y reducir las amenazas a los objetivos del proceso de auditoría, están siendo llevadas a cabo.

Durante esta fase se identifican, analizan nuevos riesgos, se realiza el seguimiento de los riesgos identificados anteriormente, se vuelven a analizar los riesgos existentes y se revisa la ejecución de las respuestas a los riesgos mientras se evalúa su efectividad.

Monitoreando el curso de los riesgos se logra actualizar y verificar el estado de los mismos según el efecto que han tenido las respuestas planificadas para contrarrestarlos. Con el objetivo de verificar nuevamente si

es factible la respuesta ejecutada o tomar en consideración cambiar de estrategia para reducir el impacto o la ocurrencia de los riesgos sobre el proceso.

Documentación y comunicación de la información sobre los riesgos

El auditor líder es el responsable de comunicar y documentar todos los aspectos importantes dentro de la gestión de riesgos. Detallar claramente cada riesgo y las acciones convenidas para tratar las posibles amenazas. Informar oportunamente las decisiones al equipo de auditores y otros interesados.

La documentación y comunicación es fundamental ya que estará presente durante todo el proceso, tiene importancia pues garantiza un flujo de información entre todos los miembros del grupo. Formaliza las lecciones aprendidas, los elementos y herramientas relevantes del proceso y plasma esta información en un formato reutilizable para el equipo y otros interesados. La comunicación de la información sobre los riesgos debe constituirse como un proceso continuado durante la gestión de riesgos y puede ponerse en práctica en cualquier momento. Se centra en la consecución de objetivos claves de forma general:

- ✓ Proporcionar calidad a las actividades de gestión de riesgos en el proceso de auditoría para que el equipo pueda obtener información.
- ✓ Hacer acopio de las lecciones aprendidas, especialmente las relativas a la identificación de riesgos y a las estrategias de mitigación, para que otros equipos puedan hacer uso de ellas. Esta información permitirá aumentar la base de conocimientos sobre los riesgos.
- ✓ Mejorar el proceso de gestión de riesgos gracias a la información proporcionada por el equipo.
- ✓ Determinación de las necesidades de información y comunicación de los interesados.
- ✓ Definición del tipo de información que debe ser comunicada.
- ✓ Definición de los métodos o tecnología a usar para transmitir la información y distribución de la información.

2.2 Aspectos del proceso

A continuación se describen aspectos importantes del proceso: Gestión de Riesgos en la Auditoría de la Calidad del Software.

Claridad: el proceso descrito tiene como objetivo guiar a los usuarios en su aplicación. Cada descripción de procedimiento, actividad o técnica está orientada a proveer un camino de la forma más práctica posible hacia una aplicación satisfactoria. Describe claramente cuáles artefactos se utilizan, cuándo interactuar con ellos y cómo ejecutar las actividades. El proceso está descrito utilizando una estructura jerárquica simple para no perder al lector. Contiene elementos gráficos para ayudar en su claridad.

Complejidad: no presentan alta complejidad en cuanto a su fundamento y a la estructura de los subprocesos involucrados. Sin embargo, este aspecto está dado principalmente por su forma descriptiva y la complejidad propia de las técnicas que define.

Define un conjunto de técnicas aplicables a las actividades de cada subproceso. A las técnicas se les reduce la complejidad al detallar cómo aplicarla e integrarla con el resto de los elementos de la gestión de riesgos. Describe cuándo comienza cada fase o etapa, para guiar a los auditores en la aplicación de la gestión de riesgos.

Descripción de los roles: describe en cada fase y actividad los roles participantes. Se describen las tareas que realizan cada uno de estos roles y las responsabilidades que tienen dentro de la gestión de riesgos.

Descripción de los artefactos: se detallan los artefactos generados e incluye la lista los riesgos de ejemplo en los anexos. Durante la descripción de las etapas se referencia cuáles documentos se usan como entrada, como salida, cuándo se hace y por quién se hace.

Aplicabilidad: es aplicable al proceso de auditoría de la calidad el software, como se describe en la sección Alcance del Modelo, no obstante es fácilmente adaptable a otros procesos que requieran el uso de la gestión de riesgos para alcanzar sus objetivos. Propone categorías de riesgos, métodos de identificación entre otros elementos aplicables a este y diferentes tipos de procesos en la universidad.

Otros criterios sobre el proceso: el proceso documentado a través de un procedimiento orienta a los interesados en gestionar los riesgos en el proceso de auditoría de la calidad del software, sus actividades y técnicas hacia la mejora de las oportunidades y la mitigación de los factores que pueden provocar el fracaso de la misma. Centra la explicación de sus técnicas de (identificación, análisis, seguimiento y control), en los objetivos de tiempo, eficiencia y seguridad principalmente, sirviendo como una herramienta para asegurarlos en alguna medida.

2.3 Validación del proceso de gestión de riesgos en la auditoría de la calidad del software

La validación de un proceso puede ser llevada a cabo aplicando varias técnicas. Entre las más utilizadas se encuentran las simulaciones a través de aplicaciones informáticas. Esta técnica no se pudo poner en práctica, por ser privativos estos sistemas. Otra de las técnicas es la aplicación práctica del procedimiento. Esta técnica no se ejecutó porque en el período de realización de la presente investigación no se tenían auditorías planificadas.

Por lo antes expuesto se adopta el método de experto, el cual permite tomar decisiones para aceptar o no la propuesta de acuerdo con los criterios definidos (37).

Para la aplicación de este método se efectuaron un conjunto de pasos, los cuales se detallan a continuación:

1. Se elaboran los criterios de evaluación de acuerdo a las características de la propuesta y se organizan por grupos.
    - ✓ Grupo # 1 Criterios de mérito científico
        - Valor científico de la propuesta.
        - Calidad de la investigación.
        - Aporte científico.
        - Novedad científica
    - ✓ Grupo # 2 Criterios de implantación
        - Satisfacción de las necesidades de los auditores del Centro Nacional de Calidad del Software.
        - Necesidad del empleo de la propuesta.
    - ✓ Grupo # 3 Criterios de flexibilidad
        - Adaptabilidad a las auditorías de la calidad del software.
        - Adaptabilidad a cualquier servicio referente a la gestión de riesgos.
    - ✓ Grupo # 4 Criterios de Impacto
        - Repercusión en las auditorías de la calidad del software.
        - Aceptación de la propuesta por los auditores.
        - Posibilidades de aplicación.
        - Impacto en el centro para el cual está destinado.
2. Se le asigna un peso relativo a cada grupo de criterios de acuerdo al porcentaje que representa cada grupo del total y los intereses a evaluar.
    - ✓ Grupo # 1……..25

- ✓ Grupo # 2……..20
- ✓ Grupo # 3……..20
- ✓ Grupo # 4……..35

3. Se organiza un comité de expertos con una cantidad mínima de 7 teniendo en cuenta su especialidad, grado científico y currículo.
4. Se les entrega a los expertos la propuesta para que estudien el tema a evaluar y dos modelos, uno para que valore el peso relativo de cada criterio y así poder calcular la concordancia entre los expertos (Anexo # 5) y otro para calcular el nivel de aceptación de la propuesta con una escala de 1 a 5 y la apreciación cualitativa con una clasificación final de la propuesta en excelente, bueno, aceptable, cuestionable y malo, (Anexos # 6). También se da la posibilidad de dar su opinión haciendo una valoración final de la propuesta, emitiendo todas aquellas consideraciones que estimaron convenientes.
5. Después de recibir los valores del peso relativo de cada criterio se construye la Tabla No.42
   - ✓ C es el número de criterios que van a evaluarse.
   - ✓ E el número de expertos que realizan la evaluación.
   - ✓ G: es el número del grupo al que pertenecen los criterios.

Tabla 6 Resultado del trabajo de expertos

G	C/E	E1	E2	E3	E4	E5	E6	E7	Ep
25	C1								
	C2								
	C3								
	C4								
20	C5								
	C6								
20	C7								

	C8								
35	C9								
	C10								
	C11								
	C12								
T									

6. Se verifica la consistencia en el trabajo de los expertos, para lo que se utiliza el coeficiente de concordancia de Kendall y el estadígrafo Chi cuadrado (X^2). Se sigue el procedimiento siguiente:

   ✓ Para cada criterio se determina:

   - ΣE: Sumatoria del peso dado por cada experto.
   - Ep: Puntuación promedio del peso dado por cada experto.
   - MΣE: media de los ΣE.
   - ΔC: Diferencia entre ΣE y MΣE.

   ✓ Se determina la desviación de la media, que posteriormente se eleva al cuadrado para obtener la dispersión (S) por la expresión:

   $$S = \Sigma (\Sigma E - \Sigma \Sigma E / C)^2$$

   ✓ Conociendo la dispersión se puede calcular el coeficiente de concordancia de Kendall (W):

   $$W = S / E^2 (C^3 – C)/ 12$$

   ✓ El coeficiente de concordancia de Kendall permite calcula r el Chi cuadrado real:

   $$X^2 = E (C-1) W$$

   ✓ Los valores obtenidos se muestran en la siguiente tabla.

Tabla 7 Tabla para el cálculo de concordancia de Kendall

Expertos/Criterios	E1	E2	E3	E4	E5	E6	E7	ΣE	Ep	ΔC	ΔC^2

C1											
C2											
C3											
C4											
C5											
C6											
C7											
C8											
CC9											
C10											
C11											
C12											
DC											
MΣE											
W											
X^2											

- ✓ El Chi cuadrado calculado se compara con el obtenido de las tablas estadísticas. Si se cumple: $X^2_{real} < X^2_{(\alpha, c-1),}$ existe concordancia en el trabajo de expertos

7. Si no existe concordancia se hace necesario repetir el trabajo de expertos. Una vez comprobada la consistencia del trabajo de expertos se puede determinar el nivel de aceptación de la propuesta entre los expertos, para esto se debe seguir los siguientes pasos:

    - ✓ Después de comprobar la consistencia del trabajo de expertos se puede definir el peso relativo de cada criterio (P).

- ✓ Conociendo el peso de cada criterio y la calificación dada por los evaluadores en una escala de 1 - 5 se puede construir la Tabla No.3, para obtener el valor de P × c., donde (c), es el criterio promedio concebido por los expertos.

Tabla 8 Tabla de calificación de cada criterio

Criterios	Clasificación					P	P x c
	1	2	3	4	5		
C1							
C2							
C3							
C4							
C5							
C6							
C7							
C8							
C9							
C10							
C11							
C12							

- ✓ Se calcula el Índice de Aceptación del proyecto (IA)

  IA = Σ (P × c) / 5.
- ✓ Por último se determina la probabilidad de éxito de la propuesta.

Tabla 9 Rangos predefinidos de Índice de Aceptación.

IA > 0,7	Existe alta probabilidad de éxito.
IA > 0,7 > 0,5	Existe probabilidad media de éxito.

0,5 > IA > 0,3	Probabilidad de éxito baja.
0,3 > IA	Fracaso seguro.

Para la aplicación de esta técnica se consultaron a 7 expertos para que dieran su opinión y valoraran la propuesta. La tabla de valores de peso se creó con los valores emitidos por cada uno de los expertos relativo a cada criterio (Anexo # 7).

Luego se llenaron los datos de la tabla para el cálculo de concordancia entre los expertos (Anexo # 8).

El resultado de los cálculos se expone a continuación:

X^2_{real} = 10,1876 para seleccionar X^2 de la tabla se toma 1 – α = 0.99, donde α es el error permisible, entonces α = 0.01. Debe cumplirse que $X^2_{real} < X^2_{(\alpha, c-1).}$ De esta forma quedaría:

10,1876 < 24,725 por tanto se puede afirmar que existe concordancia entre los expertos, lo que posibilita la creación de la tabla de clasificación de cada criterio para saber el índice de aceptación de la propuesta (Anexo # 9).

Luego de obtener todos los datos de la tabla se pasa a calcular el Índice de Aceptación la que arrojó el siguiente resultado:

0,68398 se evalúa en los rangos que aparecen en la tabla 45. Como resultado se obtuvo que: el proceso tiene probabilidad media de éxito.

2.4 Conclusiones parciales

1. Se precisó el proceso para la gestión de riesgos en la auditoría de la calidad de software, tomando como base los procedimientos que sustentaron la propuesta a partir del análisis realizado en el Capítulo 1.
2. Se detallaron cada una de las fases que sustentan el proceso de gestión de riesgo en la auditoría de la calidad del software, los roles, actividades y artefactos de entradas y salidas que intervienen en cada una de las etapas.
3. La propuesta del proceso Gestión de riesgos en la auditoría de la calidad del software, fue validada con la aplicación del método de experto. Al aplicar el método y analizar los resultados se obtuvo una probabilidad media de éxito por lo que la aplicación de la propuesta debe brindar resultados favorables.

3 Exploración y Planificación

En este capítulo se abordan temas referidos a las principales características del sistema en desarrollo, así como la propuesta de solución elaborada por el equipo de trabajo. Además, se identifican y definen los procesos del negocio, la descripción de las HU y se define el Plan de Iteración. Para dar solución a la problemática planteada en la investigación, se decidió crear una herramienta que permita gestionar los riesgos para la auditoría de la calidad, mediante una interfaz gráfica sencilla y profesional. Para ello la aplicación contará con los siguientes requisitos funcionales y no funcionales:

3.1 Historias de Usuario

Las HU son la técnica utilizada para especificar los requisitos del software. Es una forma rápida de administrar los requisitos definidos por el cliente sin tener que elaborar muchos documentos formales y sin requerir demasiado tiempo para administrarlos. Permiten responder rápidamente a los requisitos cambiantes. Son escritas en lenguaje coloquial y proveen detalles suficientes para hacer una estimación razonable del tiempo que llevará implementarlas (30). Cuentan de tres aspectos fundamentales:

- ✓ *Tarjeta:* se almacena suficiente información para identificar y detallar la historia.
- ✓ *Conversación:* cliente y programadores discuten la historia para ampliar los detalles (verbalmente cuando sea posible, pero documentada cuando se requiera confirmación).
- ✓ *Pruebas de aceptación:* permite confirmar que la historia ha sido implementada correctamente.

A continuación se muestran las HU más relevantes.

Tabla 10: HU_01 Gestionar usuario

Historia de Usuario	
Número: HU_01	**Nombre:** Gestionar usuario
Prioridad: Media	**Complejidad:** Media
Iteración: 1	**Estimación:** 1 semana
Descripción: el Administrador del sistema es el responsable de adicionar, eliminar, modificar y listar los usuarios de la aplicación, además de asignar los privilegios.	

Nota: tiene como objetivo fundamental la asignación de roles para los distintos usuarios.

Tabla 11: HU_02 Autenticar usuario

Historia de Usuario	
Número: HU_02	**Nombre:** Autenticar usuario
Prioridad: Media	**Complejidad:** Media
Iteración: 2	**Estimación:** 1 semana
Descripción: posee como objetivo principal que el usuario se autentique correctamente con su usuario y contraseña.	
Nota: el Administrador del sistema es el encargado de proporcionar la contraseña al usuario. Una vez autenticado dicho usuario tendrá la posibilidad de cambiar su contraseña y sus datos, si así lo desea.	

Tabla 12: HU_03 Gestionar auditoría

Historia de Usuario	
Número: HU_03	**Nombre:** Gestionar auditoría
Prioridad: Alta	**Complejidad:** Media
Iteración: 2	**Estimación:** 1 semana
Descripción: el administrador del sistema es el encargado de adicionar, eliminar, modificar y listar las auditorías además de asignarlas. Al adicionar o crear una auditoría se especifica la entidad a auditar, la fecha de inicio y fin, además del tipo de auditoría.	
Nota: tiene como objetivo fundamental planificar auditorías a los distintos usuarios.	

Tabla 13: HU_04 Mis auditorías

Historia de Usuario	
Número: HU_04	**Nombre:** Mis auditorías
Prioridad: Media	**Complejidad:** Media

Iteración: 2	**Estimación:** 1 semana
Descripción: el usuario después de acceder a la aplicación podrá ver sus auditorías planificadas.	
Nota: las auditorías planificadas para dicho usuario solo se mostrarán activas y se podrán realizar si están en el intervalo de fecha (fecha inicio y fecha fin), en caso contrario se mostrará deshabilitada.	

Tabla 14: HU_05 Gestionar riesgos

Historia de Usuario	
Número: HU_05	**Nombre:** Gestionar riesgos
Prioridad: Alta	**Complejidad:** Alta
Iteración: 1	**Estimación:** 1 semana
Descripción: los auditores líderes podrán adicionar, eliminar, modificar y listar los riesgos identificados para cada una de las auditorías. De cada riesgo se conoce su descripción, impacto, probabilidad de ocurrencia, causa, respuesta o acción a tomar, responsable de la ejecución de la respuesta y fecha de ejecución de la respuesta o acción.	

Tabla 15: HU_06 Analizar riesgos

Historia de Usuario	
Número: HU_06	**Nombre:** Analizar riesgos
Prioridad: Alta	**Complejidad:** Alta
Iteración: 1	**Estimación:** 3 semanas
Descripción: una vez identificados los riesgos de una auditoría el sistema podrá analizar los datos para la toma de decisión. Una vez analizados los riesgos la aplicación emite 3 tipos de respuestas: 1. La auditoría puede ejecutarse. 2. La auditoría es aplazada. 3. La auditoría es abortada.	

Tabla 16: HU_07 Prevención de Riesgos

Historia de Usuario	
Número: HU_07	**Nombre:** Prevención de Riesgos
Prioridad: Alta	**Complejidad:** Alta
Iteración: 2	**Estimación:** 3 semanas
Descripción: prevenir los riesgos es una funcionalidad que permite llenar todos los aspectos del riesgo: causas, acciones o respuestas, responsables de la tarea, duración de la tarea.	

Tabla 17: HU_08 Generar reporte

Historia de Usuario	
Número: HU_08	**Nombre:** Generar reporte
Prioridad: Media	**Complejidad:** Media
Iteración: 1	**Estimación:** 1 semana
Descripción: después de evaluados los riesgos la aplicación permitirá descargar la lista de riesgos identificados para cada una de las auditorías.	
Nota: los usuarios deben tener el mínimo privilegio absoluto necesario para realizar las tareas asignadas.	

Tabla 18: HU_09 Ver perfil

Historia de Usuario	
Número: HU_09	**Nombre:** Ver perfil
Prioridad: Media	**Complejidad:** Baja
Iteración: 1	**Estimación:** 1 semana
Descripción: después que el usuario está autenticado podrá acceder a su perfil donde consultará sus datos y podrá modificarlos.	
Nota: los usuarios tienen que estar autenticados en el sistema.	

Tabla 19: HU_10 Mostrar listado de riesgos por auditorías

Historia de Usuario	
Número: HU_10	**Nombre:** Mostrar listado de riesgos por auditorías
Prioridad: Media	**Complejidad:** Media
Iteración: 1	**Estimación:** 1 semana
Descripción: el usuario podrá ver el listado de riesgos identificados en cada auditoría realizada, en caso contrario se mostrará un mensaje de error, que no existen riesgos para esta auditoría.	
Nota: esta opción se podrá realizar en cualquier momento que el usuario autenticado lo desee, y se podrá acceder a dicho enlace existente en el menú principal.	

3.2 Plan de iteraciones

Una vez conocidas cada una de las HU y el esfuerzo que se requiere para desarrollar cada una de ellas, se procede a dividir el trabajo en iteraciones. Así se puede obtener un trabajo incremental donde la principal idea sea la comunicación en el equipo de trabajo, donde está insertado el cliente.

Para que exista un equilibrio entre las diferentes secuencias de trabajo se procede a dividir el proyecto en 3 iteraciones que se describen a continuación. Para ello se trató de hacer un balance para que cada iteración fuera escenario de conjuntos de HU que demandaran un desempeño aproximado del equipo de trabajo.

Iteración 1: se ejecutarán las tareas más sencillas y de menos costo computacional y requieran de menos esfuerzo por parte de los desarrolladores. Son las funcionalidades más independientes, que no requieran de otras funcionalidades para ejecutarse. Al terminar esta iteración se habrá concebido una idea más real de cómo será el sistema en sí pero de forma sencilla.

Iteración 2: se le dará cumplimiento a las HU de mayor complejidad. Haciendo que el sistema esté parcialmente listo. Esta es la etapa de trabajo en la que se necesita de mayor esfuerzo ya que exige mucho de los desarrolladores. Todas las historias que incidan en la lógica de la programación del sistema deben quedar implementadas.

La siguiente tabla muestra la distribución de las HU para cada iteración:

Tabla 20: Plan de iteraciones

Historias de Usuarios	Duración estimada (semanas)	Iteraciones	Duración de las iteraciones
Gestionar usuario	1	1	4 semanas
Autenticar usuario	1		
Gestionar auditoría	1		
Mis auditorías	1		
Gestionar riesgos	1	2	5 semanas
Analizar riesgos	2		
Prevención de riesgos	2		
Ver perfil	1	3	3 semanas
Mostrar listado de riesgos por auditorías	1		
Generar reporte	1		

3.3 Plan de entrega de versiones

Partiendo de las HU y el Plan de iteraciones se elabora el Plan de entrega (release) donde en cada iteración se elabora un producto entregable que contiene un incremento de funcionalidades superior a la entrega anterior. El Plan de entrega queda plasmado en la siguiente tabla.

Tabla 21: Plan de entrega de versiones

Historia de Usuario	Primera iteración	Segunda iteración	Tercera iteración
Gestionar usuario	V 1.0	Finalizado	Finalizado
Autenticar usuario	V 1.0	Finalizado	Finalizado
Gestionar auditoría	V 1.0-	Finalizado	Finalizado

Mis auditorías	V 1.0	Finalizado	Finalizado
Gestionar riesgos	-	V 1.0	Finalizado
Analizar riesgos	-	V 1.0	Finalizado
Prevención de riesgos	-	V 1.0	Finalizado
Ver perfil	-	-	V 1.0
Mostrar listado de riesgos por auditorías	-	-	V 1.0
Generar reporte	-	-	V 1.0

3.4 Conclusiones parciales

Al finalizar el presente capítulo se obtuvieron de forma detallada las etapas de Exploración y Planificación, con la descripción de los artefactos:

- ✓ HU
- ✓ Plan de iteraciones.
- ✓ Plan de entrega de versiones (release).

4 Diseño, Implementación y Pruebas

En este capítulo se describen los artefactos generados en las fases de diseño e implementación del sistema, además de las pruebas realizadas al mismo. Al comienzo de una iteración cada historia de usuario se fragmenta en diferentes tareas de programación.

4.1 Diseño del sistema

Un prototipo es una visión preliminar del sistema, es un modelo ampliable y modificable, que tiene todas las características que hasta el momento debe tener el sistema.

La Figura 10 muestra el prototipo de interfaz no funcional que constituyó el punto de partida de la interfaz inicial que presenta la aplicación.

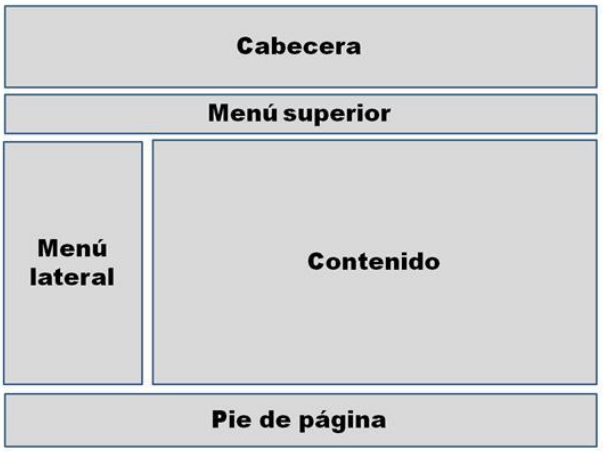

Figura 10: Prototipo de Interfaz no funcional. (Fuente elaboración: propia)

XP establece prácticas especializadas que inciden directamente en la realización del diseño para lograr un sistema reutilizable tratando de mantener su simplicidad y crear un diseño de fácil modificación que permita hacer entregas pequeñas y frecuentes al cliente. XP no especifica ninguna técnica de modelado, pueden utilizarse indistintamente sencillos esquemas en una pizarra, diagramas de clases utilizando UML o tarjetas CRC (Clase, Responsabilidad y Colaboración) siempre que sean útiles, tributen a la comprensión y no requieran mucho tiempo en su creación (18).

4.1.1 Tarjetas CRC

Las tarjetas CRC son una metodología para el diseño de software orientado a objetos. Estas tarjetas se dividen en tres secciones que contienen la información del nombre de la clase, sus responsabilidades y sus colaboradores. Una clase es cualquier persona, evento, concepto, pantalla o reporte. Las responsabilidades de una clase son las acciones que realiza. Los colaboradores son las demás clases con las que trabaja en conjunto para llevar a cabo sus responsabilidades (31).

Constituyen una primera aproximación a los objetos que luego se van a utilizar, son una herramienta para estimar si el conjunto de clases obtenidas responden bien a las necesidades dinámicas del sistema (32). A continuación se muestran las Tarjetas CRC de la aplicación:

Tabla 22: Tarjeta CRC UsuarioController

Tarjeta CRC	
Nombre de la clase: UsuarioController.	
Responsabilidades	**Colaboradores**
Tiene la responsabilidad de adicionar, modificar, eliminar y listar usuarios.	✓ BaseUser ✓ Usuario

Tabla 23: Tarjeta CRC DefaultController

Tarjeta CRC	
Nombre de la clase: DefaultController.	
Responsabilidades	**Colaboradores**
Esta clase va a tener la responsabilidad de permitir autenticar los usuarios y mostrarle su perfil.	✓ Usuario ✓ BaseUser

Tabla 24: Tarjeta CRC AuditoriaController

Tarjeta CRC
Nombre de la clase: AuditoriaController

Responsabilidades	Colaboradores
Permite crear, editar, modificar y listar las auditorías. Cuando se crea una auditoría es asignada al auditor líder que la va a realizar.	✓ AuditoriaRepository ✓ Riesgo ✓ Tipo de auditoría ✓ Auditoría

Tabla 25: Tarjeta CRC AuditoriaRepository

Tarjeta CRC	
Nombre de la clase: AuditoriaRepository	
Responsabilidades	**Colaboradores**
Su principal objetivo es realizar consultas a la base de datos.	✓ EntityRepository

Tabla 26: Tarjeta CRC RiesgoControlller

Tarjeta CRC	
Nombre de la clase: RiesgoControlller	
Responsabilidades	**Colaboradores**
Esta clase tiene la responsabilidad de adicionar, modificar, eliminar, listar e identificar los distintos riesgos existentes en una auditoría.	✓ Riesgo ✓ Auditoría ✓ Causa ✓ Impacto ✓ Probabilidad ✓ Tiempo ✓ Respuesta ✓ Usuario ✓ AuditoriaRepository

4.2 Implementación

La implementación de la aplicación estuvo basada en estándares de programación que permiten la consistencia de este. Además, facilita su comprensión y escalabilidad. La programación en pareja es uno de los principios de la metodología XP permitiendo hacer más eficiente el código.

4.2.1 Patrones arquitectónicos

Symfony 2.0 basa su funcionamiento en el patrón arquitectónico Modelo Vista Controlador (MVC) que obliga a organizar el código de acuerdo a sus convenciones. Aunque su creador Fabien Potencier (33) apunta que Symfony 2.0 no es un framework MVC, solo proporciona las herramientas para trabajar en las partes de Control y Vista. La parte de Modelo es responsabilidad del desarrollador (34).

- ✓ *Modelo:* representación específica de la información con que el sistema opera, gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.
- ✓ *Controlador:* responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información. Se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo' (Figura 11).
- ✓ *Vista:* presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida.

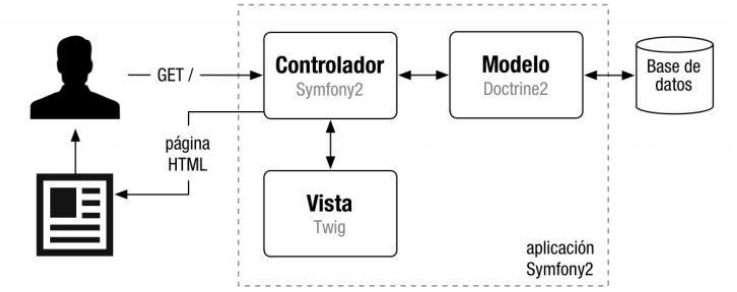

Figura 11: Modelo Vista Controlador (MVC). (Fuente elaboración: (34))

4.2.2 Patrones de diseño

En el diseño de software orientado a objeto es importante el uso de los patrones GRASP[19], ya que solucionan muchos de los problemas que se pueden presentar a la hora de programar. Para el diseño de la solución se tuvieron en cuenta algunos de los patrones de diseño como: Experto, Creador y Controlador, además Front Controller (Controlador Frontal) y Decorator (Envoltorio) como patrones GoF[20].

GRASP

- ✓ *Experto:* define el principio básico de asignación de responsabilidades. Indica que la responsabilidad de la creación de un objeto debe recaer sobre la clase que conoce toda la información necesaria para crearlo. Con la utilización de este patrón se definió dónde colocar en cada clase las funcionalidades que necesitan de esa información. Este patrón se ve identificado en las clases *AuditoriaController.php* y *RiesgoController.php*, pues estás clases controladoras presentan las principales funcionalidades a cada clase. Es uno de los patrones que más se utiliza cuando se trabaja con Symfony 2.0, con la inclusión de la librería Doctrine para mapear la Base de Datos. Utiliza esta librería para realizar su capa de abstracción en el modelo, encapsular toda la lógica de los datos y generar las clases con todas las funcionalidades comunes de las entidades.
- ✓ *Creador:* permite identificar quién debe ser el responsable de la instanciación de nuevos objetos o clases. Este patrón se ve identificado en la clase *AudiriaController.php*, pues contiene objetos creados e instanciados de la clase *Riesgo.php* para controlar los datos de los riesgos en una auditoría.
- ✓ *Controlador:* se utiliza como intermediario entre cada una de las capas, de forma tal que garantice la comunicación entre los eventos externos del sistema en la capa de presentación y los componentes de la capa de negocio, declara el constructor de clase como privado para que no sea instanciable directamente. Se puede identificar en las clases *RiesgoController.php*, *AuditoriaController.php* y *UsuarioController.php*, pues van hacer las encargadas de controlar y dar respuestas a las peticiones del usuario.

GoF

[19]General Responsibility Assignment Software Patterns (Patrones Generales de Software para Asignar Responsabilidades).

[20]Gang of Four (Banda de los cuatro).

- ✓ *Front Controller (Controlador Frontal):* es un patrón de diseño web usado como único punto de entrada a la aplicación. Realiza tareas comunes a todos los controladores: manejo de la seguridad, de las peticiones de los usuarios, carga de la configuración de la aplicación y delega la responsabilidad de responder a las peticiones al módulo específico que tiene la acción enviada en la petición. Este patrón se ve identificado en la clase *app.php***,** es la clase controladora que atiende todas las solicitudes del usuario y de brindarle respuesta mediante los controladores de cada entidad, donde cada petición va hacer asignada a un controlador para darle respuesta a su petición.
- ✓ *Decorator (Envoltorio):* normalmente, en las aplicaciones web existen contenidos que son comunes en todas las páginas que conforman la aplicación. Symfony haciendo uso de este patrón de diseño estructural delimita el código común en todas las páginas definido en un archivo global para todas las vistas denominado layout del código HTML generado como respuesta a una petición determinada. El contenido se integra en el layout, por lo que se puede afirmar que este decora la plantilla. Symfony 2.0 contienen un decorador que permite agregar funcionalidades dinámicamente a las aplicaciones desarrolladas bajo sus principios. Cada una de las vistas generadas heredan su diseño de la plantilla *layout.html.twig***.**

La Figura 12 muestra el modelo de datos del sistema que se diseñó para el manejo de los datos en el sistema.

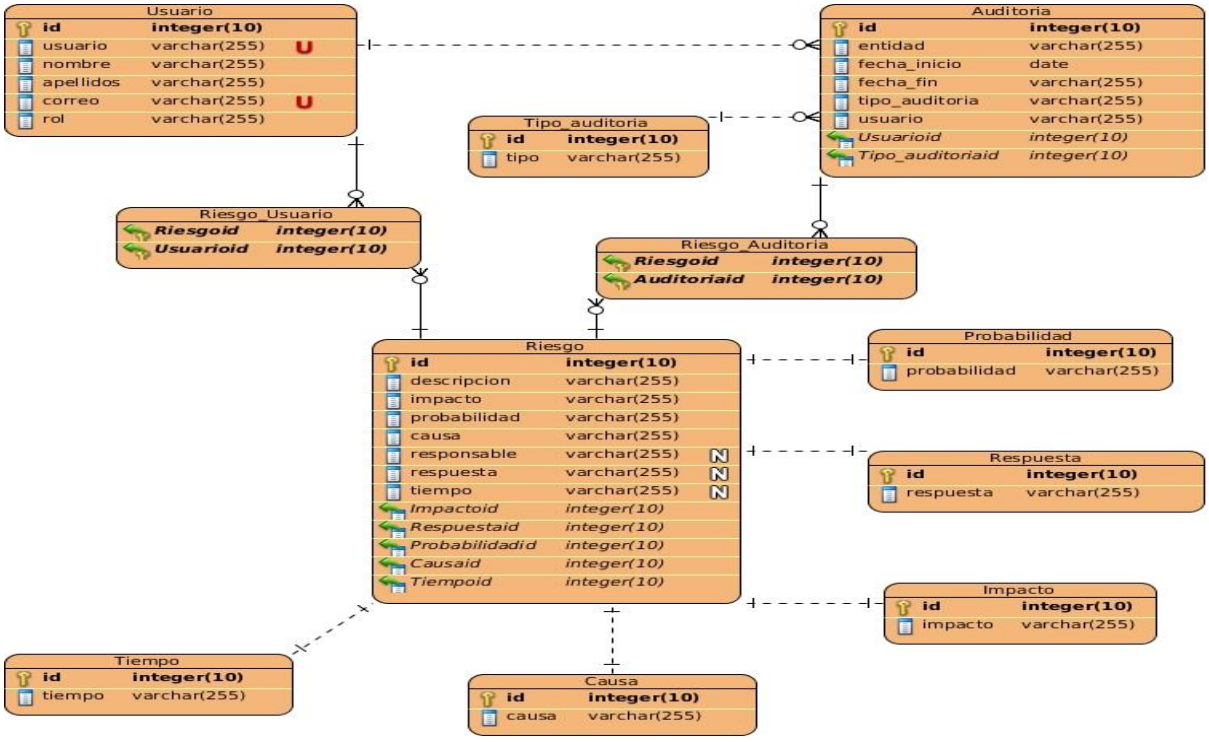

Figura 12: Modelo de datos. (Fuentes de elaboración: propia)

4.2.3 Usuarios de la aplicación

En la creación de un sistema es importante definir los usuarios que harán uso del mismo. También especificar los roles que asumirán, además de las funciones que realizarán estos usuarios en la aplicación. Estos aspectos se detallan a continuación:

Tabla 27: Usuarios de la aplicación

Usuarios	Privilegios
✓ Administrador del sistema	Tiene acceso a todas las funcionalidades de la aplicación. Además, gestiona toda la información que se maneja en el sistema.
✓ Auditor líder	Podrá ejecutar las acciones definidas para su rol.

4.2.4 Tareas de implementación por iteraciones

En esta fase XP propone desglosar las HU en tareas con el objetivo de facilitar el trabajo, por lo que durante el transcurso de cada una de las iteraciones se planifican las tareas a realizar para dar solución a las HU correspondientes a estas. Cada tarea se debe desarrollar en un período de uno a tres días de desarrollo.

Tabla 28: Descripción de las Tareas de implementación por cada HU

Historia de Usuario	Tareas de implementación
Iteración 1	
Gestionar usuario.	1. Adicionar usuario 2. Modificar usuario 3. Listar usuario 4. Eliminar usuario
Autenticar usuario	1. Autenticar usuario
Gestionar auditoría	1. Adicionar auditoría 2. Modificar auditoría 3. Listar auditoría 4. Eliminar auditoría
Mis auditorías	1. Mis auditorías
Iteración 2	
Gestionar riesgos	1. Adicionar riesgos 2. Modificar riesgos 3. Listar riesgos 4. Eliminar riesgos
Analizar riesgos	1. Identificar riesgo 2. Priorizar riesgo

	3. Listar riesgos priorizados
Prevención de riesgos	1. Adicionar responsable 2. Acción a tomar 3. Duración de la tarea
Iteración 3	
Generar reporte	1. Mostrar resultados
Ver perfil	1. Ver perfil
Mostrar listado de riesgos por auditorías	1. Mostrar listado de riesgos por auditorías

Tabla 29: TI_01 Adicionar usuario

Tarea de implementación	
Número Tarea: TI_01	**Historia de Usuario :** HU_01
Nombre Tarea: Adicionar usuario	
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 0.2
Fecha Inicio: 25/2/2013	**Fecha Fin:** 26/2/2013
Descripción: el administrador del sistema es el encargado de adicionar los usuarios con sus datos y después les transmitirá su contraseña para acceder a la aplicación informática, donde una vez autenticado dicho usuario podrá cambiar su contraseña.	

Tabla 30: TI_01 Modificar usuario

Tarea de implementación	
Número Tarea: TI_02	**Historia de Usuario:** HU_01
Nombre Tarea: Modificar usuario	
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 0.1

Fecha Inicio: 26/2/2013	**Fecha Fin:** 27/2/2013
Descripción: permitir al usuario cambiar sus datos una vez autenticado en la aplicación.	

Tabla 31: TI_01 Eliminar usuario

Tarea de implementación	
Número Tarea: TI_03	**Historia de Usuario:** HU_01
Nombre Tarea: Eliminar usuario	
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 0.1
Fecha Inicio: 27/2/2013	**Fecha Fin:** 28/2/2013
Descripción: el administrador del sistema es el encargado de eliminar los datos de cada usuario una vez que esté autenticado.	

Tabla 32: TI_01 Listar usuario

Tarea de implementación	
Número Tarea: TI_04	**Historia de Usuario:** HU_01
Nombre Tarea: Listar usuario	
Tipo de Tarea: Desarrollo	**Puntos Estimados:**0.1
Fecha Inicio: 28/2/2013	**Fecha Fin:** 1/3/2013
Descripción: el usuario con rol (ROLE_ADMINISTRADOR) puede listar los usuarios que se encuentran en la base de datos.	

Tabla 33: TI_01 Autenticar usuario

Tarea de implementación	
Número Tarea: TI_01	**Historia de Usuario:** HU_02
Nombre Tarea: Autenticar usuario	

Tipo de Tarea: Desarrollo	**Puntos Estimados:** 0.5
Fecha Inicio: 4/3/2013	**Fecha Fin:** 8/3/2013
Descripción: el usuario para acceder a la aplicación llena los campos y el sistema verifica que se encuentre en la base de datos.	

Tabla 34: TI_01 Adicionar auditoría

Tarea de implementación	
Número Tarea: TI_01	**Historia de Usuario:** HU_03
Nombre Tarea: Adicionar auditoría	
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 0.2
Fecha Inicio: 11/3/2013	**Fecha Fin:** 12/3/2013
Descripción: el usuario con rol (ROLE_ADMINISTRADOR) es el encargado de crear y planificar las auditorías para cada usuario.	

Tabla 35: TI_02 Modificar auditoría

Tarea de implementación	
Número Tarea: TI_02	**Historia de Usuario:** HU_03
Nombre Tarea: Modificar auditoría	
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 0.1
Fecha Inicio: 12/3/2013	**Fecha Fin:** 13/3/2013
Descripción: el usuario con rol (ROLE_ADMINISTRADOR) es el encargado de crear y planificar las auditorías para cada usuario.	

Tabla 36: TI_03 Eliminar auditoría

Tarea de implementación

Número Tarea: TI_03	**Historia de Usuario:** HU_03
Nombre Tarea: Eliminar auditoría	
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 0.1
Fecha Inicio: 13/3/2013	**Fecha Fin:** 14/3/2013
Descripción: el usuario con rol (ROLE_ADMINISTRADOR) es el encargado de eliminar los datos de cada auditoría.	

Tabla 37: TI_04 Listar auditoría

Tarea de implementación	
Número Tarea: TI_04	**Historia de Usuario:** HU_03
Nombre Tarea: Listar auditoría	
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 0.1
Fecha Inicio: 14/3/2013	**Fecha Fin:** 15/3/2013
Descripción: el usuario con rol (ROLE_ADMINISTRADOR) es el encargado de mostrar el listado de todas las auditorías.	

Tabla 38: TI_01 Adicionar riesgo

Tarea de implementación	
Número Tarea: TI_01	**Historia de Usuario:** HU_04
Nombre Tarea: Mis auditorías	
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 0.5
Fecha Inicio: 18/3/2013	**Fecha Fin:** 22/3/2013
Descripción: el usuario una vez autenticado podrá acceder al menú principal y ver el listado de riesgos por auditorías.	

Tabla 39: TI_01 Adicionar riesgo

Tarea de implementación	
Número Tarea: TI_01	**Historia de Usuario:** HU_05
Nombre Tarea: Adicionar riesgo	
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 0.2
Fecha Inicio: 25/3/2013	**Fecha Fin:** 26/3/2013
Descripción: el usuario podrá adicionar los distintos riesgos que cree que estarán presentes en una auditoría.	

4.3 Pruebas

XP propone realizar dos tipos de pruebas al sistema, llamadas pruebas unitarias y pruebas de aceptación. Las pruebas unitarias son las que se definen antes de realizar la implementación del código. Las pruebas de aceptación son creadas a partir de las HU, estas permiten determinar si cada HU ha sido implementada satisfactoriamente.

4.3.1 Pruebas de Aceptación

Las Pruebas de Aceptación (PA) son consideradas pruebas de caja negra y cada prueba representa una salida esperada del sistema. Como se mencionó anteriormente estas son creadas a partir de HU, con el objetivo de inspeccionar que cada una de ellas se haya desarrollado correctamente. El cliente, es el máximo responsable de verificar que los resultados de estas pruebas sean correctos. Una HU puede tener todas las PA que sean necesarias para lograr determinar su completo funcionamiento y a su vez, no puede ser considerada terminada hasta que no se le realicen sus pruebas pertinentes.

Pruebas de caja negra

Se parte de los requisitos funcionales, para diseñar pruebas que se aplican sobre el sistema sin necesidad de conocer cómo está construido por dentro. Las pruebas se aplican sobre la aplicación utilizando un conjunto de datos de entrada y comparándolos con las salidas que se producen para verificar que la función se está ejecutando correctamente. Las herramientas básicas son observar la funcionalidad y contrastar con la especificación (36).

A continuación se muestran los resultados de las PA realizadas al software. Las pruebas de aceptación contienen los siguientes campos:

Nombre: debe ser descriptivo en la medida de lo posible.

Descripción: se describe qué es lo que se desea probar. La descripción debe ser corta y precisa.

Condiciones de ejecución: condiciones especiales que deben tenerse en cuenta para ejecutar el caso de prueba.

Entradas: entradas al caso de prueba en caso de necesitarlas.

Resultado esperado: resultado que se desea tenga el caso de prueba. Descripción breve de lo que debe suceder.

Evaluación: se evalúa si el caso de prueba tuvo éxito o no. En caso de ser exitoso se asigna un resultado de satisfactorio, en caso contrario insatisfactorio.

Tabla 40: PA_01 Adicionar usuario

Prueba de Aceptación	
Código: PA_01	**Historia de Usuario:** HU_01
Nombre: Adicionar usuario	
Descripción: el usuario con rol Administrador será el encargado de crear el usuario, donde llenará todos sus datos y si son válidos se guardará en la base de datos, mostrando un mensaje informando que el usuario se ha creado correctamente.	
Condiciones de Ejecución: el usuario con rol administrador debe estar autenticado.	
Entrada / Pasos de ejecución: el administrador del sistema selecciona la opción *Usuario,* que aparece en Menú superior. Se mostrará un listado con todos los usuarios adicionados en la base de datos. Para adicionar un nuevo usuario selecciona el botón *Adicionar* y se muestra un formulario donde podrá llenar los campos requeridos para esta operación. Una vez completado todos los campos el usuario selecciona la opción *Guardar***,** si los campos son válidos le mostrará un mensaje: “El usuario ha sido creado satisfactoriamente”.	

Resultado Esperado: el sistema le permite al usuario que se cree correctamente si los campos a llenar son válidos.
Evaluación de la Prueba: Satisfactoria

Tabla 41: PA_02 Modificar usuario

Prueba de Aceptación	
Código: PA_02	**Historia de Usuario:** HU_01
Nombre: Modificar usuario	
Descripción: el administrador del sistema podrá modificar los datos de los usuarios.	
Condiciones de Ejecución: el usuario debe estar autenticado.	
Entrada / Pasos de ejecución: el administrador del sistema selecciona la opción *Usuario,* que aparece en Menú superior. Se mostrará un listado con todos los usuarios adicionados en la base de datos. Para modificar un nuevo usuario selecciona el botón *Editar* y se muestra un formulario donde podrá llenar los campos requeridos para esta operación. Una vez completado todos los campos el usuario selecciona la opción *Actualizar***,** si los campos son válidos le mostrará un mensaje: ¨El usuario se ha actualizado correctamente¨.	
Resultado Esperado: el sistema le permite al usuario cambiar todos sus datos si así lo desea y guardarlos en la base de datos si son correctos.	
Evaluación de la Prueba: Satisfactoria	

Tabla 42: PA_02 Eliminar usuario

Prueba de Aceptación	
Código: PA_03	**Historia de Usuario:** HU_01
Nombre: Eliminar usuario	
Descripción: el usuario con rol Administrador será el encargado de eliminar el usuario, donde se perderán todos sus datos.	

Condiciones de Ejecución: el usuario con rol administrador debe estar autenticado.
Entrada / Pasos de ejecución: el administrador del sistema selecciona la opción *Usuario,* que aparece en Menú superior. Se mostrará un listado con todos los usuarios adicionados en la base de datos. Para eliminar un nuevo usuario selecciona el botón *Eliminar***.**
Resultado Esperado: el sistema elimina correctamente el usuario de la base de datos.
Evaluación de la Prueba: Satisfactoria

Tabla 43: PA_02 Listar usuario

Prueba de Aceptación	
Código: PA_04	**Historia de Usuario:** HU_01
Nombre: Listar usuarios	
Descripción: el usuario con rol Administrador será el encargado de listar y ver todos los usuarios creados en la base de datos con todos sus datos.	
Condiciones de Ejecución: el usuario con rol administrador debe estar autenticado.	
Entrada / Pasos de ejecución: el administrador del sistema selecciona la opción *Usuario,* que aparece en Menú superior. Se mostrará un listado con todos los usuarios adicionados en la base de datos.	
Resultado Esperado: el sistema muestra todos los usuarios de la base de datos.	
Evaluación de la Prueba: Satisfactoria	

Tabla 44: PA_05 Autenticar usuario

Prueba de Aceptación	
Código: PA_05	**Historia de Usuario:** HU_02
Nombre: Autenticar usuario	
Descripción: el usuario accede a la aplicación llenando los campos para ello. La aplicación verifica que los datos sean correctos y que el usuario se encuentre en la base de datos. Si esto sucede el usuario tendrá acceso a la aplicación, sino aparecerá un cartel requiriendo los datos correctamente.	

Condiciones de Ejecución: el usuario con rol administrador debe estar autenticado.
Entrada / Pasos de ejecución: la aplicación debe brindar la opción de autenticar.
Resultado Esperado: el sistema permite que el usuario pueda acceder a la aplicación si está registrado en ella.
Evaluación de la Prueba: Satisfactoria

Tabla 45: PA_06 Adicionar auditoría

Prueba de Aceptación	
Código: PA_06	**Historia de Usuario:** HU_03
Nombre: Adicionar auditoría	
Descripción: el usuario con rol Administrador será el encargado de crear y planificar cada una de las auditorías además de asignarlas a un usuario.	
Condiciones de Ejecución: el usuario con rol administrador debe estar autenticado.	
Entrada / Pasos de ejecución: el administrador del sistema selecciona la opción *Auditoría,* que aparece en el Menú lateral. Se mostrará un listado con todas las auditorías adicionadas en la base de datos. Para adicionar una nueva auditoría selecciona el botón *Adicionar***,** mostrándose un formulario donde podrá llenar los campos requeridos para esta operación. Una vez completado todos los campos el usuario, selecciona la opción *Guardar***,** si los campos son válidos le mostrará un mensaje: ¨La auditoría se ha adicionado correctamente¨.	
Resultado Esperado: el sistema permite que la auditoría se cree correctamente.	
Evaluación de la Prueba: Satisfactoria	

Resultados de las pruebas

Las pruebas se realizaron en tres iteraciones, detectándose para la primera siete no conformidades significativas, cinco no significativas y tres recomendaciones.

Para la segunda iteración, tres se detectaron no conformidades significativas, cuatro no significativas y dos recomendaciones.

Para la tercera y última iteración, se detectó una no conformidad significativa, una no significativa y dos recomendaciones.

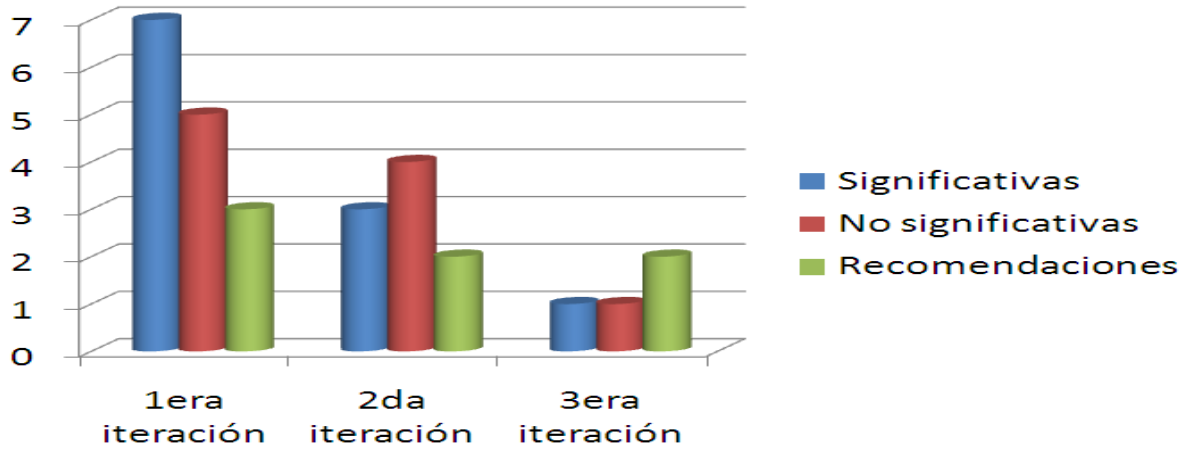

Figura **13:** Resultado de las pruebas por iteración**.**

Las no conformidades (NC) no significativas se centraron en errores ortográficos como: omisiones de tildes, paréntesis, cambio de mayúscula por minúscula y las significativas, en errores de validación y cambios en el diseño. Las recomendaciones estaban dirigidas a mejoras en el diseño, seguridad y escalabilidad.

Las NC encontradas luego de concluida cada iteración de pruebas, fueron analizadas por el equipo de desarrollo para corregir los errores detectados, lo que contribuyó a mejorar la calidad y funcionalidad del software.

4.4 Conclusiones parciales

1. La realización de las tareas de ingeniería que propone la metodología de desarrollo XP permitió el desarrollo de la aplicación para la gestión de riesgos en una auditoría de la calidad de software de manera organizada y teniendo en cuenta las mejores prácticas que propone esta metodología.
2. La realización de las pruebas por iteraciones permitió obtener un producto con mayor calidad y todas las No conformidades detectadas en cada iteración fueron resueltas. También se tuvo en cuenta por parte del equipo de desarrollo las recomendaciones que arrojó la etapa de pruebas.

Conclusiones generales

Al finalizar la presente investigación se arribaron a las siguientes conclusiones:

1. El estudio realizado permitió profundizar en los aspectos referentes a la gestión de riesgos, los modelos, normas y políticas. Además de los conceptos concernientes a las auditorías de la calidad, que contribuyeron al desarrollo de la presente investigación.
2. La selección de las herramientas, metodología de desarrollo, lenguajes de programación y tecnologías más apropiadas permitieron dar cumplimiento al objetivo general.
3. El diseño de la propuesta del procedimiento apoyó la planificación de las actividades de gestión de riesgos a desarrollar, identificar los riesgos que podían afectar los componentes u objetivos del proceso, analizar y priorizar los mismos, definir las acciones para contrarrestar el impacto aparejado a cada riesgo y desarrollar un seguimiento y control sobre los riesgos que se materialicen en la ejecución de la auditoría de la calidad del software.
4. La aplicación informática contribuirá a agilizar el proceso de gestión de riesgos en las auditorías de la calidad de software, además de apoyar al personal auditor en la toma de decisiones.
5. Las diferentes iteraciones de pruebas de aceptación realizadas a la aplicación, demostraron que la misma satisface las especificaciones definidas por el cliente. Además para la validación de la propuesta se aplicó el criterio de expertos el cual arrojó como resultado que la probabilidad de éxito es media.

Para la futura explotación y puesta en práctica de la aplicación para la gestión de riesgos en una auditoría, a los auditores y desarrolladores de software de Calisoft se les recomienda lo siguiente:

1. Perfeccionar las técnicas o mecanismos para realizar cada una de las etapas definidas en el proceso de gestión de riesgos en la auditoría de la calidad.
2. Incluir nuevas funcionalidades a la aplicación, relacionadas con el control estadístico en el tratamiento de los riesgos, que permita a los auditores verificar el comportamiento de cada uno de los riesgos.

Referencia bibliográfica

1. Española, Real Academia. RAE Real Academia Española. [En línea] RAE Real Academia Española, Junio de 2012. [Citado el: 10 de 01 de 2013.] http://www.rae.es.

2. Cortez, Liliana Uriarte. Scribd. [En línea] 23 de Mayo de 2011. [Citado el: 15 de 11 de 2012.] http://es.scribd.com/doc/56037001/factores-de-riesgo-y-dano-para-la-salud-vrdadero**.**

3. Carnegie Mellon University. Software Engineering Institute . [En línea] 2013. [Citado el: 15 de 02 de 2013.] http://www.sei.cmu.edu/.

4. Montesino, Miguel. PMI. [En línea] Noviembre de 2011. [Citado el: 22 de febrero de 2013.] http://www.pmi.org/.

5. Pressman, Roger S. Ingeniería de Software. Un enfoque práctico (Sexta Edición). 2005.

6. Yeleny Zulueta, Eder Despaigne, Anaisa Hernández. La gestión de riesgos en la producción de software y la formación de profesionales de la informática: experiencias de una universidad cubana. España : Revista Española de Innovación, Calidad, 2009. ISSN (Versión electrónica): 1885-4486.

7. Infante, Liuba María. Disminución de la exposición a los riesgos en el proyecto. Sistema de Información Geográfica de la Universidad de las Ciencias Informáticas a través de la aplicación de un modelo formal de Gestión de Riesgos. La Habana : s.n., 2009.

8. PMI - Project Management Institute. Guía de los Fundamentos de la Dirección de Proyectos (Guía del PMBOK®). Pennsylvania, Estados Unidos Americanos : s.n., 2004. ANSI/PMI 99-001-2004.

9. Justicia, Ministerio de. Gaceta Oficial de la República de Cuba. República de Cuba : s.n., 2011. ISSN 1682-7511.

10. Yandielys Reyes Plano, Ing. Yurisbel Vega Ortiz.. Aplicación y mejora del Modelo de Gestión de Riesgos “MoGeRi” al proyecto “Captura y Catalogación de Medias”. La Habana : s.n., 2009.

11. buenas tareas. [En línea] Agosto de 2010. [Citado el: 5 de 12 de 2012.] http://www.buenastareas.com/ensayos/Metodos-Revision-De-Evidencia-Auditoria-Informatica/614650.html.

12. Acosta Molina, Ing. Dialexis. Diseño e implementación del proceso de auditoría de la calidad para la actividad productiva en la universidad de las ciencias informáticas (UCI). Ciudad de La Habana : s.n., 2008-2009.

13. Norma Cubana ISO 19011: 2012. Directrices para la auditoría de los sistemas de gestión. La Habana : ISO, 2011.

14. Jurán, Joseph. Manual del Control de la Calidad. 1987**.**

15. MAGERIT – versión 3.0. Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información. s.l. : Subdirección General de Información, Documentación y Publicaciones, Centro de Publicaciones, 2012. Vol. Libro I. NIPO: 630-12-171-8.

16. Sistemas Informáticos. Sistemas Informáticos. [En línea] 2005. [Citado el: 11 de Marzo de 2013.] http://www.sitsoft.com.ar/Audita.asp.

17. Canós, José H., Letelier, Patricio y Penad, Mª Carmen. Métodologías Ágiles en el Desarrollo de Software. [En línea] [Citado el: 23 de Marzo de 2013.] http://noqualityinside.com.ar/nqi/nqifiles/XP_Agil.pdf.

18. Escribano, Gerardo Fernández. Introducción a eXtreme Programming.

19. Rodríguez Corbea, Maite y Ordóñez Pérez, Meylin. La metodología XP aplicable al desarrollo del software educativo en Cuba. Ciudad de la Habana : s.n., 2007.

20. Gutierrez, Jorge A. Saavedra. El Mundo Informático. [En línea] http://jorgesaavedra.wordpress.com/2007/05/05/lenguajes-de-programacion/**.**

21. LIBROSWEB. [En línea] 2013. [Citado el: 3 de Febrero de 2013.] http://www.librosweb.es/symfony_1_2/capitulo_1/symfony_en_pocas_palabras.html.

22. Carrera, Cristian. PSD a HTML Paso a Paso. [En línea] [Citado el: 21 de Marzo de 2013.] http://www.psdahtmlpasoapaso.com/blog/5-razones-para-usar-html5.

23. Introducción a Symfony. [En línea] [Citado el: 3 de 12 de 2012.] http://www.librosweb.es/symfony_1_2/capitulo_1/symfony_en_pocas_palabras.html.

24. PHP. [En línea] 2001. [Citado el: 26 de Abril de 2013.] http://php.net.

25. Álvarez, Miguel Angel. desarrolloweb.com. [En línea] [Citado el: 11 de 12 de 2012.] http://www.desarrolloweb.com/#_blank.

26. Cursos de Informática. [En línea] http://nilossgp.blogspot.com/2012/10/entornos-de-desarrollo.html**.**

27. Limonta Gutiérrez, Dayron. Extensión del IDE Netbeans para el desarrollo de aplicaciones empleando el marco de trabajo Sauxe. La Habana : Universidad de las Ciencias Informáticas , 2012.

28. PostGreSQL vs. MySQL. [En línea] [Citado el: 03 de 01 de 2012.] http://www.danielpecos.com/docs/mysql_postgres/index.html.

29. Carmona, M. La Auditoría Interna de Gestión: Aspecto Teóricos. El Caso Particular Cubano . La Auditoría Interna de Gestión: Aspecto Teóricos. El Caso Particular Cubano . España, UNIVERSIDAD DE HUELVA : s.n., 1998.

30. Quijano, Juan. GENBETA:dev desarrollo y software. GENBETA:dev desarrollo y software. [En línea] 28 de febrero de 2012. [Citado el: 10 de 04 de 2013.] https://sites.google.com/a/uji.es/gesproin/.

31. Letelier, Patricio y Carmen Penadés, Mª. Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP). Valencia, España : s.n., 2009.

32. Jummp. [En línea] 10 de 01 de 2012. http://jummp.wordpress.com/2012/01/10/desarrollo-de-software-tarjetas-crc/.

33. Potencier, Fabien. What is Symfony2? What is Symfony2? [En línea] 25 de October de 2011. [Citado el: 9 de 04 de 2013.] fabien.potencier.org/article/49/what-is-symfony2**.**

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

35. Larman, Craig. UML y patrones. . 1999. ISBN 970-1 7-0261-1.

36. Ingeniero de Gestión. [En línea] 20 de Agosto de 2012. [Citado el: 10 de Febrero de 2013.] http://ingenierogestion.blogspot.com/2009/06/pruebas-de-caja-negra-y-caja-blanca.html.

37. León, R.A.H. El paradigma cuantitativo de la investigación científica. 2002. p.959-16-0343-6.

Bibliografía

Carnegie Mellon University. 2013. Software Engineering Institute. Software Engineering Institute. [En línea] 2013. [Citado el: 03 de 02 de 2013.] http://www.sei.cmu.edu/.

19011:2012, Norma Cubana ISO. 2011. Directríces para la auditoría de los sistemas de gestión. La Habana : s.n., 2011.

2011. Directríces para la auditoría de los sistemas de gestión. 2011.

Alfaro, Econ. Félix Murillo. 1999. Instituto nacional de estadística e informática. [en línea] noviembre de 1999. [citado el: 27 de noviembre de 2012.] Http://www.inei.gob.pe/biblioineipub/bancopub/inf/lib5103/libro.pdf.

Álvarez, Miguel Angel desarrolloweb.com. desarrolloweb.com. [En línea] [Citado el: 11 de 12 de 2012.] http://www.desarrolloweb.com/#_blank.

2005. AUDITA - Software para Administración de Auditorías con orientación a Riesgos. AUDITA - Software para Administración de Auditorías con orientación a Riesgos. [En línea] 2005. [Citado el: 12 de febrero de 2013.] http://www.sitsoft.com.ar/Audita.asp.

Baldají, Delisay Susé. 2007. Propuesta de procedimiento para el desarrollo y aplicación de la Gestión del Riesgo en proyectos de producción de software. Propuesta de procedimiento para el desarrollo y aplicación de la Gestión del Riesgo en proyectos de producción de software. La Habana : s.n., 2007.

Bolivariana, Universidad Union. Ingeniería de Software. Ingeniería de Software. [En línea] [Citado el: 20 de 03 de 2013.] http://ingenieriadesoftware.mex.tl/images/18149/PROGRAMACI%C3%93N%20EXTREMA.pdf.

2010. buenas tareas. buenas tareas. [En línea] Agosto de 2010. [Citado el: 5 de 12 de 2012.] http://www.buenastareas.com/ensayos/Metodos-Revision-De-Evidencia-Auditoria-Informatica/614650.html.

Cabrera., Lic Aries M. Cañellas. [En línea] http://dialnet.unirioja.es/servlet/articulo?codigo=2037601.

calidad, Asociacion española para la. AEC. AEC. [En línea] http://www.aec.es/c/document_library/get_file?uuid=783d8fbd-12df-43f3-b12c-b1c5ca5ce5d7&groupId=10128.

Canós, José H., Letelier, Patricio y Penad, Mª Carmen. Métodologías Ágiles en el Desarrollo de Software. [En línea] [Citado el: 23 de Marzo de 2013.] http://noqualityinside.com.ar/nqi/nqifiles/XP_Agil.pdf.

Carmona, M. 1998. La Auditoría Interna de Gestión: Aspecto Teóricos. El Caso Particular Cubano . La Auditoría Interna de Gestión: Aspecto Teóricos. El Caso Particular Cubano . España, Universidad de Huelva : s.n., 1998.

Carnegie Mellon University. 2013. Software Engineering Institute . Software Engineering Institute . [En línea] 2013. [Citado el: 15 de 02 de 2013.] http://www.sei.cmu.edu/.

Carrera, Cristian . PSD a HTML Paso a Paso. PSD a HTML Paso a Paso. [En línea] [Citado el: 21 de Marzo de 2013.] http://www.psdahtmlpasoapaso.com/blog/5-razones-para-usar-html5.

2013. Centro Nacional de Calidad del Software (Calisoft). Centro Nacional de Calidad del Software (Calisoft). [En línea] 2013. [Citado el: 15 de 03 de 2013.] calisoft.uci.cu.

Cortez, Liliana Uriarte. 2011. Scribd. Scribd. [En línea] 23 de Mayo de 2011. [Citado el: 15 de 11 de 2012.] http://es.scribd.com/doc/56037001/FACTORES-DE-RIESGO-Y-DANO-PARA-LA-SALUD-vrdadero.

Cursos de Informática. Cursos de Informática. [En línea] http://nilossgp.blogspot.com/2012/10/entornos-de-desarrollo.html.

2012. Cursos de Informática. Cursos de Informática. [En línea] Nilson gongora, 14 de octubre de 2012. [Citado el: 15 de 02 de 2013.] http://nilossgp.blogspot.com/2012/10/entornos-de-desarrollo.html.

Daniel Pecos. PostGreSQL vs. MySQL. PostGreSQL vs. MySQL. [En línea] [Citado el: 03 de 01 de 2012.] http://www.danielpecos.com/docs/mysql_postgres/index.html.

Dayvis Malfará, Diego Cukerman, Fernando Cócaro, Juan Pablo Cassinelli, Renzo Séttimo. 2006. Gestión de Software. Texting en XP. 2006.

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

Escribano, Gerardo Fernández. Introducción a eXtreme Programming.

Española, Real Academia. 2012. RAE Real Academia Española. RAE Real Academia Española. [En línea] RAE Real Academia Española, Junio de 2012. [Citado el: 10 de 01 de 2013.] http://www.rae.es.

Figueroa, Arturo Reyes. 2007. Control Interno sobre el Reporte Financiero - Guía para empresas pequeñas públicas (COSO III). 2007.

2011. Gaceta oficial de la República de Cuba. Cuba : s.n., 2011. ISSN 1682-7511.

2007. Gestión de Riesgos. 2007.

González, María Cristina Pérez. 2011. [En línea] 01 de 01 de 2011. www.pedagogiamagna.com.

Guerra, Roilán González. 2012. Sistema para la gestión de la información de postgrado en la facultad 3. . La Habana : s.n., 2012.

Gutierrez, Jorge A. Saavedra. El mundo informático. el mundo informático. [en línea] [citado el: 15 de 01 de 2013.] http://jorgesaavedra.wordpress.com/2007/05/05/lenguajes-de-programacion/.

Hernández, Dr. Mario Jorge Malagón y Cabrera., Ing. Yicel Frias. 2010. [En línea] 2010. www.univ-paris-diderot.fr/comm/infodoc/cdrom1/Comision%208/16%20Jorge%20Malagon%20Hdez.%202.pdf.

Humberto Martín Morales, Marlon Abreu Lugo. 2008. Herramienta para la gestión de contenido y configuración de ejercicios y clases reutilizables. Ciudad de La Habana : s.n., 2008.

Infante, Liuba María. 2009. Disminución de la exposición a los riesgos en el proyecto Sistema de Información Geográfica de la Universidad de las Ciencias Informáticas a través de la aplicación de un modelo formal de Gestión de Riesgos. La Habana : s.n., 2009.

Informática, Instituto Nacional de Estadística E. Herramientas Case. s.l. : Sub-Jefatura de Informática. 875-99-OI-OTDETI-INEI.

Informáticas, Universidad de las Ciencias. IPP-3500:2008 Libro de Proceso para Definir Procesos.

Informático, Revista de Derecho. 2006. s.l. : Alfa-Redi, 2006. ISSN 1681-5726.

Ing Yeleny Zulueta Veliz, Dra Anaisa Hernández Gonzalez. 2007. Modelo de Gestión de Riesgos en Proyectos de Desarrollo de Software. Modelo de Gestión de Riesgos en Proyectos de Desarrollo de Software. La Habana : s.n., 2007.

2012. Ingeniero de Gestión. Ingeniero de Gestión. [En línea] 20 de Agosto de 2012. [Citado el: 10 de Febrero de 2013.] http://ingenierogestion.blogspot.com/2009/06/pruebas-de-caja-negra-y-caja-blanca.html.

Introducción a Symfony. Introducción a Symfony. [En línea] [Citado el: 3 de 12 de 2012.] http://www.librosweb.es/symfony_1_2/capitulo_1/symfony_en_pocas_palabras.html.

J. J. Gutiérrez, M. J. Escalona, M. Mejías, J. Torres. Pruebas del sistema en programación extrema. España : s.n.

2012. Jummp. Jummp. [En línea] 10 de 01 de 2012. http://jummp.wordpress.com/2012/01/10/desarrollo-de-software-tarjetas-crc/.

Jurán, Joseph. 1987. Manual del Control de la Calidad. 1987.

1951. Manual del Control de la Calidad. 1951.

Justicia, Gaceta Oficial de la República de Cuba. Ministerio de. 2009. CUBA : s.n., 2009. V||-30. ISSN 1682-7511.

Lay, José Manuel Taveras. 2011. Relación entre administración de riesgo y auditoría interna. República Dominicana : s.n., 2011.

LibrosWeb. LibrosWeb. [En línea] [Citado el: 3 de 12 de 2012.] http://www.librosweb.es/symfony_1_2/capitulo_1/symfony_en_pocas_palabras.html.

2013. Librosweb. Librosweb. [en línea] 2013. [citado el: 3 de febrero de 2013.] Http://www.librosweb.es/symfony_1_2/capitulo_1/symfony_en_pocas_palabras.html.

León, R.A.H. El paradigma cuantitativo de la investigación científica. 2002. P.959-16-0343-6.

Limonta Gutiérrez, Dayron. 2012. Extensión del IDE Netbeans para el desarrollo de aplicaciones empleando el marco de trabajo Sauxe. La Habana : Universidad de las Ciencias Informáticas , 2012.

Lomeli, Walter Ivan Reyes. [En línea] [Citado el: 10 de 12 de 2012.] http://es.scribd.com/doc/52208534/29/CARACTERISTICAS-Y-VENTAJAS-DEL-APACHE.

López, Ing. Nohra. Sistema de Gestión de la Calidad. [En línea] [Citado el: 10 de 01 de 2013.] http://www.slideshare.net/nohramilo/la-norma-iso.

M., Itzcoalt Álvarez. Desarrollo Ágil con Scrum. Desarrollo Ágil con Scrum. [En línea] http://cic.puj.edu.co/wiki/lib/exe/fetch.php?media=materias:sg07.p02.scrum.pdf.

2012. MAGERIT – versión 3.0. Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información. s.l. : Subdirección General de Información, Documentación y Publicaciones, Centro de Publicaciones, 2012. Vol. Libro I. NIPO: 630-12-171-8.

Maniasi, Sebastián. Un Modelo para la Identificación de Riesgos en Basea Taxonomías. Argentina: Global Software Group (GSG) Argentina . ISSN 1667-5002.

Mario Piattini, Emilio del Peso. 2008. Auditoria de tecnologías y sistemas de información. S.l.: Alfaomega grupo edit., 2008. ISBN 9789701513781.

Marroquin, Lic. César Ramírez. 2012. [En línea] 12 de 07 de 2012. http:\\interculturalidad.usac.edu.gt/elggdata/D/o/n/i/s/Donis/file/1315072921tic%27s.docx.

2013. Metodología de SCRUM. Metodología de SCRUM. [En línea] 18 de abril de 2013. [Citado el: 20 de 12 de 2012.]

http://wiki.monagas.udo.edu.ve/index.php/Metodolog%C3%ADas_SCRUM_y_XP#CARACTER.C3.8DSTICAS_DE_LA_METODOLOG.C3.8DA_SCRUM.

Miranda, Prof. Dr. Ing. Arturo Luis Romero y Lic. Sandor Luis. 2007 . gestiopolis. gestiopolis. [En línea] 10 de 08 de 2007 . [Citado el: 10 de 01 de 2013.] http://www.gestiopolis.com/administracion-estrategia/la-calidad-historia-conceptos-y-terminos-asociados.htm.

Molina., Ing. Dialexis Acosta. 2008-2009. Diseño e implementación del proceso de auditoría de la calidad para la actividad productiva en la universidad de las ciencias informáticas (UCI). Ciudad de la Habana : s.n., 2008-2009.

Montesino, Miguel. 2011. PMI. PMI. [En línea] Noviembre de 2011. [Citado el: 22 de febrero de 2013.] http://www.pmi.org/.

2011. Norma Cubana ISO 19011: 2012. Directríces para la auditoría de los sistemas de gestión. La Habana : iso, 2011.

2008. Norma Internacional ISO 9001. Norma Internacional ISO 9001. [En línea] 14 de 11 de 2008. [Citado el: 01 de 03 de 2013.] http://www.iescinoc.edu.co/NORMOGRAMA/ISO%209001-2008.pdf.

2007. Normas ISO 9000 y Calidad. Normas ISO 9000 y Calidad. [En línea] 2007. [Citado el: 14 de 12 de 2012.] http://normas-iso-9000.blogspot.com/2007/11/terminos-relativos-la-gestion.html.

OpenLogic. OpenLogic. [En línea] [Citado el: 15 de 12 de 2012.] http://www.openlogic.com/wazi/bid/188125/PostgreSQL-vs-MySQL-Which-Is-the-Best-Open-Source-Database.

Osiris Pérez Moya, Rislaidy Pérez Ramos. 2010. Gestión de proyectos analizando riesgos y situaciones de incertidumbre. 2010.

Pérez, Mauel. 2009. Qué es Symfony? [En línea] 16 de mayo de 2009. [Citado el: 5 de 12 de 2012.] http://www.tecnoretales.com/linux/que-es-symfony/.

2001. PHP. [En línea] 2001. [Citado el: 26 de Abril de 2013.] http://php.net.

PMI - Project Management Institute . 2004. Guía de los Fundamentos de la Dirección de Proyectos (Guía del PMBOK®). Pennsylvania, Estados Unidos Americanos : s.n., 2004. ANSI/PMI 99-001-2004.

PostGreSQL vs. MySQL. PostGreSQL vs. MySQL. [En línea] [Citado el: 03 de 01 de 2012.] http://www.danielpecos.com/docs/mysql_postgres/index.html.

Potencier, Fabien. 2011. What is Symfony2?. [En línea] 25 de October de 2011. [Citado el: 9 de 04 de 2013.] fabien.potencier.org/article/49/what-is-symfony2.

Programación Extrema. Programación Extrema.

Pullas Cabezas, Elizabeth Tatiana. 2010. Desarrollo de un sistema para voto electrónico y emisión de resultados en procesos electorales de la escuela politécnica nacional. Quito : s.n., febrero de 2010.

Quijano, Juan. 2012. GENBETA:dev desarrollo y software. GENBETA:dev desarrollo y software. [En línea] 28 de febrero de 2012. [Citado el: 10 de 04 de 2013.] https://sites.google.com/a/uji.es/gesproin/.

Radel Calzada Pando, José Manuel de León Cano. 2010. Revsita cubana de ciencias informáticas. [En línea] 2010. [Citado el: 25 de 12 de 2012.] http://rcci.uci.cu/index.php/rcci/article/view/90. issn-e: 2227.1899.

Radel Calzada Pando, José Manuel de León Cano. Características de la gestión de riesgos en las empresas cubanas. s.l. : Revista Cubana de Ciencias Informáticas. ISSN: 1994-1536 | RNPS: 0547.

Ricardo Mercado del Collado, Mónica López Granados,Gustavo Balderas Rosas. El aseguramiento de la calidad en el instituto. Mexico : s.n. I.S.S.N.: 1138-2783.

Rodolfo Bertone, Pablo Thomas, Daniel Taquias, Sebastián Pardo. 2010. Herramienta para la Gestión de Riesgos en Proyectos de Software. Argentina : s.n., 2010.

Rodríguez Corbea, Maite y Ordóñez Pérez, Meylin. 2007. La metodología XP aplicable al desarrollo del software educativo en Cuba. Ciudad de la Habana : s.n., 2007.

Rodríguez, Lydia Rosa Ríos, y otros. E+I: Tecnología de la Educación. E+I: Tecnología de la Educación. [En línea] [Citado el: 04 de 02 de 2013.] Dialnet.unirioja.es/servlet/dcart?Info=link&codigo=2304267&orden=117395. ISSN: 1681-5653.

Roger.Pressman. 2008. Ingeniería.del.Software. Un enfoque práctico. 2008.

Romaní, Juan Cristóbal Cobo. 2009. Zer - Revista de Estudios de Comunicación. Zer - Revista de Estudios de Comunicación. [En línea] 22 de 09 de 2009. [Citado el: 10 de 01 de 2013.] Http://www.ehu.es/ojs/index.php/Zer/article/view/2636/2184. ISSN: 1137-1102.

Rosario, Jimmy. 2005. [En línea] 2005. [Citado el: 06 de 01 de 2013.] http://www.cibersociedad.net/archivo/articulo.php?art=218.

2007. Scribd. Scribd. [En línea] 03 de 2007. [Citado el: 10 de 12 de 2012.] http://es.scribd.com/doc/52208534/29/CARACTERISTICAS-Y-VENTAJAS-DEL-APACHE. 02490334.

Secretaría Central de ISO en Ginebra, Suiza. 2008. Norma internacional ISO 9001. Sistemas de gestión de la calidad — requisitos. 2008. Iso 9001:2008 .

2005. Sistemas Informáticos. Sistemas Informáticos. [En línea] 2005. [Citado el: 11 de Marzo de 2013.] http://www.sitsoft.com.ar/Audita.asp.

Solís., Manuel Calero. 2003. Una explicación de la programación extrema (XP). Madrid : s.n., 2003.

Universidad de las ciencias informaticas. [En línea] [Citado el: 1 de 12 de 2012.] http://www.uci.cu/.

Yandielys Reyes Plano, Ing. Yurisbel Vega Ortiz. 2009. Aplicación y mejora del Modelo de Gestión de Riesgos “MoGeRi” al proyecto “Captura y Catalogación de Medias”. La Habana : s.n., 2009.

Yeleny Zulueta, Eder Despaigne, Anaisa Hernández. 2009. La gestión de riesgos en la producción de software y la formación de profesionales de la informática: experiencias de una universidad cubana. España : Revista Española de Innovación, Calidad, 2009. ISSN (Versión electrónica): 1885-4486.

5 Anexos

Anexo # 1: Descripción textual del Proceso de Gestión de Riesgos del proceso Auditoría de la Calidad del software.

Etapa 1: Identificación de riesgos

1. El auditor líder junto al equipo auditor se encargan de identificar las posibles amenazas.
   1.1. A partir del Plan de Auditoría los auditores identifican los posibles riesgos creando un listado.
   1.2. En caso de que se quedaron riesgos sin identificar el auditor líder se encarga de adicionarlos al listado.
   1.3. El auditor líder confecciona el listado de riesgos identificados.

Etapa 2: Análisis y evaluación de riesgos

2. El auditor líder analiza y evalúa cada uno de los riesgos identificados obteniendo una lista de riesgos priorizados, este listado va a contener los riesgos de mayor relevancia.
   2.1. Partiendo del listado de riesgos identificados el líder de la auditoría realiza el análisis cualitativo y cuantitativo de los riesgos.
   2.2. Una vez analizados los riesgos, estos son evaluados por el auditor líder para la toma de decisiones.
   2.3. El auditor líder documenta los riesgos analizados actualizando el listado de riesgos.
   2.4. El equipo auditor analiza si es viable la realización de la auditoría a partir de la evaluación de los riesgos.
      - ✓ En caso de no ser viable la realización de la auditoría, esta es abortada informándoselo al auditado y exponiéndose las causas en el informe final de auditoría.
      - ✓ En caso contrario se procede a la prevención de los riesgos.

Etapa 3: Prevención de riesgos

3. El auditor líder tiene la responsabilidad de elaborar un listado con las acciones a tomar para cada uno de los riesgos. De esta forma el listado de riesgos se actualiza conteniendo los aspectos fundamentales asociados a cada uno de los ellos.
   3.1. El equipo auditor realiza el estudio de las posibles acciones y respuestas a los riesgos
   3.2. El auditor líder verifica cada una de las acciones y las estrategias a seguir.
   3.3. Se actualiza el listado de riesgos con las acciones a tomar para cada uno ellos.

Etapa 4: Seguimiento y control de riesgos

4. Durante el proceso de auditoría se le da seguimiento a cada uno de los riesgos, esto posibilita mantener un control sobre los mismos.
   4.1. El auditor líder se encarga de chequear el estado de los riesgos y actualiza los riesgos clasificándolos en:
      - ✓ **Eliminado**: Si el riesgo no se materializa durante el desarrollo de la auditoría.
      - ✓ **Avance**: Si del total de riesgos se materializan menos del cincuenta por ciento durante el desarrollo de la auditoría.
      - ✓ **Estancado**: Si del total de riesgos se materializan más del cincuenta por ciento durante el desarrollo de la auditoría.
      - ✓ **Retroceso**: Si se materializan más riesgos de los previstos convirtiéndose en dificultades inesperadas.
   4.2. El chequeo de las actividades definidas para enfrentar cada uno de los riesgos es responsabilidad del auditor líder.
   4.3. Se actualiza los valores de los riesgos en el listado de riesgos, tarea ejecutada por el líder de la auditoría.
   4.4. En esta fase o etapa se monitorean los riesgos y las respuestas asociadas a cada uno para evaluar el comportamiento y hacer cambios de estrategias en caso de ser necesarias.
   4.5. El auditor líder informa de la situación de los riesgos así como los aspectos relacionados con los mismos.
   4.6. En esta fase se verifica si han surgido nuevos riesgos.
      - ✓ Si han surgido nuevos riesgos se identifican (Etapa 1).
      - ✓ En caso contrario se continua verificando y controlando cada uno de los riesgos durante todo el proceso de auditoría.

Anexo # 2: Listado de riesgos comunes en las auditorías de la calidad del software.

- ✓ Comprobar la falta de formación, de los auditores y de los auditados, en el qué y el cómo se va a auditar y para qué sirven las auditorías.
- ✓ Averiguar si las condiciones excepcionales han sido tomadas como normales.
- ✓ Averiguar si el auditor tiene prejuicios, por los que emite observaciones subjetivas.
- ✓ Fallos por parte del equipo auditor.

- ✓ Impuntualidad por parte de ambos equipos a las actividades de auditoría.
- ✓ No enviar el local de la reunión de apertura en el tiempo estipulado por parte del equipo de proyecto.
- ✓ Ausencias del personal responsable por el proyecto de recibir a los auditores.
- ✓ Problemas con los locales de realización de la auditoría.
- ✓ Demora en socializar el expediente de proyecto al equipo auditor.
- ✓ No enviar en el tiempo estipulado las acciones correctivas.
- ✓ No alcanza el tiempo de la auditoría planificada.
- ✓ No esté disponible el líder auditor en la fase de seguimiento.

Anexo # 3: Tablas de las tareas de implementación

Tabla 46: TI_02 Modificar riesgo

Tarea de implementación	
Número Tarea: TI_03	**Historia de Usuario:** HU_05
Nombre Tarea: Modificar riesgo	
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 0.1
Fecha Inicio: 26/3/2013	**Fecha Fin:** 27/3/2013
Descripción: el usuario podrá modificar todos los datos de los riesgos.	

Tabla 47: TI_03 Eliminar riesgo

Tarea de implementación	
Número Tarea: TI_03	**Historia de Usuario:** HU_05
Nombre Tarea: Eliminar riesgo	
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 0.1
Fecha Inicio: 27/3/2013	**Fecha Fin:**28/3/2013
Descripción: el usuario con rol (ROLE_ADMINISTRADOR) es el encargado de eliminar todos los datos de un riesgo.	

Tabla 48: TI_04 Listar riesgo

Tarea de implementación	
Número Tarea: TI_04	**Historia de Usuario:** HU_05
Nombre Tarea: Listar riesgo	
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 0.1
Fecha Inicio: 28/3/2013	**Fecha Fin:** 29/3/2013
Descripción: el usuario puede mostrar el listado de todos los riesgos existentes para una auditoría.	

Tabla 49: TI_01 Identificar riesgo

Tarea de implementación	
Número Tarea: TI_01	**Historia de Usuario:** HU_06
Nombre Tarea: Identificar riesgos	
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 0.1
Fecha Inicio: 1/4/2013	**Fecha Fin:** 3/4/2013
Descripción: el usuario podrá identificar los distintos riesgos que cree que pueden ocurrir en una auditoría.	

Tabla 50: TI_02 Priorizar riesgo

Tarea de implementación	
Número Tarea: TI_02	**Historia de Usuario:** HU_06
Nombre Tarea: Priorizar riesgos	
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 0.2
Fecha Inicio: 4/4/2013	**Fecha Fin:** 10/4/2013

Descripción: esta tarea tiene como objetivo ordenar los riesgos atendiendo al a probabilidad y al impacto de cada riesgo.

Tabla 51: TI_03 Listar riesgos analizados

Tarea de implementación	
Número Tarea:TI_03	**Historia de Usuario:** HU_06
Nombre Tarea: Listar riesgos analizados	
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 0.2
Fecha Inicio: 10/4/2013	**Fecha Fin:** 12/4/2013
Descripción: el usuario podrá ver el listado de riesgos priorizados.	

Tabla 52: TI_01 Prevención de riesgos

Tarea de implementación	
Número Tarea: TI_01	**Historia de Usuario:** HU_07
Nombre Tarea: Prevención de riesgos	
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 0.3
Fecha Inicio: 15/4/2013	**Fecha Fin:** 26/4/2013
Descripción: el usuario debe prevenir los distintos riesgos analizados, asignándole un responsable, una respuesta y un tiempo para darle solución al riesgo.	

Tabla 53: TI_01 Generar reporte

Tarea de implementación	
Número Tarea: TI_17	**Historia de Usuario:** HU_08
Nombre Tarea: Generar reporte	

Tipo de Tarea: Desarrollo	**Puntos Estimados:** 0.2
Fecha Inicio: 29/4/2013	**Fecha Fin:** 3/5/2013
Descripción: el usuario podrá generar un reporte en formato word y así ver el listado de riesgos que identificó en una auditoría.	

Tabla 54: TI_01 Ver perfil

Tarea de implementación	
Número Tarea: TI_01	**Historia de Usuario:** HU_09
Nombre Tarea: Ver perfil	
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 0.2
Fecha Inicio: 6/5/2013	**Fecha Fin:** 10/5/2013
Descripción: una vez que el usuario esté autenticado podrá acceder a su perfil y modificar sus datos.	

Tabla 55: TI_01 Listar los riesgos de una auditoría

Tarea de implementación	
Número Tarea: TI_01	**Historia de Usuario:** HU_10
Nombre Tarea: Listar los riesgos de una auditoría	
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 0.1
Fecha Inicio: 13/5/2013	**Fecha Fin:** 17/5/2013
Descripción: una vez que el usuario esté autenticado podrá ver el listado de riesgos de una auditoría.	

Anexo # 4: Tabas de los casos de pruebas de aceptación.

Tabla 56: PA_07 Modificar auditoría

Prueba de Aceptación	
Código: PA_07	**Historia de Usuario:** HU_03
Nombre: Modificar auditoría	
Descripción: el usuario con rol Administrador será el encargado de modificar todos los datos de una auditoría y así seleccionar quién será el usuario encargado de realizar la auditoría. Se mostrará un mensaje "Se ha creado correctamente la auditoría".	
Condiciones de Ejecución: el usuario con rol administrador debe estar autenticado.	
Entrada / Pasos de ejecución: el administrador del sistema selecciona la opción *Auditoría,* que aparece en Menú lateral. Se mostrará un listado con todas las auditorías adicionadas en la base de datos. Para modificar una auditoría selecciona el botón *Editar,* mostrándose un formulario donde podrá llenar los campos requeridos para esta operación. Una vez completado todos los campos, el usuario selecciona la opción *Actualizar***,** si los campos son válidos le mostrará un mensaje: ¨La auditoría se ha actualizado correctamente¨.	
Resultado Esperado: el sistema permite que la auditoría se modifique correctamente.	
Evaluación de la Prueba: Satisfactoria	

Tabla 57: PA_07 Eliminar auditoría

Prueba de Aceptación	
Código: PA_07	**Historia de Usuario:** HU_03
Nombre: Eliminar auditoría	
Descripción: el usuario con rol Administrador será el encargado de eliminar la auditoría de la base de datos con toda su información.	
Condiciones de Ejecución: el usuario con rol administrador debe estar autenticado.	

Entrada / Pasos de ejecución: el administrador del sistema selecciona la opción *Auditoría***,** que aparece en Menú lateral. Se mostrará un listado con todas las auditorías en la base de datos. Para eliminar una auditoría selecciona el botón *Eliminar***.**
Resultado Esperado: el sistema permite eliminar una auditoría correctamente de la base de datos con toda su información.
Evaluación de la Prueba: Satisfactoria

Tabla 58: PA_08 Mostrar listado de auditorías

Prueba de Aceptación	
Código: PA_08	**Historia de Usuario:** HU_03
Nombre: Mostrar listado de auditorías	
Descripción: el usuario con rol Administrador será el encargado de ver el listado de auditorías planificadas con todos sus datos.	
Condiciones de Ejecución: el usuario con rol administrador debe estar autenticado.	
Entrada / Pasos de ejecución: el administrador del sistema selecciona la opción *Auditoría,* que aparece en Menú lateral. Se mostrará un listado con todas las auditorías en la base de datos.	
Resultado Esperado: el sistema permite mostrar el listado de auditorías planificadas con toda su información.	
Evaluación de la Prueba: Satisfactoria	

Tabla 59: PA_09 Adicionar riesgo

Prueba de Aceptación	
Código: PA_09	**Historia de Usuario:** HU_04
Nombre: Adicionar riesgo	

Descripción: el usuario una vez autenticado podrá crear distintos riesgos con todos sus datos. Se mostrará un mensaje “Se ha creado correctamente”
Condiciones de Ejecución: el usuario con rol administrador debe estar autenticado.
Entrada / Pasos de ejecución: el usuario selecciona la opción *Riesgo,* que aparece en Menú lateral. Se mostrará un listado con todos los riesgos en la base de datos. Para adicionar un nuevo riesgo selecciona el botón *Adicionar*, mostrándose un formulario donde podrá llenar los campos requeridos para esta operación. Una vez completado todos los campos el usuario selecciona la opción *Guardar***,** si los campos son válidos le mostrará un mensaje: ¨El riesgo se ha adicionado correctamente¨.
Resultado Esperado: el sistema permita adicionar nuevos riesgos.
Evaluación de la Prueba: Satisfactoria

Tabla 60: PA_10 Modificar riesgo

Prueba de Aceptación	
Código: PA_10	**Historia de Usuario:** HU_04
Nombre: Modificar riesgo	
Descripción: el usuario una vez autenticado en la aplicación podrá modificar el riesgo con todos sus datos.	
Condiciones de Ejecución: el usuario debe estar autenticado.	
Entrada / Pasos de ejecución: el usuario selecciona la opción *Riesgo,* que aparece en Menú lateral. Se mostrará un listado con todos los riesgos en la base de datos. Para modificar un riesgo selecciona el botón *Editar***,** mostrándose un formulario donde podrá llenar los campos requeridos para esta operación. Una vez completado todos los campos el usuario selecciona la opción *Actualizar***,** si los campos son válidos le mostrará un mensaje: ¨El riesgo se ha actualizado correctamente¨.	
Resultado Esperado: el sistema permita modificar los riesgos correctamente.	

Evaluación de la Prueba: Satisfactoria

Tabla 61: PA_11 Eliminar riesgo

Prueba de Aceptación	
Código: PA_11	**Historia de Usuario:** HU_04
Nombre: Eliminar riesgo	
Descripción: el usuario podrá eliminar los riesgos correctamente si así lo desea, y se borran todos sus datos de la base de datos.	
Condiciones de Ejecución: el usuario debe estar autenticado.	
Entrada / Pasos de ejecución: el usuario selecciona la opción *Riesgo,* que aparece en Menú lateral. Se mostrará un listado con todos los riesgos en la base de datos. Para eliminar un riesgo selecciona el botón *Eliminar***.**	
Resultado Esperado: el sistema permita eliminar los riesgos correctamente con toda su información de la base de datos.	
Evaluación de la Prueba: Satisfactoria	

Tabla 62: PA_12 Mostrar listado de riesgo

Prueba de Aceptación	
Código: PA_12	**Historia de Usuario:** HU_04
Nombre: Mostrar listado de riesgos	
Descripción: el usuario una vez autenticado podrá acceder a la lista de riesgos y verlos si así lo desea en el menú principal o revisando una auditoría.	
Condiciones de Ejecución: el usuario debe estar autenticado.	
Entrada / Pasos de ejecución: el usuario selecciona la opción *Riesgo*, que aparece en Menú lateral. Se mostrará un listado con todos los riesgos en la base de datos.	

Resultado Esperado: el sistema permita listar los riesgos sus datos correctamente.
Evaluación de la Prueba: Satisfactoria

Tabla 63: PA_13 Identificar riesgos

Prueba de Aceptación	
Código: PA_13	**Historia de Usuario:** HU_05
Nombre: Identificar riesgos	
Descripción: el usuario podrá identificar todos los riesgos existente en una auditoría, marcándolos en un checkbox.	
Condiciones de Ejecución: el usuario debe estar autenticado.	
Entrada / Pasos de ejecución: el usuario selecciona la opción *Mis Auditorías*, que aparece en el Menú Superior. Se mostrará un listado con todas las auditorías asignadas. Para identificar los riesgos accede al listado haciendo “clic” sobre el botón *Revisar*, luego aparecerá un listado de posibles riesgos para esa auditoría. Para identificar los riesgos, selecciona la opción *Continuar* donde mediante los checkbox podrá identificar los riesgos para esa auditoría. Para ver los riesgos identificados selecciona la opción *Continuar* donde aparecerá el listado con los riesgos identificados.	
Resultado Esperado: el sistema permita mostrar un listado de riesgos identificados en esa auditoría.	
Evaluación de la Prueba: Satisfactoria	

Tabla 64: PA_14 Priorizar riesgos

Prueba de Aceptación	
Código: PA_14	**Historia de Usuario:** HU_05
Nombre: Priorizar riesgos	
Descripción: el usuario una vez autenticado y después de identificar los riesgos en una auditoría podrá priorizar el listado de riesgos.	

Condiciones de Ejecución: el usuario debe estar autenticado.
Entrada / Pasos de ejecución: el usuario selecciona la opción *Mis Auditorías*, que aparece en el Menú Superior. Se mostrará un listado con todas las auditorías asignadas. Para identificar los riesgos accede al listado haciendo “clic” sobre el botón *Revisar*, luego aparecerá un listado de posibles riesgos para esa auditoría. Para identificar los riesgos selecciona la opción *Continuar* donde mediante los checkbox podrá identificar los riesgos para esa auditoría. Para ver los riesgos identificados selecciona la opción *Continuar* donde aparecerá el listado con los riesgos identificados. Al hacer “clic” en *Analizar,* se mostrará un mensaje y un listado con los riesgos ordenados según su probabilidad e impacto.
Resultado Esperado: el sistema permita que se prioricen los riesgos existente en una auditoría.
Evaluación de la Prueba: Satisfactoria

Tabla 65: PA_16 Prevención de riesgos

Prueba de Aceptación	
Código: PA_16	**Historia de Usuario:** HU_06
Nombre: Prevención de riesgos	
Descripción: el usuario podrá acceder a mis auditorías y si está activa la auditoría podrá prevenir los riesgos identificados.	
Condiciones de Ejecución: el usuario debe estar autenticado.	
Entrada / Pasos de ejecución: el usuario selecciona la opción *Mis Auditorías* que aparece en el Menú Superior. Se mostrará un listado con todas las auditorías asignadas. Para identificar los riesgos accede al listado haciendo “clic” sobre el botón *Revisar*, luego aparecerá un listado de posibles riesgos para esa auditoría. Para identificar los riesgos selecciona la opción *Continuar* donde mediante los checkbox podrá identificar los riesgos para esa auditoría. Para ver los riesgos identificados selecciona la opción *Continuar* donde aparecerá el listado con los riesgos identificados. Al hacer “clic” en *Analizar,* se mostrará un mensaje y un listado con los riesgos ordenados según su probabilidad e impacto.	

Si el mensaje que se muestra es “La auditoría se puede ejecutar satisfactoriamente” o “La auditoría es aplazada” se selecciona el botón *Continuar* donde aparecerá un formulario para prevenir los riesgos. Si el mensaje que se muestra es “La auditoría es abortada” podrá generar un reporte en formato .doc.
Resultado Esperado: el sistema le permita al usuario la prevención a los riesgos.
Evaluación de la Prueba: Satisfactoria

Tabla 66: PA_17 Generar reporte

Prueba de Aceptación	
Código: PA_17	**Historia de Usuario:** HU_07
Nombre: Generar reporte	
Descripción: el usuario una vez autenticado podrá generar reportes después de abortada una auditoría o ejecutada o aplazada.	
Condiciones de Ejecución: el usuario debe estar autenticado.	
Entrada / Pasos de ejecución: el usuario selecciona la opción *Mis Auditorías,* que aparece en el Menú Superior. Se mostrará un listado con todas las auditorías asignadas. Para identificar los riesgos accede al listado haciendo “clic” sobre el botón *Revisar*, luego aparecerá un listado de posibles riesgos para esa auditoría. Para identificar los riesgos selecciona la opción *Continuar* donde mediante los checkbox podrá identificar los riesgos para esa auditoría. Para ver los riesgos identificados selecciona la opción *Continuar* donde aparecerá el listado con los riesgos identificados. Al hacer “clic” en *Analizar,* se mostrará un mensaje y un listado con los riesgos ordenados según su probabilidad e impacto. Si el mensaje que se muestra es “La auditoría se puede ejecutar satisfactoriamente” o “La auditoría es aplazada” se selecciona el botón *Continuar* donde aparecerá un formulario para prevenir los riesgos. Al hacer “clic” en el botón *Continuar* aparecerá un botón para generar el reporte en formato .doc.	
Resultado Esperado: el sistema le permita al usuario que se genere el reporte correctamente.	
Evaluación de la Prueba: Satisfactoria	

Tabla 67: PA_18 Mis auditorías

Prueba de Aceptación	
Código: PA_18	**Historia de Usuario:** HU_08
Nombre: Mis auditorías	
Descripción: El usuario podrá acceder a Mis auditorías y ver las auditorías planificadas.	
Condiciones de Ejecución: El usuario debe estar autenticado.	
Entrada / Pasos de ejecución: El usuario selecciona la opción *Mis Auditorías* que aparece en el Menú Superior. Se mostrará un listado con todas las auditorías asignadas.	
Resultado Esperado: El sistema le permita al usuario acceder y ver sus auditorías planificadas.	
Evaluación de la Prueba: Satisfactoria	

Tabla 68: PA_20 Mostrar listado de riesgos identificados por auditoría

Prueba de Aceptación	
Código: PA_20	**Historia de Usuario:** HU_10
Nombre: Mostrar listado de riesgos identificados por auditoría	
Descripción: el usuario podrá listar los riesgos identificados en una auditoría.	
Condiciones de Ejecución: el usuario debe estar autenticado.	
Entrada / Pasos de ejecución: seleccionar el link que se encuentra en el menú lateral y seleccionar la opción *Riesgos de una Auditoría***.** Se le mostrará una vista donde podrá seleccionar la auditoría deseada para mostrar el listado de riesgos correspondiente a esa auditoría. Seleccionando la opción *Visualizar***.**	
Resultado Esperado: el sistema le permita al usuario visualizar el listado de riesgos para la auditoría seleccionada.	
Evaluación de la Prueba: Satisfactoria	

Anexo # 5 Guía para informar el peso de los criterios

Guía para informar el peso de los criterios.

Fecha de recepción _____

Fecha de entrega _____

Nombre y Apellidos del evaluador _____

Le otorgará un peso a cada criterio de acuerdo a su opinión y el peso total de cada grupo debe sumar:

Grupo No.1..... 25
Grupo No.2..... 20
Grupo no.3..... 20
Grupo No.4.....35

Para que el peso total asignado sea 100.

Grupo No. 1: Criterios de mérito científico

1. Valor científico de la propuesta.
   Peso.....
2. Calidad de la investigación.
   Peso.....
3. Aporte científico.
   Peso.....
4. Novedad científica.
   Peso.....

Grupo No. 2: Criterios implantación

5. Satisfacción de las necesidades de los auditores del Centro Nacional de Calidad del Software.

   Peso.....
6. Necesidad del empleo de la propuesta.
   Peso.....

Grupo No.3: Criterios de flexibilidad

7. Adaptabilidad en las auditorías de la calidad del software.
   Peso.....
8. Adaptabilidad a cualquier servicio referente a software de gestión de riesgos.
   Peso.....

Grupo No.4: Criterios de impacto

9. Repercusión en las auditorías de la calidad del software.
   Peso.....
10. Aceptación de la propuesta por los auditores.
    Peso.....
11. Posibilidades de aplicación.
    Peso.....
12. Impacto en el centro para el cual está destinado.
    Peso.....

Anexo # 6 Guía para la evaluación

Guía para informar el peso de los criterios.

Fecha de recepción _____

Fecha de entrega _____

Nombre y Apellidos del evaluador _____

Criterios de medida que se evalúan en una escala de 1 - 5

Grupo No. 1: Criterios de mérito científico

1. Valor científico de la propuesta.
   Peso.....
2. Calidad de la investigación.
   Peso.....
3. Aporte científico.
   Peso.....
4. Novedad científica.
   Peso.....

Grupo No. 2: Criterios implantación

5. Satisfacción de las necesidades de los auditores del Centro Nacional de Calidad del Software.

   Peso.....
6. Necesidad del empleo de la propuesta.
   Peso.....

Grupo No.3: Criterios de flexibilidad

7. Adaptabilidad en las auditorías de la calidad del software.

Peso.....

8. Adaptabilidad a cualquier servicio referente a software de gestión de riesgos.
   Peso.....

Grupo No.4: Criterios de impacto

9. Repercusión en las auditorías de la calidad del software.
   Peso.....
10. Aceptación de la propuesta por los auditores.
    Peso.....
11. Posibilidades de aplicación.
    Peso.....
12. Impacto en el centro para el cual está destinado.
    Peso.....

Anexo # 7 Tabla de los valores del peso relativo a cada criterio

G	C/E	E1	E2	E3	E4	E5	E6	E7	Ep
25	**C1**	7	0	4	2	9	5	6	4,714
	C2	6	5	5	10	6	10	6	6,857
	C3	6	10	4	5	5	5	6	5,857
	C4	6	10	4	8	5	5	7	6,428
20	**C5**	10	10	7	10	10	10	5	8,857
	C6	10	10	13	10	10	10	15	11,142
20	**C7**	11	10	10	12	10	10	5	9,714
	C8	9	10	10	8	10	10	15	10,285
35	**C9**	9	5	10	10	11	10	5	8,571
	C10	10	10	7	10	7	5	5	7,714
	C11	10	15	8	8	9	10	20	11,428
	C12	6	5	10	7	8	10	5	7,285

T		100	100	100	100	100	100	100	100

Anexo # 7 Tabla para el cálculo de concordancia

Expertos/Criterios	**E1**	**E2**	**E3**	**E4**	**E5**	**E6**	**E7**	**ΣE**	**Ep**	**ΔC**	**ΔC^2**
C1	7	0	4	2	9	5	6	33	4,714	17	289
C2	6	5	5	10	6	10	6	48	6,857	2	4
C3	6	10	4	5	5	5	6	41	5,857	9	81
C4	6	10	4	8	5	5	7	45	6,428	5	25
C5	10	10	7	10	10	10	5	62	8,857	12	144
C6	10	10	13	10	10	10	15	78	11,142	28	784
C7	11	10	10	12	10	10	5	68	9,714	18	324
C8	9	10	10	8	10	10	15	72	10,285	22	484
CC9	9	5	10	10	11	10	5	60	8,571	10	100
C10	10	10	7	10	7	5	5	54	7,714	4	16
C11	10	15	8	8	9	10	20	80	11,428	30	900
C12	6	5	10	7	8	10	5	51	7,285	1	1
DC	100	100	100	100	100	100	100	692	98,852	158	3152
MΣE	50										
W	0,1157										
X^2	10,876										

Anexo # 7 Tabla para la clasificación de cada criterio

Criterios	Clasificación					P	P x c
	1	2	3	4	5		
C1			x			0,0909	0,2727
C2				X		0,0833	0,3332
C3				X		0,0975	0,39
C4			x			0,0666	0,1998
C5				X		0,0645	0,258
C6					x	0,0641	0,3205
C7				X		0,0588	0,2352
C8				X		0,0555	0,222
C9				X		0,0666	0,2664
C10				X		0,0740	0,296
C11					X	0,0625	0,3125
C12				X		0,0784	0,3136
Total						**0,8627**	**3,4199**
IA	**0,68398**						