

Universidad de las Ciencias Informáticas
Facultad 5



Título: “Herramienta para la Gestión de Riesgos en el Polo de Hardware y Automática”.

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Autores: Yanet Campo Pérez

Juan Luís Vento Hernández

Tutor: Ing. Irina Elena Argota Vega

Ciudad de la Habana, Mayo 2009

“Año del 50 Aniversario del Triunfo de la Revolución

Datos del Contacto

Ing. Irina Elena Argota

Ingeniera en Ciencias Informáticas, graduada en la Universidad de las Ciencias Informáticas (UCI) en el 2007, Profesora en la UCI en adiestramiento, con 2 años de experiencia en su desempeño laboral.

A nuestros profesores que con su dedicación y esfuerzo nos han transmitido valiosas enseñanzas logrando con ellos despertar un gran interés para nuestra carrera.

A la Ing. Irina Elena Argota Vega, ya que gracias a su ayuda y desinterés contribuyó a la confección de este trabajo.

A nuestros compañeros y amigos que de una forma u otra nos ayudaron y nos dieron aliento constante.

Gracias por ser especiales y estar ahí... Vento.

A mis queridos padres y mi abuela por quererme tanto, por su dedicación y sacrificio han logrado de mí un buen hijo.

A mis tíos por su apoyo y ayuda constante.

A mis hermanos por estar siempre pendiente de mí.

Mil Gracia... Juan Luis

Resumen

La Gestión de Riesgo (GR) es un proceso complejo que conduce al planeamiento y aplicación de políticas, estrategias, instrumentos y medidas orientadas a impedir, reducir, prever y controlar los efectos adversos de fenómenos peligrosos sobre el proyecto; por lo que se llevan acciones integradas de reducción de riesgos a través de actividades de prevención, mitigación, preparación y atención de emergencias. En la facultad 5 de la Universidad de las Ciencias Informáticas (UCI) se encuentra el Polo de Hardware y Automática (HA) donde existen dificultades con la GR, pues no se tiene un mecanismo de control y seguimiento de los riesgos; por tanto, surge la idea de implementar una extensión al Trac, herramienta utilizada en el Polo para la Gestión de Proyecto (GP), que permita la realización de este proceso. El sistema estará desarrollado sobre la plataforma GNU/Linux con su distribución Debian, como lenguaje de programación se utilizará Python, Entorno de Desarrollo Integrado (IDE) Eclipse y la librería Genshi. Para el modelo se empleará la herramienta NetBeans y la Metodología OpenUP para el desarrollo de software.

Palabras Clave

Extensión al Trac, GR, Riesgos

Tabla de Contenido

Introducción.....	8
Capítulo 1: Fundamentación Teórica.....	12
1.1 Introducción.....	12
1.2 Gestión de Riesgos en Proyectos Informáticos.....	12
1.2.1 Planificación de la Gestión de Riesgos	13
1.2.2 Identificación de riesgos.....	14
1.2.3 Análisis cualitativo de los riesgos	16
1.2.4 Análisis cuantitativo de los riesgos	19
1.2.5 Planificación de la respuesta a los riesgos	20
1.2.6 Seguimiento y control de Riesgos	21
1.3 Gestión de Riesgos en la Producción de Software	22
1.3.1 Gestión de Riesgo en el Mundo	22
1.3.2 Gestión de Riesgos en el Polo de Hardware y Automática.....	23
1.4 Sistema Operativo GNU/Linux: Distribución Debian.....	24
1.5 Metodología de Desarrollo de Software: OpenUp.....	25
1.6 Lenguaje de Programación: Python	25
1.7 Librería de Genshi.....	26
1.8 Base de datos SQLite	26
1.9 Herramientas a utilizar.....	26
1.9.1 Herramienta de modelado UML: NetBeans	26
1.9.2 Herramienta de Gestión de Proyecto: Trac	27
1.9.3 Entorno de Desarrollo Integrado de programación: Eclipse.....	27
1.10 Conclusiones Parciales.....	27
Capítulo 2: Característica, Diseño e Implementación del Sistema	28
2.1 Introducción.....	28
2.2 Alcance de la aplicación.....	28
2.3 Estrategia definida.....	28
2.4 Gestión de Riesgos.....	29
2.4.1 Planificación de riesgos	29
2.4.2 Identificación de riesgos	29
2.4.3 Análisis de los riesgos	30
2.4.4 Planeación de respuesta al riesgo	30
2.4.5 Seguimiento y control del riesgo	31
2.5 Especificación de los requisitos de software.....	31
2.5.1 Requisitos Funcionales	32
2.5.2 Requisitos no Funcionales	32
2.6 Diagrama de Casos de Usos.....	35

Tabla de Contenido

2.7 Descripción de los Casos de Uso	36
2.8 Arquitectura.....	40
2.9 Modelo de Diseño.....	41
2.9.1 Diagrama de clases	42
2.9.3 Realizaciones de Caso de Uso	52
2.10 Implementación.....	57
2.11 Conclusiones Parciales.....	59
Capítulo 3: Prueba y Seguimiento de la Solución Planteada	60
3.1 Introducción.....	60
3.2 Plan de Pruebas.....	60
3.3 Diseño de clases de pruebas.....	61
3.4 Manual de Usuario basado en los prototipos funcionales	68
3.5 Seguimiento y Control de la Gestión de Riesgos.....	69
3.6 Resultados presentes y visión futura.....	70
3.7 Aplicación de la Herramienta de Gestión de Riesgos al SCADA.....	70
3.8 Conclusiones.....	71
Conclusiones.....	72
Recomendaciones.....	73
Anexo.....	77
Glosario de Términos.....	78

Introducción

La UCI, surgida al calor de la batalla de ideas, es una universidad productiva cuya misión se enmarca en la producción de software y servicios informáticos a partir de la vinculación estudio-trabajo como modelo de la formación. Para ello integra los procesos de formación, investigación y producción en torno a una temática para convertirla en una rama productiva.

Esta integración garantiza la innovación continua que genera y aporta valor a los productos y servicios, promueve la gestión del conocimiento garantizando un mayor rendimiento, logra una mayor utilización y aprovechamiento de los recursos humanos y materiales, generando alta especialización y colaboración.

Desde sus inicios se comenzaron a formar proyectos de investigación y producción, alcanzando un número considerable de proyectos, por lo que surge la idea de agruparlos en polos productivos, con el objetivo de crear un ambiente más natural para ellos.

Entre los polos productivos de la universidad existe el polo de HA, que posee las siguientes líneas de trabajo:

- Formación de postgrado y pregrado, la cual tiene como objetivo capacitar a sus estudiantes y profesores en el área de Automática y en otros temas productivos y de gestión de software.
- Cooperación nacional e internacional, en esta línea se busca la participación de instituciones nacionales como internacionales.
- Líneas de investigación, con el objetivo de alcanzar los conocimientos necesarios en el desarrollo de las aplicaciones.
- Líneas de desarrollo, con el objetivo de impulsar el trabajo buscando la reutilización de lo implementado.
- Proyectos (Contratos), con el objetivo de poder satisfacer las necesidades de los clientes nacionales e internacionales que soliciten servicios del polo.

El primer proyecto productivo del polo surgió en el 2006, el “SCADA Guardián del Alba”, cuyas siglas significan Supervisory Control and Data Acquisition, en español, Control, Supervisión y Adquisición de Datos, desarrollado bajo los principios de software libre, el desarrollo endógeno y del socialismo. Conformado por un equipo multidisciplinario de estudiantes y profesores de la Facultad 5 y especialistas automáticos de empresas y universidades cubanas y venezolanas con altos conocimientos en el campo de la automática y la informática.

Durante sus inicios se identificaron riesgos y se trazó un plan de mitigación de los mismos. Sin embargo, estos se fueron acumulando y creciendo durante el ciclo de vida del software y por falta de personal que le diera seguimiento se produjo un llamado de alerta ante posibles retrasos de entregas de trabajo según fechas acordadas, encontrándose en riesgo los planes trazados por los líderes del mismo.

Junto al desarrollo de este proyecto, se fueron creando otros proyectos a los que se le descuidó el desarrollo de la GR; incluso no se identificaron posibles herramientas que permitieran automatizar este proceso.

A raíz de las condiciones descritas anteriormente se identificó como **problema científico** de la presente investigación, la siguiente interrogante: ¿Cómo gestionar eficientemente los riesgos en los proyectos productivos del Polo de HA?

Según el problema científico expuesto se plantea como **objeto de estudio** “La GR en el Polo de HA” y como **campo de acción** “Desarrollo del proceso de GR en el Polo de HA, aplicado al proyecto SCADA Guardián del Alba.

El **Objetivo General** que se desea alcanzar es, “Implementar una herramienta para la GR en el Polo de HA”.

Para darle cumplimiento al objetivo trazado se determinó que las **Tareas a realizar en la presente investigación** estarían dirigidas a:

- Estudio sobre las técnicas existentes de GR para la selección y aplicación de dichas técnicas al SCADA Guardián del Alba.
- Investigación sobre la GR en proyectos de producción de software para conocer cuáles son las herramientas y técnicas de dicha área.

- Selección de las técnicas existentes más eficientes para la GR en proyectos productivos de desarrollo de software.
- Realización del proceso de Gestión de Riesgo en el Proyecto SCADA "Guardián del Alba" para demostrar la eficacia de las herramientas y técnicas seleccionadas.
- Realización de una extensión a la herramienta Trac que permita la GR en el Polo de HA para automatizar el proceso en el proyecto.
- Realización de un estudio del lenguaje Python para la implementación de la extensión al Trac como gestor de proyecto.
- Validación de la herramienta implementada con el objetivo de comprobar su funcionamiento.

Para el desarrollo de las tareas científicas se combinan diferentes **Métodos y Técnicas en la búsqueda y procesamiento de la información**, los fundamentales son:

A nivel teórico:

Analítico-Sintético: Para desarrollar un profundo análisis sobre las técnicas de mitigación de riesgos a través del estudio de teorías y documentos extrayendo de estos los elementos más importantes sobre dichas técnicas.

Histórico-Lógico: Para estudiar de forma analítica la trayectoria histórica real de los problemas de riesgos existentes y estudiar cómo va evolucionando a través del plan que se trace para su solución.

A nivel Empírico:

Observación: Realizar un registro visual de lo que ocurre en el proyecto SCADA Nacional con respecto a la GR clasificando y consignando los hechos y acontecimientos pertinentes de acuerdo con algún esquema previsto.

Encuesta: A través de un conjunto de preguntas estrictas se pretende obtener información sobre el proceso de GR en el proyecto SCADA Guardián del Alba.

El presente documento está estructurado en tres capítulos:

Capítulo 1: Fundamentación Teórica: En el presente capítulo se mostrarán las técnicas para la GR: planeación, identificación, análisis cualitativo y cuantitativo, planificación de la respuesta a los riesgos, seguimiento y control de los mismos, también aquí se da una breve explicación de las herramientas existentes y que características presentan.

Capítulo 2: Características, Diseño e implementación del sistema: En el presente capítulo se expondrá la aplicación de las técnicas de GR en el proyecto SCADA más conocido como Guardián del Alba. También se harán algunas especificaciones de los requisitos de software junto con una visión general de la aplicación.

Capítulo 3: Pruebas y Seguimiento de la Solución Planteada: En el presente capítulo se presentará el desarrollo de la extensión para la GR, permitiendo llegar a conclusiones a través de pruebas realizadas.

1.1 Introducción

La producción de software es una de las esferas más prometedoras en la actualidad, el hombre se esfuerza cada día por construir un mundo cada vez más informatizado, cada software que se hace debe contar con la calidad requerida y ser desarrollado en el tiempo establecido, metas elevadas pero alcanzables para cualquier equipo de desarrollo. La GP es un factor clave dentro del cual hay varios aspectos a tener en cuenta; uno de ellos es la GR.

El presente capítulo estará enfocado a mostrar detalladamente los procesos de GR, la aplicación de este concepto en el mundo y en el Polo de HA de la UCI, junto con la metodología, herramientas y técnicas seleccionadas que permitirán modelar e implementar el sistema.

1.2 Gestión de Riesgos en Proyectos Informáticos

Un riesgo es una condición futura o circunstancia que existe por fuera del control del jefe del proyecto, del proyecto y de su equipo y que puede tener un impacto negativo (amenaza) o positivo (oportunidad) por lo menos en uno de los objetivos del proyecto: costos (cumplir con los costos planificados al principio del proyecto), tiempo (obtener cada versión del software en el tiempo acordado), alcance (cumplir con las características del producto que se acordaron) y calidad (cumplir con los requerimientos de calidad pactados).

La Gestión de los Riesgos en proyectos de software implica identificar, estudiar y eliminar las fuentes de riesgo antes de que comiencen a amenazar el éxito o la finalización exitosa de un proyecto de desarrollo de software. Para lograr dicho objetivo debemos incluir los procesos relacionados con la planificación, la identificación, el análisis (cualitativo y cuantitativo), las respuestas, y el seguimiento y control de los riesgos en un proyecto; la mayoría de estos procesos se actualizan durante el proyecto, los procesos de GR del proyecto incluyen los siguientes: (1)

Planificación de la GR: decidir cómo enfocar, planificar y ejecutar las actividades de Gestión de Riesgo para un proyecto.

Identificación de Riesgos: determinar que riesgos pueden afectar al proyecto y documentar sus características.

Análisis Cualitativo de Riesgos: priorizar los riesgos para realizar otros análisis o acciones posteriores, evaluando y combinando su probabilidad de ocurrencia y su impacto.

Análisis Cuantitativo de Riesgos: analizar numéricamente el efecto de los riesgos identificados en los objetivos generales del proyecto.

Planificación de la Respuesta a los Riesgos: desarrollar opciones y acciones para mejorar las oportunidades y reducir las amenazas a los objetivos del proyecto.

Seguimiento y control de Riesgos: realizar el seguimiento de los riesgos identificados, supervisar los riesgos residuales, identificar nuevos riesgos, ejecutar planes de respuesta a los riesgos y evaluar su efectividad a lo largo del ciclo de vida del proyecto. (2)

1.2.1 Planificación de la Gestión de Riesgos

Para mejorar la posibilidad de éxito de un proyecto a la hora de gestionar los riesgos debe hacerse una planificación con un cuidado absoluto y muy explícito, por lo que el proceso de planificación en la GR debe completarse en las fases tempranas de la planificación del proyecto.

La planificación de los riesgos es el proceso que decide la forma de enfocar y planear las actividades de la administración de riesgos para un proyecto, y su principal resultado es el plan de manejo de riesgos. Un plan de manejo o administración de riesgos documenta paso a paso el procedimiento para administrar los riesgos de un proyecto; es decir, resume cómo la administración de riesgos será ejecutada en un proyecto en particular. Los elementos que se deben incluir en dicho plan de administración de riesgos son: (1)

Metodología: Se debe establecer cómo gestionar la administración de riesgo que será ejecutada en el proyecto. Determinar qué herramientas y fuentes de información están disponibles y que más convenga aplicar en el proyecto.

Roles y responsabilidades: Determinar quiénes son las personas responsables de implementar las tareas específicas y proporcionar los informes relacionados a la administración de riesgo del proyecto. Se proponen los siguientes roles y responsabilidades vinculadas a la GR:

- **Jefe del proyecto:** Es el máximo responsable de la GR de un proyecto, controlando y dando seguimiento a los mismos. En caso, de que el proyecto sea de gran magnitud y se encuentre estructurado por módulos deberá exigir a los Jefes de Línea/Equipo/Módulo la información correspondiente a la GR permitiendo la toma de decisiones sobre el control y seguimiento de los riesgos.
- **Jefe de Línea/Equipo/Módulo:** deberá integrar la información de todos los riesgos, reconsiderar su prioridad para determinar los más importantes, revisar recomendaciones para las acciones de mitigación, implementar decisiones de control, construir planes de acción e informar periódicamente al jefe del proyecto sobre la situación de los riesgos.
- **Equipo interno:** Se recomienda formar un equipo interno de GR, el cual será capaz de coordinar las actividades con el fin de identificar y analizar los riesgos, mantener la lista de los riesgos del proyecto actualizada, notificar nuevos riesgos e informar periódicamente al jefe del proyecto el estado actual de los riesgos.

Presupuesto y plazos: Determinar cuáles son los costos y plazos estimados para ejecutar las tareas relacionadas con los riesgos.

Categoría de riesgos: Cuando se analizan los riesgos es importante cuantificar el nivel de incertidumbre y el grado de pérdidas asociado con cada riesgo. Para hacerlo, se consideran diferentes categorías de riesgos:

- **Los riesgos del proyecto** amenazan al plan del proyecto. En otras palabras, si los riesgos del proyecto se hacen realidad, es probable que la planificación del proyecto se retrase y que los costos aumenten. Los riesgos del proyecto identifican los problemas potenciales de presupuesto, planificación temporal, personal (asignación y organización), recursos, cliente y requisitos y su impacto en un proyecto de software.
- **Los riesgos técnicos** amenazan la calidad y la planificación temporal del software que hay que producir. Si un riesgo técnico se convierte en realidad, la implementación puede

Capítulo 1: Fundamentación Teórica

llegar a ser difícil o imposible. Los riesgos técnicos identifican problemas potenciales de diseño, implementación, de interfaz, verificación y de mantenimiento. Además, las ambigüedades de especificaciones, incertidumbre técnica, técnicas anticuadas y las "tecnologías punta" son también factores de riesgo. Los riesgos técnicos ocurren porque el problema es más difícil de resolver de lo que pensábamos.

- **Los riesgos del negocio** amenazan la viabilidad del software a construir. Los riesgos del negocio a menudo ponen en peligro el proyecto o el producto.

Probabilidad de riesgo e impacto: Cuáles son las probabilidades y los impactos de los riesgos que serán evaluados. Cuáles son las técnicas cualitativas o cuantitativas que serán utilizadas para evaluar los riesgos.

Documentación de los riesgos: Determinar los formatos de los reportes y los procesos que serán utilizados para las actividades de la administración de riesgos.

Como continuidad al proceso de planificación de riesgos, se procede a la identificación inicial de los riesgos, posteriormente, se analizan, priorizando los más importantes y finalmente se plantean estrategias de respuesta a los riesgos, dándole seguimiento de manera que sean controlados. (3)

1.2.2 Identificación de riesgos

La identificación de riesgos es el proceso de comprender que eventos potencialmente podría dañar o mejorar a un proyecto. Es importante identificar los riesgos potenciales lo más pronto posible. Una buena identificación de los riesgos involucra el examinar todas las fuentes de riesgo y la participación de todas las personas que integran el proyecto ya sean internos o externos. Otro factor importante es la buena calidad de la información y el comprender cómo y dónde estos riesgos han tenido o pueden tener su efecto. Aunque no siempre es posible obtener toda la información necesaria, esta debe ser lo más amplia, integral, precisa y oportuna en la medida que lo permitan los recursos disponibles u otros factores limitativos.

La Identificación de Riesgos es un proceso iterativo puesto que se pueden descubrir nuevos riesgos a medida que el proyecto avanza a lo largo de su ciclo de vida. El proceso Identificación de Riesgos suele llegar al proceso Análisis Cualitativo de Riesgos, en donde se le da una prioridad a cada riesgo para que posteriormente ir atendiendo en dependencia de dicha prioridad. Como alternativa, puede llevar directamente al proceso Análisis Cuantitativo de Riesgos cuando lo dirige un director de riesgos experimentado. En algunas ocasiones, simplemente la identificación de un riesgo puede sugerir su respuesta, y esto debe registrarse para realizar otros análisis y para su implementación en el proceso Planificación de la Respuesta a los Riesgos.

El objetivo de este proceso de identificación de riesgos es la elaboración de una lista con todos los posibles riesgos con los que el equipo deberá enfrentarse. Dicha lista debe ser lo más extensa posible y deberá cubrir todas las áreas del proyecto.

Para el proceso de identificación de riesgos es necesario tener definido los Datos de Entradas y las Actividades.

Los *Datos de Entrada* es toda aquella información que permita identificar el riesgo, basado en lecciones aprendidas en proyectos anteriores, experiencia del equipo, estado actual e historial del proyecto en cuanto ha estado de los riesgos. Para la extracción de los datos de entradas es necesario al inicio del proyecto, realizar sesiones dirigidas o talleres que permitan obtener información sobre las percepciones que tienen el equipo de proyecto y los clientes acerca de los riesgos.

Capítulo 1: Fundamentación Teórica

Las *Actividades* son las diferentes técnicas que se pueden realizar para la identificación de riesgos. Las cinco técnicas más utilizadas son: la Tormenta de Ideas, el Método Delphi, las Entrevistas, el Análisis de Causa y Efecto y el Análisis FODA (Fortalezas, Debilidades, Oportunidades y Amenazas). (3)

Tormenta de Ideas: Este método es un proceso que agrupa a todos los integrantes del equipo para fomentar la creatividad y tratar de aprovechar los conocimientos del grupo para asegurar la identificación de las causas posibles de un problema. Como orientaciones para realizar una tormenta de ideas cabe mencionar las siguientes:

- ✓ Anotar las ideas y hacer cumplir las normas; y animar la reunión.
- ✓ Dedicar los primeros minutos a reflexionar sobre el problema.
- ✓ Enunciar las ideas por turno y anotarlas sin hacer comentarios.
- ✓ Al final de la reunión revisar, agrupar y seleccionar las ideas. (4)

Delphi: considerado como uno de los métodos subjetivos de pronóstico más confiables, constituye un procedimiento para confeccionar un cuadro de la evolución de situaciones complejas, a través de la elaboración estadística de las opiniones de expertos en el tema tratado. Las tres principales características del método son las siguientes:

Anonimato: se expresa a través del no conocimiento de las respuestas, puesto que los miembros del grupo contestan las preguntas sin confrontarse incluso sin conocerse entre sí.

Retroalimentación controlada: después de cada ronda de preguntas se tabulan las respuestas y se procesan de forma tal, que antes de la siguiente ronda los participantes pueden evaluar los resultados de la ronda anterior, así como las razones dadas para cada respuesta y su dispersión del promedio. Se tiene la opinión de que los encuestados están influidos en cierto grado después de analizar las respuestas de sus compañeros del grupo, aumentando el acuerdo al transcurrir varias iteraciones del proceso.

Respuesta estadística del grupo: entre cada ronda de preguntas, la información obtenida se procesa por medio de técnicas estadístico-matemáticas, como, por ejemplo, el diseño experimental no paramétrico, las que dotan al investigador de un instrumento objetivo y concreto en el cual pueden apoyarse para tomar una decisión final. (5)

Entrevistas: El método de las entrevistas orientadas a riesgos con varios de los involucrados (personas que están involucradas por el proyecto) puede ayudar a identificar riesgos no identificados durante actividades normales de la planificación. Los registros de las entrevistas previas al proyecto deben estar disponibles.

Entrevistar es una técnica que consiste en buscar hechos mediante la colección de información mediante: reuniones cara a cara, teléfono, correo electrónico, chat.

Entrevistar personas con experiencia de proyecto similares, es una herramienta importante para identificar riesgos potenciales. (6)

Análisis de Causa y Efecto: El método Causa y Efecto es una investigación de las causas esenciales de los riesgos. Permite refinar la definición de riesgo y permite agrupar los riesgos por causas. Usa la técnica de espina de pescado, instrumento eficaz para el análisis de las diferentes causas

Capítulo 1: Fundamentación Teórica

que ocasionan el problema. Su ventaja consiste en poder visualizar, en equipo, las causas principales y secundarias de un problema analizando procesos en búsqueda de mejoras, que pueden estar presentes en un problema.

FODA: El método FODA significa Fortalezas, Oportunidades, Debilidades y Amenazas, permite identificar los riesgos positivos y negativos que aplican a un proyecto. Las *Fortalezas* son recursos internos que posee el proyecto en mejores condiciones que su competencia y que la hacen tener ventaja sobre los demás, y por lo tanto la hace rigurosa en su accionar; el proyecto puede emplear estos elementos para lograr sus objetivos y mejorar su posición competitiva en el mercado. Las *Oportunidades* son aquellas situaciones del entorno que son fuerzas que surgen desde fuera de las fronteras de un proyecto, pero que afectan sus decisiones y acciones internas, así como sus actividades y desarrollo. Las *Amenazas* son aquellas situaciones del entorno desfavorables para el proyecto que pueden afectar negativamente la marcha de la institución, de no tomarse las medidas necesarias en el momento oportuno. Si el proyecto no toma una determinada acción estratégica ante esta tendencia desfavorable, que proviene del entorno, puede llevarla a su estancamiento o incluso su desaparición. Las *Debilidades* son limitaciones, defectos o inconsistencias en la institución, que constituyen un obstáculo para la consecución de los objetivos y baja la calidad de la gestión. Son recursos de tipo interno que poseen las instituciones y que por el solo hecho de poseerlo la hacen vulnerable en su accionar en relación a su competencia. Aspectos internos en los que debemos mejorar para lograr una posición más competitiva. (2)

1.2.3 Análisis cualitativo de los riesgos

En esta etapa se examina la lista de riesgos obtenida en el proceso anterior de identificación de riesgos y le asigna una prioridad, registrando el orden final en la lista de riesgos. Utilizando esta lista, se puede determinar los riesgos que son más importantes y reservar recursos para planificar y ejecutar una estrategia específica, pudiendo identificar además, los riesgos que por su poca prioridad, pueden quitarse de la lista.

El análisis cualitativo está basado en la opinión, en la intuición y en la experiencia, para estimar la probabilidad de ocurrencia de potenciales riesgos y medir la intensidad de las pérdidas o ganancias potenciales, es simple, intuitiva, rápida y económica.

El proyecto puede mejorar su rendimiento de manera positiva centrándose en los riesgos de alta prioridad. El Análisis Cualitativo de Riesgos evalúa la prioridad de los riesgos identificados usando la probabilidad de ocurrencia, el impacto correspondiente.

A medida que el proyecto se acerca al final y las circunstancias del mismo van cambiando, la identificación y el análisis de riesgos se repetirán y la lista de riesgos se modificará. Puede que surjan nuevos riesgos y puede que los riesgos más antiguos que han bajado de prioridad se eliminen o “desactiven”. Estos análisis de riesgos deben llevarse a cabo como apoyo para la asignación de prioridades empleada en la toma de decisiones y nunca debe realizarse pensando únicamente en el análisis. (7)

Datos de entrada: El equipo recurrirá a la experiencia e información en la literatura, en la selección de las escalas para medir cualitativamente el riesgo. La información necesaria para priorizar los riesgos cualitativamente se recogerán de la identificación realizada, donde se obtuvo la lista de los riesgos.

Actividades: Existen varias técnicas cualitativas para asignar las prioridades a una lista de riesgos. Dentro de las técnicas más utilizadas se encuentran la evaluación de probabilidad e impacto de los riesgos, matriz de probabilidad e impacto, Evaluación de la Calidad de los Datos sobre Riesgos, Categorización de Riesgos y Evaluación de la Urgencia de los Riesgos.

Capítulo 1: Fundamentación Teórica

- **Matriz de Probabilidad e Impacto:** Esta técnica de la matriz de probabilidad e impacto calcula la prioridad del riesgo basándose en una combinación de probabilidad e impacto, asumiendo que los riesgos con alta probabilidad y alto impacto son más priorizados.

La probabilidad de un riesgo debe ser mayor que cero o el riesgo no representa una amenaza para el proyecto. La tabla 1 describe la escala a utilizar para definir la probabilidad de un riesgo.

Tabla 1: Escala de probabilidad de ocurrencia.

Escala de Probabilidad de Ocurrencia		
Escala	Definición	Descripción
0.1	Muy Improbable	La posibilidad de que ocurra es 0, imposible.
0.3	Poco Probable	La probabilidad de que ocurra es mínima.
0.5	Probable	La probabilidad de que ocurra es de un 50%.
0.7	Altamente Probable	Es completamente posible que ocurra.
0.9	Casi Probable	Es muy alta la probabilidad de que ocurra.

Cualquier riesgo puede tener efectos en costo, en el atraso del cronograma de la entrega de cada uno de los entregables y en la calidad. Para medir el impacto se usa una escala de valores no lineales. La tabla 2 describe la escala a utilizar cuando se trata el impacto negativo de los riesgos.

Tabla 2: Escala de impacto negativo

Escala de impacto negativo					
Objetivo	Escala				
	Muy Poco (0.05)	Poco (0.10)	Moderado (0.20)	Alto (0.40)	Muy Alto (0.80)
Costo	Incremento Insignificante	Incremento en costo < 5%	Incremento en costo entre 5-10%	Incremento en costo entre 10-20%	Incremento en costo > 20%
Tiempo	Insignificante	Retraso < 5%	Retraso global entre 5-10%	Retraso global entre 10-20%	Retraso global entre > 20%
Calidad	Escasamente apreciable	Solo aplicaciones muy exigentes se afectan	Reducción de calidad que requiere aprobación	Reducción de calidad inaceptable por el proyecto	Fin del proyecto

Capítulo 1: Fundamentación Teórica

Finalmente teniendo la probabilidad y el impacto, se calcula la exposición o la amenaza al riesgo; que no es más que el resultado de la multiplicación entre estos dos valores que se obtienen de la tabla 1 y 2. Dicho resultado se encuentran entre:

- 0.15 y 0.72 es un riesgo con una prioridad *alta*, estos son los riesgos que van a tener un alto impacto sobre el proyecto y se requiere asignarle prioridad sobre los demás, planeando estrategias agresivas de mitigación.
- 0.06 y 0.14 se clasificará como un riesgo de prioridad *media* o *moderada*.
- 0.01 y 0.05 se clasificará como un riesgo menos importante, es decir, prioridad *baja*.

Para un mejor análisis y entendimiento es muy práctico crear una matriz en donde tenga todas las posibles combinaciones y las clasifique según su prioridad, baja, media o alta. En la tabla 3 se muestra dicha matriz, en donde el área de gris oscura representa los riesgos más altos, el área de gris intermedio representa los riesgos menos importantes y el área de gris más claro representa los riesgos moderados.

Tabla 3: Matriz de probabilidad e impacto

PROBABILIDA	AMENAZA				
0.9	0.05	0.09	0.18	0.36	0.72
0.7	0.04	0.07	0.14	0.28	0.56
0.5	0.03	0.05	0.10	0.20	0.4
0.3	0.02	0.03	0.06	0.12	0.24
0.1	0.01	0.01	0.02	0.04	0.08
IMPACTO	0.05	0.10	0.20	0.40	0.80

- **Evaluación de la Calidad de los Datos:** Un análisis cualitativo de riesgos requiere datos exactos y sin errores para que sea creíble. El análisis de la calidad de los datos sobre riesgos es una técnica para evaluar el grado de utilidad de los datos sobre los riesgos para la GR. Implica examinar el grado de entendimiento del riesgo, y la exactitud, calidad, fiabilidad e integridad de los datos sobre los riesgos del proyecto.

El uso de datos sobre riesgos de baja calidad puede llevar a un análisis cualitativo de riesgos de poca utilidad para el proyecto. Si la calidad de los datos es imperdonable, puede ser necesario recopilar datos mejores. A cada rato, la recogida de información acerca de los riesgos es difícil, y consume tiempo y recursos que exceden lo planificado originalmente.

- **Categorización de Riesgos:** los riesgos del proyecto pueden categorizarse por fuentes de riesgo, área del proyecto afectada u otra categoría útil (fase del proyecto) para determinar las áreas del proyecto que están más expuestas a los efectos de la incertidumbre. Agrupar los riesgos por causas comunes puede contribuir a desarrollar respuestas efectivas a los riesgos.

Capítulo 1: Fundamentación Teórica

- **Evaluación de la Urgencia de los Riesgos:** Los riesgos que requieren respuestas a corto plazo pueden ser considerados como más urgentes, en caso contrario no. Entre los indicadores de prioridad pueden incluirse el tiempo para dar una respuesta a los riesgos, los síntomas y señales de advertencia, y la calificación del riesgo.
- **Evaluación de Probabilidad e Impacto de los Riesgos:** La evaluación de probabilidad de los riesgos investiga que tan probable es que un riesgo en específico ocurra. La evaluación del impacto de los riesgos investiga el posible efecto sobre un objetivo del proyecto, como tiempo, coste, alcance o calidad, incluidos tanto los efectos negativos por las amenazas que implican, como los efectos positivos por las oportunidades que generan.

Para cada uno de los riesgos identificados se evalúan la probabilidad y el impacto. Los riesgos pueden ser evaluados en entrevistas o reuniones con participantes seleccionados por su familiaridad con los riesgos. Entre ellos se incluyen los miembros del equipo del proyecto y expertos ajenos al proyecto. Un miembro del proyecto experimentado puede dirigir la discusión, puesto que los participantes pueden tener poca experiencia en la evaluación de riesgos. (2)

1.2.4 Análisis cuantitativo de los riesgos

El Análisis Cuantitativo de Riesgos se realiza respecto a los riesgos priorizados en el Análisis Cualitativo, por tener un posible impacto significativo sobre las demandas concurrentes del proyecto; analiza el efecto de esos riesgos y les asigna una calificación numérica. Generalmente sigue al proceso Análisis Cualitativo puesto que en algunos casos, es probable que no haga falta realizar el proceso de Análisis Cuantitativo de Riesgos para implementar respuestas satisfactorias a los riesgos del proyecto.

Debe repetirse después de la Planificación de la Respuesta, también como parte del Seguimiento y Control, para determinar si los riesgos generales del proyecto han sido reducidos satisfactoriamente.

En algunos proyectos, el equipo puede poner en práctica el Análisis Cuantitativo, dependiendo de la naturaleza del proyecto, la disponibilidad de tiempo y dinero, también influyen en el tipo de técnica a utilizar. Los proyectos grandes y complejos que involucran tecnología de punta requieren la aplicación de técnicas cuantitativas. Las principales técnicas para el análisis cuantitativo exigen la recolección de datos y técnicas de modelamiento, entre las principales, se encuentran: el análisis de árboles de decisión, la simulación, el análisis de sensibilidad y el análisis del valor monetario esperado.

Análisis mediante árbol de decisiones: Esta técnica normalmente se estructura usando un diagrama de árbol de decisiones que describe una situación que se está considerando, y las implicaciones de cada una de las opciones disponibles y los posibles escenarios. Incorpora el coste de cada opción disponible, las probabilidades de cada escenario posible y la recompensa de cada camino lógico alternativo. Al resolver el árbol de decisiones se obtiene el valor monetario esperado (u otra medida de interés para el proyecto) correspondiente a cada alternativa, cuando todas las recompensas y las decisiones subsiguientes son cuantificadas. (8)

Análisis de sensibilidad: El análisis mediante la sensibilidad se refiere al impacto que sobre los indicadores económicos del proyecto tendrían las posibles variaciones de los parámetros críticos o bases consideradas del proyecto (inversiones, precios, costos, tiempo), tomados individualmente. Para este análisis, se toman los valores máximos y mínimos que pudieran alcanzar los diferentes parámetros.

Modelos y Simulación: Esta técnica es una herramienta de investigación y planeamiento. Examina todas las posibles combinaciones derivadas de la variación de los parámetros de un proyecto. Una simulación de proyecto usa un modelo que traduce las incertidumbres especificadas a un nivel deta-

Capítulo 1: Fundamentación Teórica

llado del proyecto en su impacto posible sobre los objetivos del proyecto. Las simulaciones normalmente se realizan usando la técnica Monte Carlo.

En una simulación, el modelo del proyecto se calcula muchas veces, utilizando valores de entrada seleccionados al azar de una función de distribución de probabilidad que se elige para cada iteración de las distribuciones de probabilidad de cada variable. Se calcula una distribución de probabilidad que sería el coste total o fecha de conclusión. En el análisis mediante la simulación de Monte Carlo variamos un parámetro a la vez. Cuando analizamos por escenarios, escogemos un número limitado de combinación de variables. Si queremos examinar todas las posibles combinaciones derivadas de la variación de los parámetros de un proyecto, debemos recurrir a otros métodos estadísticos, tales como el método de Monte Carlo. (9)

Valor Monetario Esperado: El análisis del valor monetario esperado es un concepto estadístico que calcula el resultado promedio cuando el futuro incluye escenarios que pueden ocurrir o no (es decir, análisis con incertidumbre). El valor monetario esperado de las oportunidades generalmente se expresará con valores positivos, mientras que el de los riesgos será negativo. El valor monetario esperado se calcula multiplicando el valor de cada posible resultado por su probabilidad de ocurrencia, y sumando los resultados. Este tipo de análisis se usa comúnmente en el análisis mediante árbol de decisiones. Se recomienda el uso del modelado y la simulación para el análisis de los riesgos de costes y del cronograma, porque son más efectivos y están menos sujetos a errores de aplicación que el análisis del valor monetario esperado.

1.2.5 Planificación de la respuesta a los riesgos

Una vez identificados los riesgos en un proyecto y conocidos sus posibles efectos o daños a través del proceso de análisis, debe realizarse la gestión que es equivalente a determinar la respuesta adecuada a cada riesgo.

La Planificación de la Respuesta a los Riesgos aborda los riesgos en función de su prioridad, introduciendo recursos y actividades en el presupuesto, cronogramas y plan de GP, según sea necesario.

Las respuestas de los riesgos planificadas deben ser coherentes con la importancia del riesgo, tener un coste efectivo en relación al desafío, ser aplicadas a su tiempo, ser realistas dentro del contexto del proyecto, estar acordadas por todas las partes implicadas, y a cargo de una persona responsable. A menudo, es necesario seleccionar las mejores respuestas a los riesgos entre varias opciones. (1)

Datos de Entradas: Entre los componentes más importantes del plan de gestión de riesgo se incluyen los roles y responsabilidades, el tiempo y el presupuesto necesario para la gestión de los riesgos del proyecto y la definición del análisis de riesgo; el registro de riesgos se desarrolla por primera vez en el proceso Identificación de Riesgos, y se actualiza durante los procesos Análisis Cualitativo de Riesgos y Análisis Cuantitativo de Riesgos. Es posible que el proceso Planificación de la Respuesta a los Riesgos tenga que remitirse a los riesgos identificados, las causas de los riesgos, las listas de posibles respuestas, los síntomas y las señales de advertencia para desarrollar las respuestas a los riesgos.

Una de las entradas más importantes a la Planificación de la Respuesta a los Riesgos es la lista de prioridades o clasificaciones relativas de los riesgos del proyecto, lo que requiere respuesta a corto plazo, análisis y respuestas adicionales.

Actividades: Una técnica sencilla pero eficaz para vigilar el riesgo consiste en desarrollar una lista de los riesgos más importantes para el proyecto. Lo fundamental es identificar una cantidad limitada

Capítulo 1: Fundamentación Teórica

de los riesgos importantes que deben administrarse (por lo general 10 o menos). Aunque el equipo desee administrar más de 10 riesgos, es más efectivo centrarse en primer lugar en un número reducido de riesgos importantes y luego administrar los menos importantes cuando el primer grupo ya está controlado.

Existen diversos enfoques para reducir la exposición al riesgo. Entre estos se cuentan con:

Evitar: Es cambiar el plan del proyecto para eliminar el riesgo o para proteger sus objetivos, del impacto sin necesidad de realizar actividades arriesgadas. Ejemplos: reducir el alcance, agregar recursos, agregar tiempo, adoptar enfoques conocidos y no innovadores y evitar contratistas no bien conocidos.

Transferir: A veces, un riesgo puede transferirse para que pueda ser administrado por otra entidad fuera del proyecto. La transferencia del riesgo no significa que el riesgo se haya eliminado. En general, una estrategia de transferencia de riesgos generará riesgos que seguirán necesitando una administración proactiva pero que reducen el grado de riesgo a un nivel aceptable.

Mitigar: La mitigación de riesgos implica acciones y actividades que se realizan con anticipación para evitar que se produzca un riesgo o para reducir el impacto o las consecuencias a un nivel aceptable. Ejemplos: Adoptar procesos menos complejos, escoger un proveedor más estable, agregar más recursos, obtener más tiempo y desarrollar prototipos.

Aceptar: En algunos riesgos ya no es posible intervenir con medidas preventivas ni correctivas efectivas, pero aún así el equipo decide simplemente aceptar el riesgo para materializar la oportunidad. Así pues, el plan deberá incluir los motivos que han empujado al equipo a aceptar el riesgo sin desarrollar ningún plan de mitigación o contingencia. (3)

1.2.6 Seguimiento y control de Riesgos

Proceso que se ocupa del seguimiento de los riesgos identificados, de la supervisión de los riesgos residuales y de la identificación de nuevos riesgos, asegurando la ejecución de los planes de riesgo y evaluando su eficacia en la reducción de los mismos. La supervisión y control de los riesgos registra las métricas que están asociadas con la implementación de los planes de contingencia. Este es un proceso que se realiza continuamente durante todo el ciclo de vida del proyecto. Los riesgos cambian a medida que el proyecto madura; nuevos riesgos aparecen o riesgos previstos desaparecen

Los procesos de supervisión y control de los riesgos proveen información que ayuda a tomar decisiones eficaces en forma anticipada a la ocurrencia del riesgo. El propósito de supervisar es determinar si:

- Las respuestas a los riesgos ha sido implementadas como fueron planificadas.
- Las acciones de las respuestas a los riesgos son tan efectivas como se esperaba o si se deben desarrollar nuevas respuestas.
- Las hipótesis del proyecto son aún válidas.
- Un disparador de riesgo ha ocurrido.
- Han aparecido u ocurrido riesgos que no habían sido identificados.

1.3 Gestión de Riesgos en la Producción de Software

Desarrollar proyectos de software es un negocio que involucra a muchos riesgos. Un instituto u organización de desarrollo de software que no cuente con un adecuado control de su proyecto; siempre se encontrara con atrasos en su plan de trabajo o en sus entregables, con resultado que son altamente costosos. Tratar de hacer un mejor trabajo de desarrollo de software, sin contar con un proceso de GR bien definido, trae consigo que las buenas ideas no puedan ser integradas eficientemente, desperdicia demasiado tiempo y deberíamos de estar enfocados en las mejoras del desarrollo de software.

1.3.1 Gestión de Riesgo en el Mundo

El proceso de GR es unos de los pilares de todo proyecto, y para que todo funcione como es debido debemos asegurar el perfecto desarrollo del mismo, una equivocación puede traer consigo grandes consecuencias negativas. En el mundo, numerosos son los proyectos de software que se desarrollan; para el éxito futuro; estos se encargan de construir metodologías, técnicas y herramientas para la realización del proceso de GR para sus propios beneficios.

Algunos de los ejemplos de herramientas más utilizadas son: MSAT y GxSGSI.

La primera es una Herramienta de Evaluación de Seguridad de Microsoft (MSAT) gratuita. Presenta un listado de cuestiones ordenadas por prioridad para minimizar los riesgos. MSAT permite fortalecer la seguridad de su entorno informático y de su negocio de manera fácil y efectiva.

Esta Herramienta consta de más de 200 preguntas que abarcan infraestructura, aplicaciones, operaciones y usuarios. La evaluación está diseñada para identificar los posibles riesgos de negocio de su organización y las medidas de seguridad implementadas para mitigarlos. Centrándose en problemas comunes, las preguntas han sido desarrolladas para proporcionar una evaluación de seguridad de alto nivel de la tecnología, procesos y usuarios que participan en el negocio. (10)

El programa GxSGSI, es un software desarrollado en Visual Basic bajo SQL Server destinado a realizar Análisis de Riesgos de Seguridad. En su versión avanzada, incluye un componente de simulación de fallos, mediante el cual puede generar, de manera ficticia, un fallo en cualquiera de los equipos del sistema de información con el fin de analizar las amenazas y vulnerabilidades que lo afectan o podrían afectar y detectar el número y coste de las contramedidas necesarias.

Sus principales características son:

- Ha sido diseñado para automatizar, agilizar y realizar por completo el análisis de riesgos de seguridad de una organización.
- Su versatilidad de configuración le permite trabajar con multitud de revisiones.
- Es multiempresa.
- Reduce el tiempo y de captura de datos hasta en un 80%
- Proporciona una base sólida para diseñar cualquier plan de continuidad ante desastres. (11)

1.3.2 Gestión de Riesgos en el Polo de Hardware y Automática

Con el objetivo de conocer si se trabaja correctamente para evitar los riesgos en el polo de HA, y de qué forma se hace en busca de alternativas que sean de ayuda para encontrar deficiencias y sugerencias que puedan ser resueltas y puesta en práctica, por lo que se encuestaron 9 líderes de proyectos, siendo así la mayoría de los proyectos que conforman el polo, estos proyectos son: SCADA Galba, SIGGE, SCADA-Etecsa, Supervisión Energética-UNE, SCADA-Hospitales, SCADA-Edificios Inteligentes, Siprogas, SCIA, Retrieve.

Todos los encuestados reconocen la importancia de este proceso, consideran que es muy importante en el desarrollo y ciclo de vida del software. Los resultados de la encuesta muestran el nivel de uso y conocimiento que se tiene sobre los procesos de la GR, en busca de información que permita mejorar estos procesos e incorporarlos a una GR exitosa.

En las encuestas realizadas se demuestra que el 100% de los líderes encuestados gestionan los riesgos de sus proyectos respectivamente. El problema radica en que ninguno de los proyectos encuestados pone en práctica correctamente el proceso de GR, es decir, solamente el 22% de dichos proyectos realiza una previa planificación para afrontar los riesgos, el 33% le asigna una prioridad a los riesgos que hayan sido identificados, brindando una respuesta de solución a los de mayor prioridad y el 66% de estos proyectos facilitan un seguimiento y control a los riesgos que fueron mitigados; esto nos da una idea de que todavía hay mucho camino por recorrer en cuanto a la GR en el Polo de HA, y que puede influir mucho en la mayoría de las tareas asignadas por nuestros líderes.

De acuerdo al criterio de cuáles son los procesos de GR que realizan los proyectos del Polo de HA se obtiene en la figura 1:

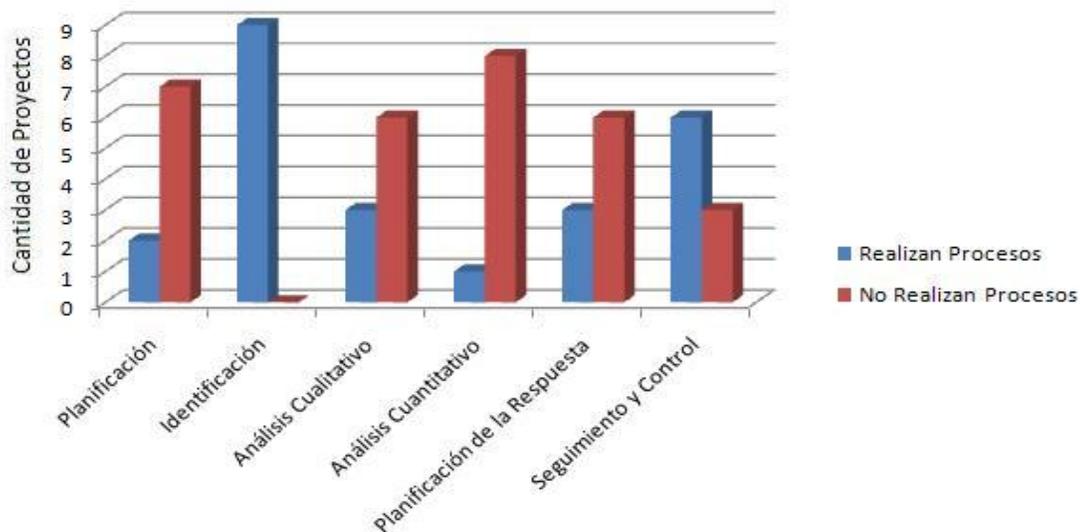


Figura 1: Proceso de GR que realizan los proyectos del Polo de HA

En la pregunta donde se plantea que si se utiliza alguna herramienta para GR, se encuentran criterios diversos, solo el 22.2% representa los proyectos que utilizan alguna herramienta para este fin, quiere decir que en la mayoría de estos proyectos encuestados no presentan ningún registro de los riesgos que le han afectado durante el ciclo de vida de software. El resultado de los proyectos que utilizan herramientas para la GR se pueden ver en la figura 2:



Figura 2: Proyectos que utilizan herramientas para la GR

1.4 Sistema Operativo GNU/Linux: Distribución Debian

GNU es un acrónimo recursivo que se traduce como "GNU No es Unix". Este proyecto, iniciado por Richard Stallman en 1983, pretende desarrollar un sistema operativo completamente libre equivalente o compatible con UNIX.

Linux fue adoptado como núcleo del sistema GNU, y junto a él y otros muchos programas, constituye el actual sistema operativo GNU/Linux. El resultado de la combinación GNU y Linux es un sistema operativo libre, multiplataforma, multiusuario y multitarea. GNU/Linux es completamente configurable y optimizable, posee un funcionamiento muy rápido ya que es capaz de utilizar todas las posibilidades de hardware del sistema.

Entre las distribuciones de GNU /Linux se encuentra la distribución Debian, mantenida solamente por voluntarios, es decir, sin un enfoque comercial, lo que trae consigo ventajas y desventajas.

En primer lugar, las personas que se dedican a Debían tienen una alta motivación en participar en la misma, y se actualiza la distribución diariamente, apareciendo paquetes nuevos de software constantemente. (12)

Al mismo tiempo, existe un compromiso de calidad, no se desea distribuir software con errores.

En segundo lugar, dada su actitud abierta a la participación de todos, en el mismo espíritu original de Linux, constantemente hay personas que se unen a Debían para participar aportando su granito de arena, no solamente haciendo paquetes de programas, sino colaborando en el Servidor de Web, traduciendo documentación de Debían, documentando fallos, o ayudando a los usuarios a través de las listas de correo que mantiene la comunidad.

Como desventajas tiene un mayor componente técnico que otras distribuciones.

También, es posible que ciertos paquetes no estén tan actualizados como debieran, quizás porque sus desarrolladores han dejado de actualizarlos y nadie se ha hecho cargo. Sin embargo esto es algo que todos los desarrolladores tratan de evitar y, aunque cada desarrollador mantiene sus paquetes, no es raro que otro desarrollador (incluso un usuario) envíe una nueva versión del paquete para arreglar un problema o actualizarlo.

Capítulo 1: Fundamentación Teórica

El presente trabajo de diploma será desarrollado a partir de las herramientas que brinda el software libre. Buscando la independencia tecnológica que este brinda al posibilitar la libertad de uso y distribución de los programas sin incurrir en litigios de licenciamiento o asuntos legales.

Se decidió utilizar Debian porque a diferencia de otras distribuciones tiene un magnífico soporte de estabilidad en las aplicaciones (no requieren ser compiladas en la máquina que las esté usando). Los módulos del LDAP (Lightweight Directory Access Protocol) se pueden ejecutar sin problemas permitiendo que los usuarios usen sus sesiones en cualquier máquina dentro del área de trabajo, ahorrando recursos de hardware. (13)

1.5 Metodología de Desarrollo de Software: OpenUp

El desarrollo de software no es sin dudas una tarea fácil. Como resultado a este problema ha surgido una alternativa desde hace mucho tiempo: la Metodología. Desarrollar un buen software depende de un sin número de actividades y etapas, donde el efecto y el impacto de elegir la mejor metodología para un equipo es crucial y vital para el éxito del producto. El papel importante y destacado de las metodologías es, sin duda alguna, esencial en un proyecto y en el paso inicial, que debe encajar en el equipo, guiar y organizar actividades que den lugar a las metas trazadas en el grupo.

OpenUp (Proceso Unificado Abierto) es un marco del proceso del desarrollo del software de código abierto, diseñado para pequeños equipos organizados, quienes quieren tomar una aproximación ágil del desarrollo, que en un cierto plazo, se espera que cubra un amplio sistema de necesidades para los proyectos de desarrollo.

OpenUp es un proceso iterativo para el desarrollo de software, incluye solamente el contenido del proceso fundamental, es decir es la mínima cantidad de procesos para un equipo pequeño, puede ser manifestado como proceso entero para construir un sistema y es utilizado como base para agregar o para adaptar más procesos, en otras palabras puede ser extendido y personalizado para propósitos específicos.

OpenUP posee algunos beneficios:

- Es apropiado para proyectos pequeños y de bajos recursos permite disminuir las probabilidades de fracaso en los proyectos pequeños e incrementar las probabilidades de éxito.
- Permite detectar errores tempranos a través de un ciclo iterativo.
- Evita la elaboración de documentación, diagramas e iteraciones innecesarios requeridos en la metodología RUP.
- Por ser una metodología ágil tiene un enfoque centrado al cliente y con iteraciones cortas.(14)

1.6 Lenguaje de Programación: Python

Python es un lenguaje de programación creado por Guido Van Rossum a principios de los años 90. Es un lenguaje similar a Perl, pero con una sintaxis muy limpia y que favorece un código legible.

Se trata de un lenguaje interpretado o de script, con tipado dinámico (se refiere a que no es necesario declarar el tipo de dato que va a contener una variable determinada), fuertemente tipado (no se permite violaciones en los tipo de datos, es decir tratar a una variable como si fuera de un tipo distinto al que ya tiene), multiplataforma y orientado a objetos.

Capítulo 1: Fundamentación Teórica

Python presenta unas series de ventajas que lo hacen muy atractivo, tanto para su uso profesional como para el aprendizaje de la programación, entre las más interesantes tenemos:

- Python es un lenguaje muy expresivo, es decir, los programas Python son muy compactos: un programa Python suele ser bastante más corto que su equivalente en lenguajes como C.
- Python es muy legible. La sintaxis de Python es muy elegante y permite la escritura de programas cuya lectura resulta más fácil que si utilizáramos otros lenguajes de programación.
- Python ofrece un entorno interactivo que facilita la realización de pruebas y ayuda a despejar dudas acerca de ciertas características del lenguaje.
- El entorno de ejecución de Python detecta muchos de los errores de programador que es capan al control de los compiladores y proporciona información muy rica para detectarlos y corregirlos.
- Posee un rico juego de estructuras de datos que se pueden manipular de modo sencillo. (15)

1.7 Librería de Genshi.

La librería de Python conocida como Genshi, proporciona un conjunto integrado de componentes para el análisis, procesamiento y generación de códigos HTML, XML u otro contenido de texto para el desarrollo de la Web.

La principal característica es que es un lenguaje de plantilla que es inteligente sobre marcas: a diferencia del modelo convencional de lenguajes que solo tratan con bytes y caracteres. Genshi sabe la diferencia entre las etiquetas, atributos, nodos de texto y real, y utiliza ese conocimiento a su ventaja (16).

1.8 Base de datos SQLite

SQLite es una librería escrita en lenguaje C que implementa un manejador de base de datos SQL instruido. Los programas que se enlacen con la librería SQLite pueden tener acceso a una base de datos SQL, sin tener que ejecutar un Sistema Administrador de Base de Datos separado.

La distribución viene con un programa de acceso (independiente) de línea de comandos (sqlite.exe), que puede ser usado para administrar una base de datos SQLite y que sirve como un ejemplo de cómo usar la librería SQLite; también es libre, gratis y el código fuente es del dominio público. (17)

1.9 Herramientas a utilizar

1.9.1 Herramienta de modelado UML: NetBeans

NetBeans es un Entorno de Desarrollo Integrado (IDE) desarrollado usando la Plataforma NetBeans. Constituye un proyecto de código abierto de gran éxito, que fue fundado en junio del 2000 por la compañía Sun Microsystems que continúa siendo el patrocinador principal. SDE para NetBeans es una herramienta de modelado UML que soporta el ciclo de vida completo del desarrollo de software: análisis, diseño, implementación, pruebas y despliegue. Permite la captura de requisitos, el dibujo de diagramas UML, la realización de ingeniería inversa desde Java a UML y la generación de código Java. (18)

1.9.2 Herramienta de Gestión de Proyecto: Trac

El Trac es una herramienta de gestión de proyectos, que a diferencia de otros sistemas parecidos y más completos como el gforge, se caracteriza por ser intuitivo, ligero, sencillo de instalar y mantener; integrando el repositorio que mantiene la documentación/código a través de una interfaz web.(17) Ayuda a documentar y ver el progreso del proyecto, mantiene las distintas versiones utilizando el sistema de control de versiones Subversion (SVN), controla las tareas y defectos mediante la asignación de “tickets” y puede ser fácilmente modificado con la instalación de extensiones, en inglés, extensión.

Permite gestionar la información obtenida o generada durante el desarrollo del proyecto trayendo consigo un mayor rendimiento, al reducir los tiempos para el almacenamiento, localización, y búsqueda de la información. (19)

1.9.2.1 Extensión al Trac

Un complemento/extensión, en inglés, extensión, es una aplicación que se relaciona con el objetivo de insertar nuevas características o servicios específicas a otra aplicación, aportándole de esta manera mayores funcionalidades. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la Interfaz de Programación de Aplicaciones, en inglés, Application Programming Interface (API). El gestor de proyecto Trac está orientado a extensiones desde entonces soporta la inserción de nuevas funcionalidades basadas en su arquitectura de componentes.

1.9.3 Entorno de Desarrollo Integrado de programación: Eclipse

Eclipse es un IDE de código abierto multiplataforma que incluye una extensión de sistema para ampliarla. Esta escrito principalmente en Java y se utiliza para desarrollar aplicaciones en este idioma y a través de los diversos extensión en otro lenguaje C/C++, Python, PHP, entre otros.

Eclipse fue desarrollado como el sucesor de su familia de herramientas para VisualAge, continuando su desarrollo por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

Publicado bajo los términos de la Licencia Pública de Eclipse, Eclipse es libre y software de código abierto. (20)

1.10 Conclusiones Parciales

Con el desarrollo de la anterior investigación se mostró la importancia de la gestión de los riesgos, evidenciando el alto valor que tiene la utilización de un modelo bien estructurado y detallado para administrar los riesgos en un proyecto, ya que de esta forma se contribuye directamente a evitar un posible fracaso que pudiera constituir una gran pérdida en el futuro.

Se realizó un profundo estudio del estado del arte de la disciplina a desarrollar, las herramientas a utilizar para modelar, los patrones de arquitectura, el lenguaje de programación y la metodología a tener en cuenta. Esto permitió la preparación teórica que sustenta el desarrollo del trabajo.

Capítulo 2: Característica, Diseño e Implementación del Sistema

Capítulo 2: Característica, Diseño e Implementación del Sistema

2.1 Introducción

En este capítulo se presenta el alcance de la aplicación, lo que permite tener una visión de cuál es el objetivo que se pretende alcanzar con su desarrollo, se muestran además, los requisitos funcionales y no funcionales, la interacción del autor con dichos requisitos, el Diagrama de Casos de Uso, de actividades, de clases y el de secuencia, permitiendo de esta forma cumplir con la metodología de desarrollo y la herramienta de modelado seleccionada como vía de mostrar el desarrollo de la aplicación que se desea construir.

2.2 Alcance de la aplicación

Una GR mal aplicada podría traer consigo retrasos de tareas o de entregas de trabajos según plazos acordados, encontrándose amenazados los planes diseñados por los líderes del proyecto. Por ellos se plantea la necesidad del estudio y la implementación de una aplicación que permita la GR, para la planificación, la identificación, el análisis, la solución y el seguimiento y control de los riesgos más significativos, siendo necesaria la integración de nuevas funcionalidades a la herramienta de GP, en este caso el Trac. Esta aplicación se va a poder utilizar en todos los proyectos del Polo, lo que traería consigo la implementación de una metodología continua innovadora en dicho proceso, con el propósito de desarrollar estrategias para tratar los riesgos de forma proactiva a lo largo del ciclo de vida del proyecto.

2.3 Estrategia definida.

Para realizar un proceso de gestión de riesgo continuo, exitoso y que logre todas las expectativas planificadas es necesario ser proactivos; lo que significa, que se debe de actuar de forma rápida y precisa para evitar que los riesgos no nos lleguen a afectar, o sea, adelantarnos a la acción de los riesgos sobre el proyecto. Aunque el propósito siempre debe estar enfocada a evitar el riesgo. Precisamente en la estrategia proactiva es que se basará la presente investigación, teniendo como caso de estudio el proyecto SCADA “Guardián Alba”.

El sistema que se desarrollará estará dirigido hacia el proceso de GR, incluyendo la gestión, monitorización y mitigación de riesgos, marcando las estrategias que debe seguir el equipo de trabajo frente a los riesgos que se enfrenta.

Capítulo 2: Característica, Diseño e Implementación del Sistema

2.4 Gestión de Riesgos

Si en el proyecto se planifican los recursos y el esfuerzo para reducir las probabilidades y el impacto de los riesgos, entonces se puede considerar que hay una GR, para que este proceso sea continuo, exitoso y que logre todas las expectativas planificadas es necesario ser proactivo, lo que significa, que se debe de efectuar de forma rápida y precisa para evitar que los riesgos no nos llegue a afectar, o sea, adelantarnos a la acción de los riesgos sobre el proyecto. Aunque el propósito siempre debe de estar dado a evitar los riesgos. A continuación se describe los diferentes pasos utilizados para la GR en el proyecto en estudio, Guardián del Alba.

2.4.1 Planificación de riesgos

Para desarrollar el plan de GR propuesto, se utilizó como metodología base el modelo del Project Management Institute (PMI), el cual tiene como objetivo garantizar que los riesgos del proyecto sean identificados, analizados, documentados, mitigados y controlados correctamente durante todo el ciclo de vida del proyecto. En esta Fase se plantea un proceso de planificación, pero se recomienda que el proyecto defina el momento y la forma en que el proceso de GR será iniciado y las circunstancias bajo las que se producirán las transiciones entre los pasos para cada riesgo.

La metodología a seguir figura de 4 etapas: identificación de riesgos, análisis, planeación de respuesta y seguimiento y control. En identificación se descubrirán los riesgos, antes de que se conviertan en problemas, en el análisis se clasificarán y priorizarán los riesgos. Posteriormente se planean las estrategias de respuesta al riesgo y se da seguimiento a aquellos riesgos potenciales que están en observación o mitigación.

2.4.2 Identificación de riesgos

El proceso de identificación de riesgos tiene como propósito identificar los riesgos que pueden afectar el proyecto potencialmente. La información a utilizar será, la información que nos puede brindar el polo sobre proyectos anteriores, el plan de riesgos de la fase anterior, el plan de gestión de proyectos (descripción del proyecto, el cronograma de actividades, recursos asignados, el proceso de estimación de tiempos, costos y presupuesto, las restricciones y demás) y el punto de vista del personal.

Entre las herramientas y técnicas se elaboró un cuestionario de identificación de riesgos basado en la técnica de la tormenta de ideas. Posteriormente se analizó la información disponible sobre toda la gestión del proyecto y la información de riesgos acerca de otros proyectos similares dentro del mismo polo, para hacer lo más extensa posible la lista de riesgos del proyecto.

Capítulo 2: Característica, Diseño e Implementación del Sistema

2.4.3 Análisis de los riesgos

El análisis de riesgos consiste en convertir los atributos del riesgo en información que sirva como base para tomar decisiones. El análisis de riesgos en el proyecto en estudio se organizó en dos pasos: análisis cualitativo de los riesgos, y clasificación y priorización de los riesgos.

2.4.3.1 Análisis cualitativo de los riesgos

En esta etapa se desarrolló una matriz de severidad de riesgo, en donde se identifican los riesgos usando la probabilidad de ocurrencia y el impacto en los objetivos del proyecto, en caso de que los riesgos llegaran a ocurrir.

Evaluación de los riesgos en la matriz de probabilidad e impacto del riesgo: La escala a utilizar y demás aspectos fueron descritos en el capítulo 1. En esta etapa se utiliza la lista de riesgos recolectada en la etapa de identificación anterior, con el propósito de clasificar los riesgos que más impactan al proyecto.

La forma en que el personal debe analizar los riesgos es tomando cada riesgo, perteneciente a la lista de riesgos recolectada en la etapa anterior; identificado los potenciales efectos adversos (consecuencia) y el criterio o información utilizado para la clasificación, valorando su probabilidad y luego su impacto. Posteriormente se multiplican estos dos factores y se calcula el valor de exposición, lo cual dará a conocer si el riesgo es alto, moderado o bajo. Finalmente se ordena la lista por el nivel de clasificación (alto a bajo), lo cual proporciona una base para asignar prioridades en el proceso de planificación.

Esta lista priorizada es un punto de partida del proceso de GR subsiguiente y por ello debe actualizarse a lo largo de todo el ciclo de análisis, planeamiento y supervisión de los riesgos.

Si se observa, de acuerdo a la probabilidad e impacto, a cada riesgo se le asigna un factor de exposición al riesgo. De acuerdo a su valor se clasifica en la matriz de impacto – probabilidad y se asigna una categoría, que puede ser riesgo bajo, alto o moderado.

2.4.4 Planeación de respuesta al riesgo

El principal objetivo de esta fase de planeación de respuesta al riesgo, consiste en desarrollar un plan detallado para controlar los riesgos más importantes identificados durante el análisis de riesgos.

Capítulo 2: Característica, Diseño e Implementación del Sistema

Durante el planeamiento de la acción de riesgos, el equipo debe tener en cuenta las cuatro alternativas: Mitigar, evitar, transferir y aceptar.

2.4.5 Seguimiento y control del riesgo

El seguimiento de riesgos supervisa el estado de los riesgos y el progreso de sus planes de acción. El seguimiento de riesgos incluye la supervisión de probabilidades, impactos, exposiciones y otras medidas de riesgo para los cambios que pudiesen alterar los planes de prioridades o de riesgos y las características, los recursos o la programación del proyecto.

Por otra parte, la fase de control tiene como objetivo corregir las desviaciones de los planes de mitigación, además de controlar los riesgos de la lista actual del proyecto. Por lo tanto se debe de estar atento a nuevos riesgos que puedan aparecer en su entorno a medida que el proyecto avanza.

Finalmente para una gestión eficaz de los riesgos, un proyecto debe tener una comunicación abierta ejercida de manera continua. Ésta puede ser tanto formal como informal. Se recomienda llevar a cabo las siguientes actividades: presentaciones y talleres de gestión del riesgo a los miembros del equipo, publicación de la lista de riesgos, informes periódicos del estado de los riesgos del proyecto a los jefes de líneas y al jefe del proyecto. Además todos estos documentos deberán ser parte integral del proceso de planificación.

2.5 Especificación de los requisitos de software.

Un requisito es la condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente (21).

Su principal objetivo es (22):

- Definir el ámbito del sistema.
- Definir una interfaz de usuarios para el sistema, enfocada a las necesidades y metas del usuario.
- Establecer y mantener un acuerdo entre clientes y otros involucrados sobre lo que el sistema debería hacer.

Capítulo 2: Característica, Diseño e Implementación del Sistema

- Proveer a los desarrolladores un mejor entendimiento de los requerimientos del sistema.
- Proveer una base para estimar recursos y tiempo de desarrollo del sistema.
- Proveer una base para la planeación de los contenidos técnicos de las iteraciones.

2.5.1 Requisitos Funcionales

Los Requisitos Funcionales (RF) son capacidades o condiciones que el sistema debe cumplir, características y restricciones sobre la funcionalidad del software. A continuación se muestra los RF identificados para la GR:

RF1 El sistema le debe permitir al jefe del proyecto Gestionar los riesgos.

RF1.1 El sistema le debe brindar la oportunidad de registrar los riesgos.

RF1.2 El sistema le debe ofrecer modificar los riesgos.

RF1.3 El sistema le debe aprobar eliminar los riesgos.

RF1.4 El sistema le debe proponer al jefe del proyecto tomar la opción de encuestar para definir los riesgos.

RF1.5 El Sistema debe calcular la prioridad de los riesgos identificados

RF2 El Sistema le debe permitir al jefe del proyecto introducir la respuesta a los riesgos que tengan mayor prioridad y mostrar dicho reporte.

RNF3 El sistema debe ofrecer la oportunidad de mostrar un reporte al Administrador de Proyecto.

2.5.2 Requisitos no Funcionales

Los Requerimientos no Funcionales (RNF) son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

Capítulo 2: Característica, Diseño e Implementación del Sistema

En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requerimientos funcionales, es decir una vez que se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. (22)

A continuación se muestra los RNF identificados para la GR.

Usabilidad

RNFUS1 El extensión luego de instalado deberá visualizarse con calidad en los principales navegadores existentes (Internet Explorer, Opera, Firefox y otros).

Fiabilidad

RNFF1 El sistema debe de ser capaz de mantener la calidad de los datos de manera que garantice su integridad durante su aplicación, procesamiento y almacenamiento en la Base de Datos.

RNFF2 Cada vez que se detecte un cambio en la Base de Datos se debe replicar de manera automática los cambios.

Soportabilidad

El sistema debe ser:

RNFSO1 De fácil instalación, configuración y puesta en marcha.

RNFSO2 De arquitectura abierta y distribuida, modular, de capacidad escalable.

RNFSO3 Programado orientado a objeto.

RNFSO4 El sistema debe correr sobre un servidor Apache 2.0 y utilizar SQLite como sistema gestor de base de datos.

Capítulo 2: Característica, Diseño e Implementación del Sistema

Interfaz de Usuario

RNFIU1 El diseño del sistema debe estar enfocado en la misma interfaz del trac, debe ser sencillo y funcional, fácil de navegar y de rápida adaptación para los usuarios.

Confiabilidad

RNF1 El sistema debe ser capaz de mantener la calidad del dato de manera que garantice su integridad durante su aplicación, procesamiento y almacenamiento en la Base de Datos.

RNF2 Cada vez que se detecte un cambio en la Base de Datos se debe replicar de manera automática los cambios.

Desempeño

RNF3 La base de datos debe permitir ser accedida, modificada, actualizada independientemente del número de datos que maneje.

Implementación

RNF4 El sistema debe ejecutarse en diversas plataformas de hardware y software.

RNF5 El sistema se realizará usando el lenguaje de programación Python, para la realización de la interfaz gráfica se utilizara la biblioteca Genshi y como herramienta para programación en Eclipse. Para la modelación de los diagramas UML se utilizará el NetBeans y como metodología de desarrollo de software el OpenUp.

RNF6 El sistema debe cumplir con los lineamientos necesarios para la producción de software libre y la comunidad de desarrollo y soporte.

Rendimiento

RNF7 Los tiempos de respuesta del sistema serán de corta duración así como la evaluación que se efectuará.

Capítulo 2: Característica, Diseño e Implementación del Sistema

Portabilidad

RNFP1 El sistema debe ser multiplataforma, debe ejecutarse de manera óptima sin importar el sistema operativo que utilice el usuario.

2.6 Diagrama de Casos de Usos.

Un caso de uso (CU) es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. Cada CU proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico. El CU que nos compete desarrollar aborda las actividades a desarrollar para que un equipo de trabajo lleve a cabo la GR en un proyecto de desarrollo de software.



Figura 3: Diagrama de Caso de Uso del Sistema

Capítulo 2: Característica, Diseño e Implementación del Sistema

2.7 Descripción de los Casos de Uso

Caso de Uso: Gestionar Riesgos

1. Breve descripción

Este CU le permite al Administrador de Proyecto realizar el proceso de GR, identificando riesgos, caracterizándolos, analizándolos, dándole una respuesta de solución y seguimiento a los mismos.

2. Actores:

Administrador de Proyecto “Activa”

3. Precondiciones:

El Administrador de Proyecto debe garantizar la completitud de los datos suministrados.

El Administrador de Proyecto debe asegurarse de la correcta introducción de los campos requeridos.

Flujo Básico de Eventos

4.1 Adicionar riesgos del proyecto

4.1.1 El Administrador de Proyecto selecciona en el panel de administración la opción Administrar de Riesgos.

4.1.2 El sistema le muestra una serie de opciones para que el Administrador de Proyecto seleccione la acción que desee realizar, pudiendo insertar, modificar (Alternativo 5.1), Analizar los riesgos críticos (Alternativo 5.3) y eliminar (Alternativo 5.2) un riesgo.

4.1.3 El Administrador de Proyecto solicita insertar un nuevo riesgo en el sistema.

4.1.4 El Sistema muestra un formulario brindando la posibilidad de insertar un nuevo riesgo. Los campos a completar son:

Nombre: nombre del nuevo riesgo. (obligatorio)

Capítulo 2: Característica, Diseño e Implementación del Sistema

Clasificación del riesgo: La clasificación de los riesgos se puede hacer de cuatro formas diferentes; por fases del modelo del ciclo de desarrollo del proyecto, por tipos de procesos a usar, por tipos de productos a utilizar, por tipo de riesgo (contractuales, de presupuesto/costo, de cronograma, de recurso y de soporte). (obligatorio)

Descripción: Se explica en qué consiste el riesgo. (obligatorio)

Consecuencias del riesgo: Las consecuencias que traería sobre el proyecto la ocurrencia de dicho riesgo. (obligatorio)

Impacto: Es el efecto que tendría el riesgo sobre los objetivos del proyecto en el caso de presentarse. (obligatorio)

Probabilidad: La posibilidad de que el riesgo ocurra. (obligatorio)

Exposición: Es la prioridad asignada a un riesgo, se obtiene a través de la combinación de la probabilidad y el impacto. (automático)

4.1.5 El Administrador de Proyecto ingresa los datos solicitados. Alterno (5.5.1)[T]

4.1.6 El sistema le muestra una interfaz brindando la posibilidad de seguir adicionando riesgos y a la vez le da la oportunidad al Administrador de Proyecto de modificar (Alterno 5.1), Analizar los riesgos críticos (Alterno 5.3) o eliminar (Alterno 5.2) un riesgo.

4.2 Fin del Caso de Uso

4.2.1 El Administrador de Proyecto solicita al sistema cerrar la aplicación.

4.2.2 El sistema cierra la aplicación.

4.3 Guardar Información

4.3.1 El sistema ofrece la posibilidad de guardar los datos.

4.3.2 El Administrador de Proyecto guarda la información.

Capítulo 2: Característica, Diseño e Implementación del Sistema

5. Flujos Alternos

5.1 Modificar riesgos del proyecto

5.1.1 En el paso 4.1.1, 4.1.5 del flujo básico o 5.3.5 del flujo alternativo el Administrador de Proyecto selecciona la opción modificar riesgos.

5.1.2 El sistema le muestra al Administrador de Proyecto los riesgos que fueron identificados, dándole la posibilidad de seleccionar el riesgo a modificar.

5.1.3 El sistema muestra un formulario con los datos del riesgo seleccionado, dando la posibilidad de cambiar el campo que se quiere modificar. [T]

5.1.4 El sistema muestra un reporte con las modificaciones realizadas, pudiendo el Administrador de Proyecto seguir modificando riesgos, y además le da la posibilidad de insertar (básico 4.1), Analizar los riesgos críticos (Alternativo 5.3) y eliminar (Alternativo 5.2) un riesgo.

5.2 Eliminar riesgos del proyecto.

5.1.1 En el paso 4.1.1, 4.1.5 del flujo básico o 5.1.4 del flujo alternativo el Administrador de Proyecto selecciona la opción mostrar reporte de todos los riesgos introducidos en el sistema.

5.2.2 El sistema muestra dicho reporte.

5.2.3 El Administrador de Proyecto selecciona los riesgos que desee eliminar. [T]

5.2.4 El sistema actualiza la base de datos.

5.3 Analizar riesgos críticos del proyecto

5.3.1 En el paso 4.1.1 y 4.1.5 del flujo básico, o 5.1.4 y 5.3.5 del flujo alternativo el Administrador de Proyecto solicita al sistema elaborar una respuesta a los riesgos más críticos previamente definidos.

5.3.2 El sistema le muestra un listado con los riesgos más críticos que no han sido mitigados aún y le brinda la posibilidad al Administrador de Proyecto de seleccionar el riesgo a analizar.

Capítulo 2: Característica, Diseño e Implementación del Sistema

5.3.3 El sistema muestra un formulario para insertar la respuesta al riesgo seleccionado por el Administrador de Proyecto.

5.3.4 El Administrador de Proyecto introduce la respuesta al riesgo seleccionado. Alternativo (5.6) [T]

5.3.5 El sistema muestra la posibilidad al Administrador de Proyecto de seguir escogiendo riesgos para su análisis y además le brinda la opción de insertar (básico 4.1), modificar (Alternativo 5.1) o eliminar (Alternativo 5.2) un riesgo.

5.4 Encuestar para Identificar Riesgos

5.4.1 En el paso 4.3.1 del flujo básico, el Administrador de Proyecto en vez de seleccionar “Adicionar”, selecciona la opción “Encuestar”, con el objetivo de utilizar un encuestador que a partir de las características de su proyecto le muestre algunos riesgos de del mismo.

5.4.2 El sistema, una vez realizada la encuesta le muestra al Administrador de Proyecto el listado de algunos riesgos de su proyecto, permitiéndole adicionar nuevos riesgos, en caso de que considere conveniente; posteriormente retorna al paso 4.3.2 del flujo básico.

5.5 Error al insertar un riesgo.

5.5.1 En el paso 4.1.5 del flujo básico el sistema verifica que existe información incompleta, mostrándole un mensaje de Error al Administrador de Proyecto.

5.5.2 El Administrador de Proyecto acepta el mensaje de error.

5.5.3 El sistema retorna al paso 4.1.4 del flujo básico.

5.6 Error al insertar respuesta a un riesgo.

5.6.1 En el paso 4.4.4 del flujo básico el sistema verifica que existe información incompleta, mostrándole un mensaje de Error al Administrador de Proyecto.

5.6.2 El Administrador de Proyecto acepta el mensaje de error.

5.6.3 El sistema retorna al paso 4.4.3 del flujo básico.

Capítulo 2: Característica, Diseño e Implementación del Sistema

2.8 Arquitectura.

La descripción de la arquitectura es una vista de los modelos del sistema, describe las partes del sistema que es importante que comprendan todos los desarrolladores y otros interesados. Una arquitectura debe ser completa pero también debe ser lo suficientemente flexible. Para el desarrollo de esta se hará uso de la arquitectura sobre la cual está desarrollado el trac.

La arquitectura que se propone, persigue que la extensión de GR pueda dar respuestas a los requisitos de comprensibilidad, extensibilidad y eficiencia, logrando la correcta integración con la herramienta trac. Por tanto, se utilizara la arquitectura del trac que está basada en componentes, los que proporcionan varios tipos de servicios en el contexto de la aplicación.

El núcleo del trac posee un componente central al cual van a estar conectado otros componentes mediante un meta-extensión-API, este componente central, a su vez, mediante las API posee varios puntos de extensión, cuya misión sería facilitar la conexión de otras interfaces o extensión a este, como se describe en la figura 4.

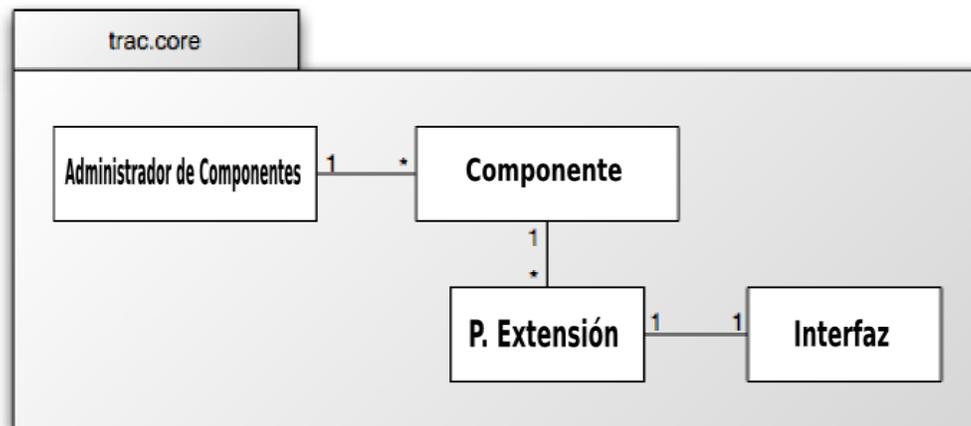


Figura 4: Arquitectura del trac basada en componentes, núcleo del trac.

De la misma forma dos interfaces conectados a un mismo componente pueden estar relacionados entre sí, facilitando esta relación el intercambio de información como se muestra en la figura 5.

Capítulo 2: Característica, Diseño e Implementación del Sistema

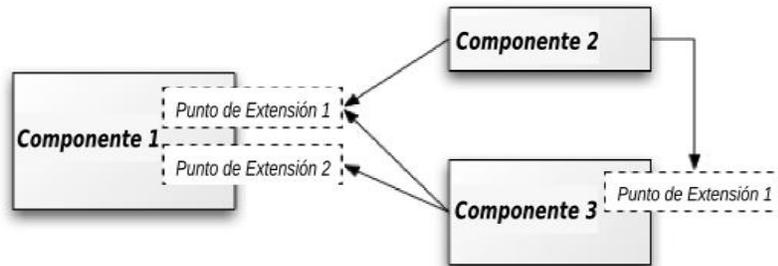


Figura 5: Arquitectura del trac basada en componentes, puntos de extensión.

Para el diseño de la aplicación se utilizaron varios patrones como son: Singleton y Polimorfismo.

- **Singleton:** Este patrón garantiza que en cada momento solo exista una instancia de cada clase ejecutándose. El subsistema manejador de componentes aplicando este patrón garantiza que solo se está ejecutando una sola instancia de cada componente en el sistema.
- **Polimorfismo:** Este patrón garantiza que la clase hija hereda todos los atributos y métodos que contiene la clase padre, solo tiene que re-implementar las funcionalidades que ella necesite de la clase padre. El uso de este patrón traerá consigo numerosas ventajas.

2.9 Modelo de Diseño.

La extensión a desarrollar estará basado en la arquitectura base del Trac, la herramienta para el trabajo con los datos históricos que se desarrollará en la investigación quedaría enmarcado dentro del subsistema Componente. El trac está compuesto por un núcleo, dentro del núcleo tenemos varios subsistemas como son Administrador de Componentes, Componente, Puntos de Extensión e Interfaces. El subsistema Administrador de Componentes está formado por uno o varios Componentes, este subsistema es el encargado de que en todo momento exista solo una instancia de cada componente en el sistema. Aplicando así el patrón Singleton. El subsistema Componente no es más que un objeto que proporciona un determinado tipo de servicio. También existen los Puntos de Extensión que son como su nombre lo indica puntos de extensión que se le hace a los componente permitiendo crear interfaces de nuestro componente que pueda ser implementado por otros componentes.

Capítulo 2: Característica, Diseño e Implementación del Sistema

Dentro del diseño de la aplicación se aplicaron los siguientes patrones: Singleton dentro de la arquitectura del trac.

El Diseño del sistema está enfocado en la misma interfaz del Trac, bajo las siguientes características: sencillo y funcional, fácil de navegar y de rápida adaptación para los usuarios. Está conformado por tres subsistemas: “Gestionar Riesgos”, “Analizar Riesgos” y “Mostrar Reporte de los Riesgos Identificados”:

1. Gestionar Riesgos.

Es el primer subsistema del diseño, brinda la funcionalidad de adicionar, modificar, eliminar y caracterizar los diferentes riesgos identificados en el proyecto, para darle cumplimiento a esta funcionalidad se deben llenar los siguientes campos de información sobre los riesgos: nombre, clasificación, consecuencias, descripción, impacto y probabilidad.

2. Analizar Riesgos.

Es el segundo subsistema del diseño, brinda la funcionalidad de darle una respuesta a los riesgos más críticos identificados en el proyecto. Para darle cumplimiento a esta funcionalidad se deben llenar los siguientes campos de información sobre los riesgos: mitigación, contingencia, activadores, responsables y resultado de la estrategia de manejo de riesgos.

3. Mostrar Reporte de los Riesgos Identificados.

Es el tercer y último subsistema del diseño, se encarga de mostrar cada vez que se solicite una tabla con la información que se ha trabajado sobre los riesgos, a través de los anteriores subsistemas.

2.9.1 Diagrama de clases

El diagrama de clases que se muestra en la figura 6 se describe la estructura del sistema propuesto mostrando sus clases, atributos y las relaciones entre ellos.

Capítulo 2: Característica, Diseño e Implementación del Sistema

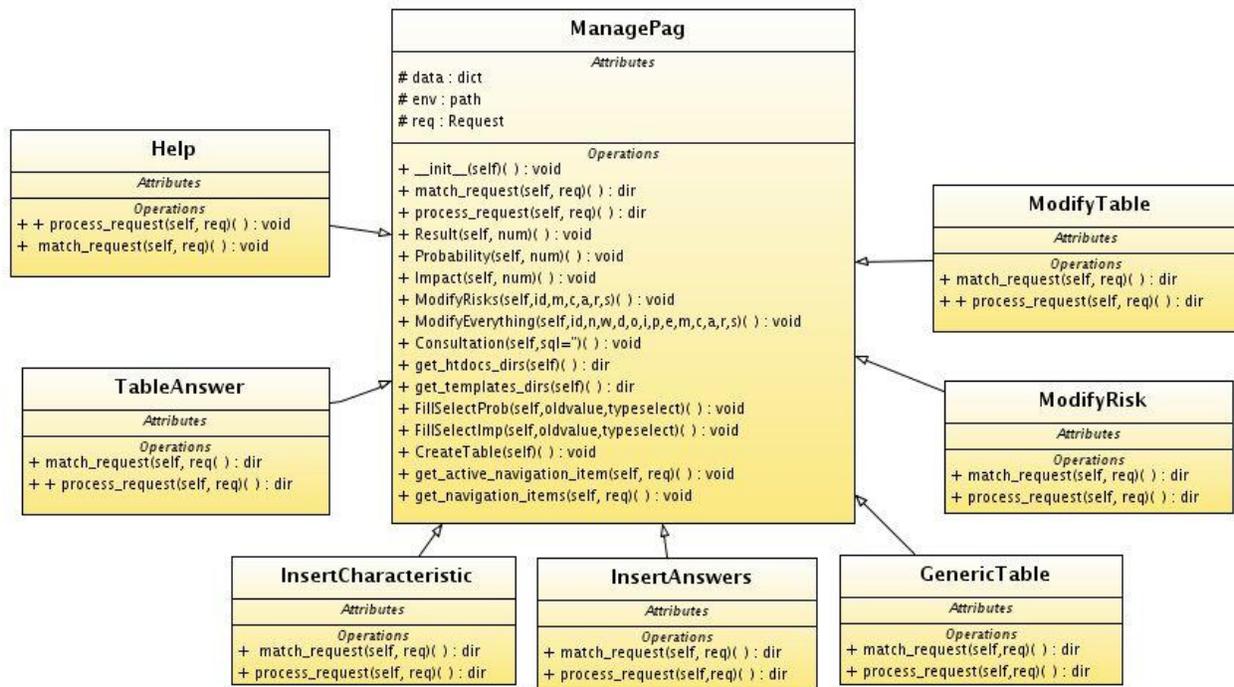


Figura 6: Diagrama de Clases del Sistema

El modelo describe un sistema basado en componentes centrándose específicamente en los componentes y las interfaces. Este sistema es el que permite agregarle componentes o extensiones al Trac a través de unos meta-extensión-API que proporcionan un número de puntos de extensión que facilitan la conexión de nuevos componentes.

2.9.2 Descripción de las clases

A continuación se realizará una descripción minuciosa y detallada de las clases que componen el sistema así como funcionalidad de los métodos que la componen.

Capítulo 2: Característica, Diseño e Implementación del Sistema

Tabla 4: Descripción de la clase ManagePag.

Nombre: ManagePag	
Descripción: Es la Clase padre encargada de administrar los riesgos en donde es implementado los métodos más importantes que posteriormente se redefinen en las clases hijas o simplemente se llama.	
Tipo de clase: Controladora	
Atributo	Tipo
data	dict
env	path
Para cada responsabilidad:	
Nombre:	<code>__init__(self)</code>
Descripción:	Constructor de la clase
Nombre:	<code>match_request(self, req)</code>
Descripción:	Se encarga de capturar la petición del usuario y retornar la dirección donde se encuentra dicha petición.
Nombre:	<code>process_request(self, req)</code>
Descripción:	Se encarga de procesar el pedido de modificar los campos.
Nombre:	<code>Result(self, num)</code>
Descripción:	Devuelve la cuantificación o la prioridad del riesgo, si es crítico o no.
Nombre:	<code>Probability(self, num)</code>
Descripción:	Devuelve la probabilidad de que el riesgo ocurra.
Nombre:	<code>Impact(self, num)</code>
Descripción:	Devuelve el efecto que tendría el riesgo sobre los objetivos del proyecto en caso de presentarse.
Nombre:	<code>ModifyRisks (self,id,m,c,a,r,s)</code>
Descripción:	Se encarga de insertar la respuesta a los riesgos priorizados.
Nombre:	<code>ModifyEverything (self,id,n,w,d,o,i,p,e,m,c,a,r,s)</code>
Descripción:	Se encarga de hacer modificaciones en caso de inconformidad o de error de un riesgo determinado.
Nombre:	<code>Consultation (self,sql)</code>

Capítulo 2: Característica, Diseño e Implementación del Sistema

Descripción:	Se encarga de realizar todas las consultas a la Base de Datos.
Nombre:	FillSelectProb (self,oldvalue,typeselect)
Descripción:	Se encarga de mostrar en un select el valor que tiene el campo probabilidad.
Nombre:	FillSelectImpact (self,oldvalue,typeselect)
Descripción:	Se encarga de mostrar en un select el valor que tiene el campo impacto.
Nombre	CreateTable (self)
Descripción	Crea la tabla en la base de datos si no ha sido creada.
Nombre	get_active_navigation_item (self, req)
Descripción	Método que devuelve la dirección de la pagina actual
Nombre	get_navigation_items (self, req):
Descripción	Se encarga de crear el botón de la navegación del extensión

Capítulo 2: Característica, Diseño e Implementación del Sistema

Tabla 5: Descripción de la clase InsertCharacteristic.

Nombre: InsertCharacteristic	
Descripción: Clase hija de ManagePag, es la encargada de insertar nuevos riesgos.	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	match_request(self,req)
Descripción:	Se encarga de capturar el pedido de modificar la información y retornar la dirección donde se encuentra el template solicitado
Nombre:	process_request(self,req)
Descripción:	Se encarga de procesar el pedido de modificar los campos.

Capítulo 2: Característica, Diseño e Implementación del Sistema

Tabla 6: Descripción de la clase InsertAnswers:

Nombre: InsertAnswers	
Descripción: Clase hija de ManagePag, es la encargada de insertar la respuesta o la solución de un riesgo dado.	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	<code>match_request(self,req)</code>
Descripción:	Se encarga de capturar el pedido de modificar la información y retornar la dirección donde se encuentra el template solicitado
Nombre:	<code>process_request(self,req)</code>
Descripción:	Se encarga de procesar el pedido de modificar los campos.

Capítulo 2: Característica, Diseño e Implementación del Sistema

Tabla 7: Descripción de la clase GenericTable

Nombre: GenericTable	
Descripción: Clase hija de ManagePag, es la encargada de mostrar un reporte de todos los riesgos insertados en el sistema, además de habilitar la opción de eliminar riesgos.	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	<code>match_request(self,req)</code>
Descripción:	Se encarga de capturar el pedido de modificar la información y retornar la dirección donde se encuentra el template solicitado
Nombre:	<code>process_request(self,req)</code>
Descripción:	Se encarga de procesar el pedido de modificar los campos.

Capítulo 2: Característica, Diseño e Implementación del Sistema

Tabla 8: Descripción de la clase ModifyRisk.

Nombre: ModifyRisk	
Descripción: Clase hija de ManagePag, es la encargada de modificar en caso de inconformidad un riesgo dado.	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	<code>match_request(self, req)</code>
Descripción:	Se encarga de capturar el pedido de modificar la información y retornar la dirección donde se encuentra el template solicitado
Nombre:	<code>process_request(self, req)</code>
Descripción:	Se encarga de procesar el pedido de modificar los campos.

Capítulo 2: Característica, Diseño e Implementación del Sistema

Tabla 9: Descripción de la clase ModifyTable.

Nombre: ModifyTable	
Descripción: Clase hija de ManagePag, es la encargada de mostrar un reporte de todos los riesgos del sistema además de brindar la opción de escoger un riesgo para ser modificado posteriormente.	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	<code>match_request(self,req)</code>
Descripción:	Se encarga de capturar el pedido de modificar la información y retornar la dirección donde se encuentra el template solicitado
Nombre:	<code>process_request(self,req)</code>
Descripción:	Se encarga de procesar el pedido de modificar los campos.

Capítulo 2: Característica, Diseño e Implementación del Sistema

Tabla 10: Descripción de la clase TableAnswer.

Nombre: TableAnswer	
Descripción: Clase hija de ManagePag, es la encargada mostrar un reporte de todos los riesgos del sistema además de brindar la opción de escoger un riesgo para insertarle posteriormente una respuesta de solución.	
Tipo de clase:	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	<code>match_request(self,req)</code>
Descripción:	Se encarga de capturar el pedido de modificar la información y retornar la dirección donde se encuentra el template solicitado
Nombre:	<code>process_request(self,req)</code>
Descripción:	Se encarga de procesar el pedido de modificar los campos.

Capítulo 2: Característica, Diseño e Implementación del Sistema

Tabla 11: Descripción de la clase Help.

Nombre: Help	
Descripción: Clase hija de ManagePag, es la encargada de mostrar un manual de usuario para que sea más fácil a la hora de trabajar con la aplicación.	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	<code>match_request(self,req)</code>
Descripción:	Se encarga de capturar el pedido de modificar la información y retornar la dirección donde se encuentra el template solicitado
Nombre:	<code>process_request(self,req)</code>
Descripción:	Se encarga de procesar el pedido de modificar los campos.

2.9.3 Realizaciones de Caso de Uso

El sistema está formado por tres casos de uso (CU). El CU Gestionar Riesgos es el encargado de las funcionalidades de insertar, modificar y eliminar riesgos, el CU Caracterizar Riesgos, se encarga de determinar las consecuencias, el impacto, la probabilidad, la extensión, las descripción y su clasificación de todos los riesgos identificados y el CU Elaborar Estrategia de Manejo de los Riesgos permite registrar la respuesta o solución de los riesgo más significativos que hayan sido identificados.

CU #1: Caso de Uso Gestionar Riesgos.

Este CU está formado por tres escenarios básicos.

Capítulo 2: Característica, Diseño e Implementación del Sistema

Vista de los participantes

Según el diseño de clases mostrado en la figura 6 las clases que realizan este CU son la clase padre "ManagePag"; y sus hijas "InsertCharacteristic" encargada de todo el proceso de adicionar riesgos, "ModifyTable", clase encargada de mostrar los riesgos con la opción de escoger uno que posteriormente va a ser modificado; "ModifyRisk" Clase encargada de modificar determinados datos sobre los Riesgos, "GenericTable", clase encargada de mostrar un reporte con la opción de eliminar uno o varios riesgos.

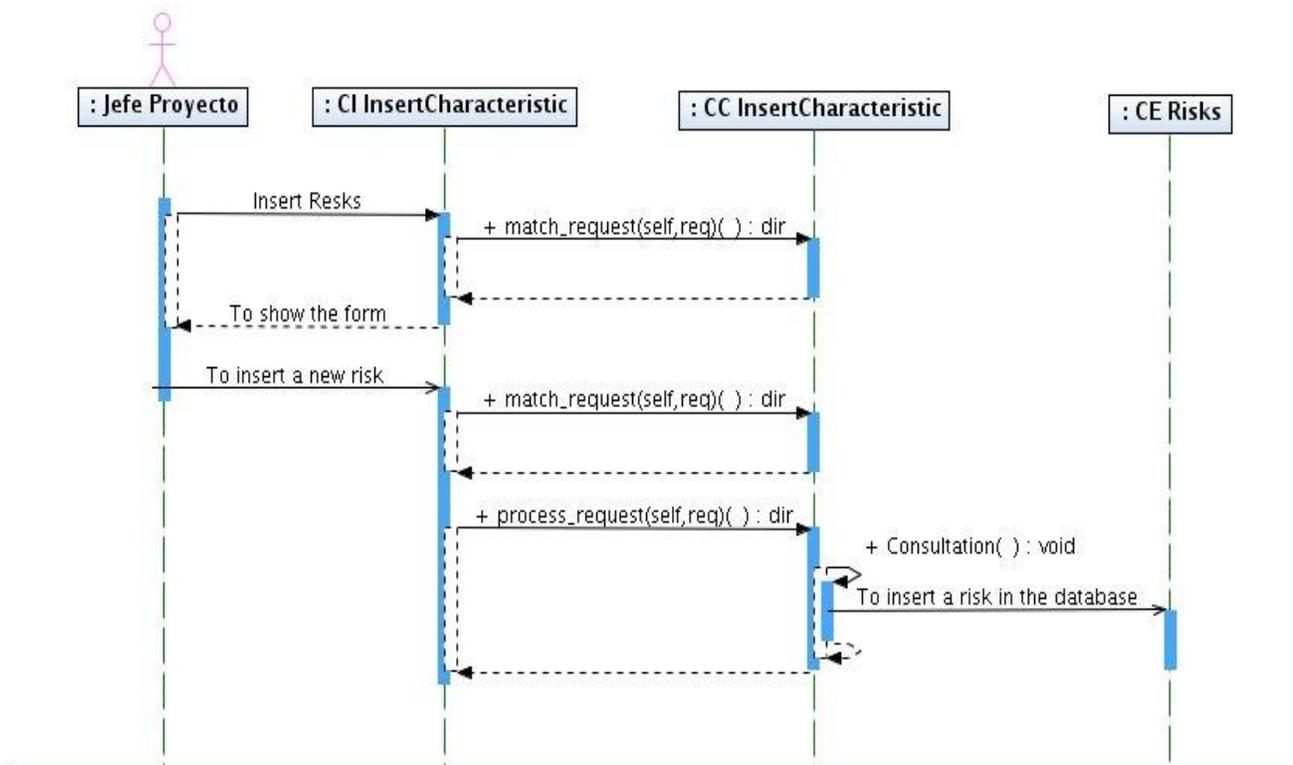


Figura 7: Diagrama de Secuencia Flujo Principal "Insertar Riesgos".

Capítulo 2: Característica, Diseño e Implementación del Sistema

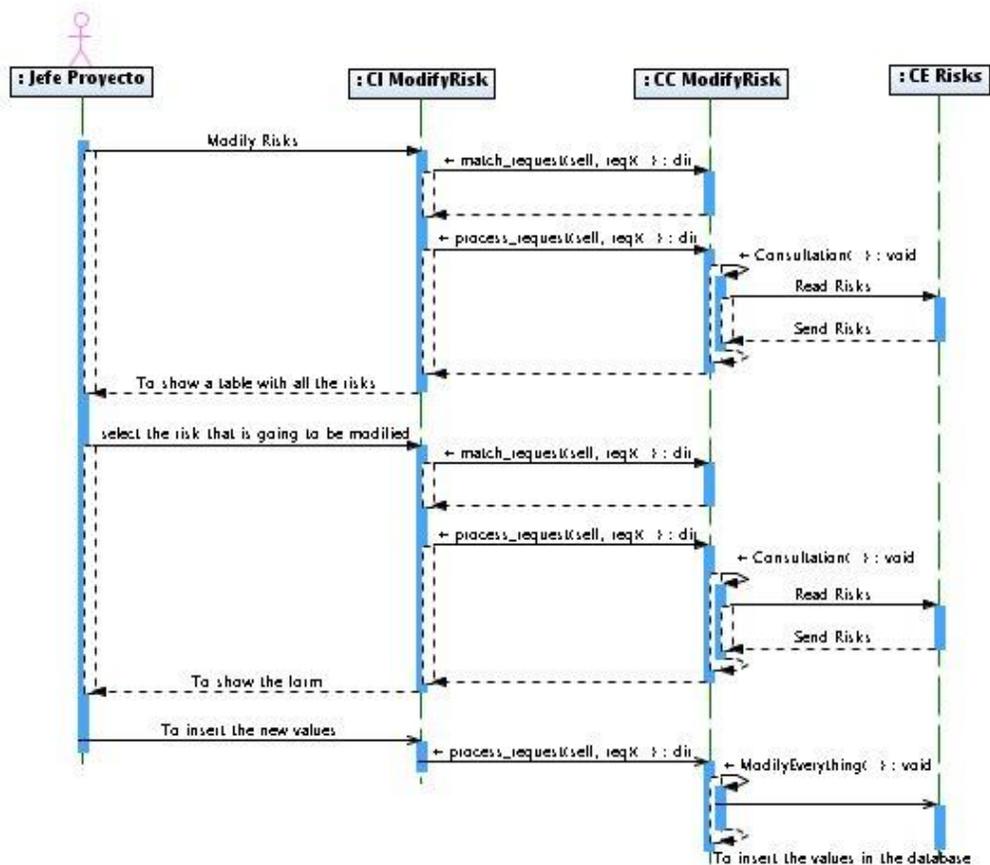


Figura 8: Diagrama de Secuencia Flujo Alterno "Modificar Riesgos".

Capítulo 2: Característica, Diseño e Implementación del Sistema

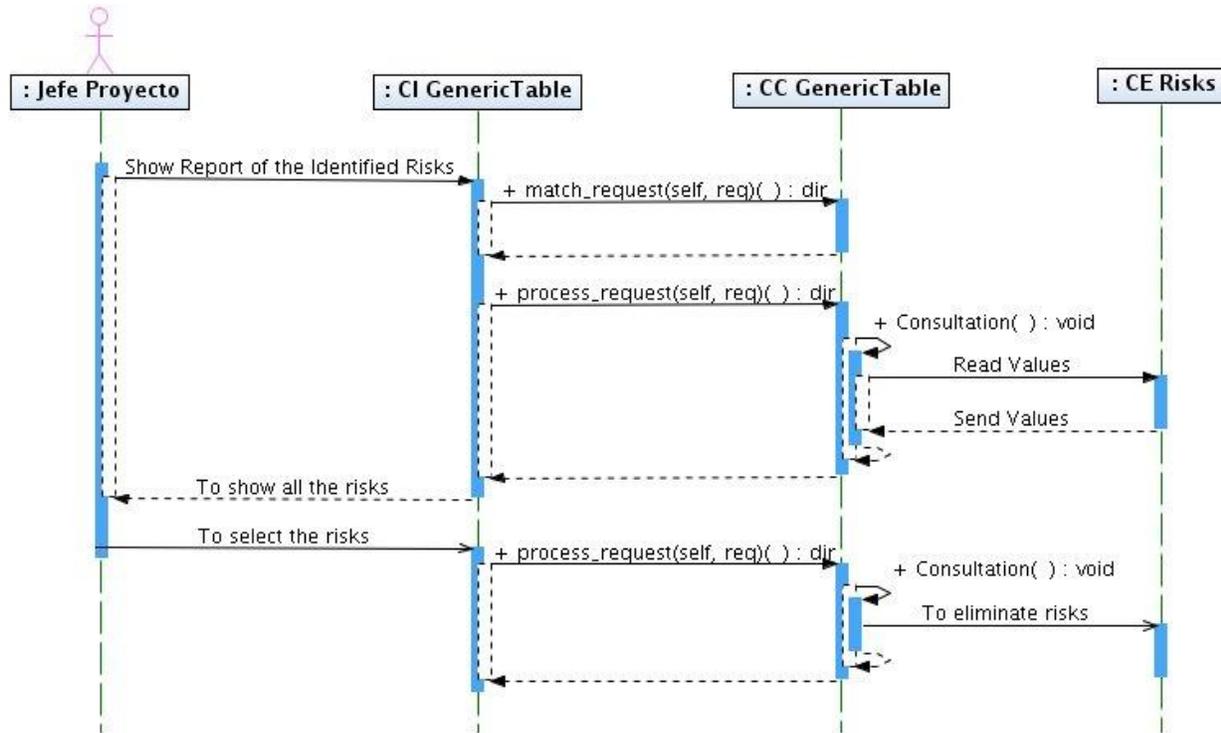


Figura 9: Diagrama de Secuencia Flujo Alterno “Eliminar Riesgos”.

Capítulo 2: Característica, Diseño e Implementación del Sistema

CU #2: Caso de Uso Elaborar Estrategia de Manejo de los Riesgos.

Estos CU están formados por un escenario básico.

Vista de los participantes

Según el diseño de clases mostrado las clases que realizan este CU son: la clase padre “ManagePag”; y las clases hijas “***InsertAnswers***”; para definir y guardar la respuesta a los riesgos más críticos y “***TableAnswer***”, clase encargada de mostrar los riesgos más críticos para darle una respuesta posteriormente.

Capítulo 2: Característica, Diseño e Implementación del Sistema

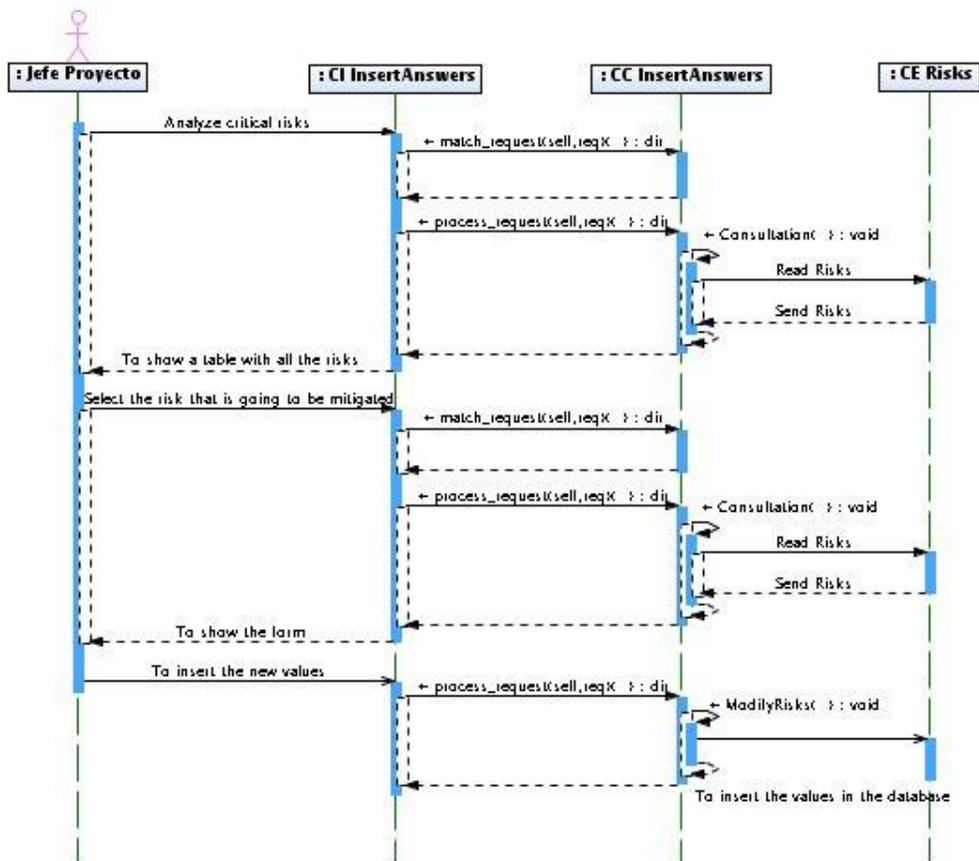


Figura 10: Diagrama de Secuencia Flujo Principal “Elaborar Estrategia de Manejo de los Riesgos”.

Capítulo 2: Característica, Diseño e Implementación del Sistema

2.10 Implementación.

La aplicación implementada debe estar bajo las siguientes dependencias:

- Disponibilidad del Sistema Operativo Libre bajo la modalidad de Código Abierto cuyas características cumplan con las requeridas para el correcto funcionamiento del Trac.
- Disponibilidad de que la Base de Datos del Trac, garantizando que la data requerida para la aplicación proporcione resultados válidos acordes a los esperados, y que la información sea actualizable y mantengan su integridad.

La descripción de esta implementación se puede apreciar mediante el diagrama de despliegue visualizado en la figura 7; una PC Cliente conectada mediante el protocolo HTTP al Server Apache sobre el que debe correr utilizando SQLite como Sistema Gestor de Base de Datos.

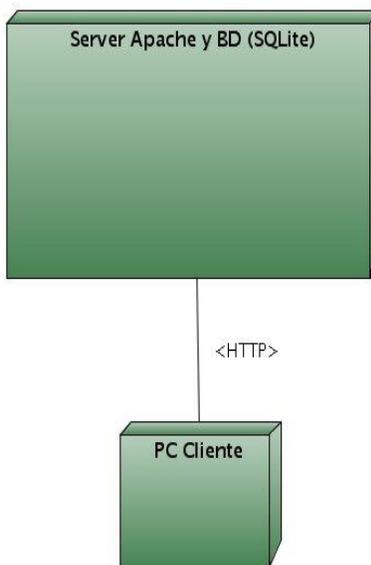


Figura 11: Diagrama de Despliegue

Capítulo 2: Característica, Diseño e Implementación del Sistema

2.11 Conclusiones Parciales

Este capítulo provee una visión general del proceso de GR diseñado y utilizado en gran parte, en un proyecto pequeño de desarrollo de software en plazos de tiempo cortos.

Se aplicaron algunas técnicas para la identificación, análisis y mitigación de los riesgos que amenazan a los proyectos del polo. Se tuvo en cuenta además, el análisis de la herramienta para la GP possibilitando diseñar e implementar una extensión a dicha herramienta, respetando la interfaz y arquitectura del sistema.

Capítulo 3: Prueba y Seguimiento de la Solución Planteada

Capítulo 3: Prueba y Seguimiento de la Solución Planteada

3.1 Introducción

El proceso de pruebas se enfoca sobre la lógica interna del software y las funciones externas. Es un proceso de ejecución de un programa con la intención de descubrir un error. Un buen caso de prueba es aquel que tiene alta probabilidad de mostrar un error no descubierto hasta entonces.

Las pruebas de software son un proceso centrado en el objetivo de encontrar defectos a un software; puede ser por razones de depuración al software o de aceptación del mismo. Tratar de encontrar defectos es una parte esencial en todo proceso de pruebas, para ellos los probadores evalúan el producto desde un punto de vista crítico sometándolo a una serie de acciones inquisitivas y el producto responde con su comportamiento como reacción.

Los Métodos de Prueba de Software tienen el objetivo de diseñar pruebas que descubran diferentes tipos de errores con menor tiempo y esfuerzo.

La prueba del software constituye un elemento crítico para la garantía de la calidad del software. Los dos métodos de prueba más utilizados son el Método de Prueba de Caja Blanca y el Método de Prueba de Caja Negra. La diferencia entre ambos consiste en que con la Caja Blanca es necesario conocer el código y estar bastante relacionado con él para saber exactamente cuál es la lógica interna de lo que se va a probar, sin embargo en la Caja Negra solo basta con conocer las posibles entradas y salidas del programa.

3.2 Plan de Pruebas.

Para la correcta realización de las pruebas, se contó con una serie de recursos que le facilitaron el trabajo de ejecutar cada caso de prueba sobre el sistema de GR.

- **Recurso Físico**

Las computadoras utilizadas para el desarrollo de las pruebas contaron con 1.0 giga de memoria RAM, microprocesador Intel Core2Duo E4500 con velocidad de 2.20 Ghz, motherboard Intel y una capacidad en disco duro de 160 gigas. Red alámbrica.

Capítulo 3: Prueba y Seguimiento de la Solución Planteada

- **Recurso Lógico**

El sistema operativo en el cual se desarrollaron las pruebas fue Debian 5.0 (lenny) Kernel Linux 2.6.26-1-686, GNOME 2.22.3.

3.3 Diseño de clases de pruebas.

Caso de Uso: Gestionar Riesgos

Este CU permite al Administrador de Proyecto realizar la actividad de adicionar nuevos riesgos en el sistema, también le brinda la opción de poder modificar y eliminar dichos riesgos.

Las pruebas a realizar en este caso de uso son:

- Insertar Riesgos.
- Modificar Riesgos.
- Eliminar Riesgos.

CP #1: Insertar Riesgos.

Breve Descripción

Este caso de prueba permite comprobar la funcionalidad de insertar un nuevo riesgo en un cuestionario.

Flujo Central

1. El jefe del proyecto accede a la aplicación.
2. El sistema muestra en una interfaz todas las operaciones que el Administrador de Proyecto puede realizar.
3. El jefe del proyecto selecciona la opción Insert Risk.
4. El Sistema presentará un formulario con los campos respectivos que permite insertar un nuevo riesgo, los campos a llenar son: Nombre, Clasificación, Descripción, Consecuencia, Impacto y Probabilidad.

Capítulo 3: Prueba y Seguimiento de la Solución Planteada

5. El Administrador de Proyecto introduce correctamente los datos del riesgo que desea insertar.
6. El sistema guarda los datos en la base de datos.
7. El jefe del proyecto sale de la aplicación.

2.4 Iteraciones

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
El jefe del proyecto introduce todos los campos correctos en el sistema: (“Name”: Entrega tardía de los trabajos, “Classification”: riesgos de proyecto, “Description”: Se entregan los trabajos fuera del tiempo acordado, “Concequencias”: La entrega de trabajo fuera de tiempo atrasa los planes trazados, “Impact”: medio, “Probability”: 0.5).		El sistema crea un nuevo riesgo introduciendo todos los datos de forma satisfactoria, brindándole la posibilidad de adicionar más riesgos.	El Sistema introduce los datos del riesgo de forma exitosa y los muestra satisfactoriamente.	
	El jefe del proyecto introduce los datos dejando campos en blanco (“Name”: Entrega tardía de los trabajos, “Classification”: riesgos de proyecto”, “Description”: “Concequencias”: “Impact”: medio, “Probability”: 0.5).	El Sistema muestra mensajes de error alertando al usuario de que existen campos vacios.	El Sistema no introduce la información y lanza un error por existir campos en blanco.	

Capítulo 3: Prueba y Seguimiento de la Solución Planteada

	El Administrador de Proyecto introduce datos números en uno de los campo: (“Name”: Entrega tardía de los trabajos, “Classification”: riesgos de proyecto, “Description”: Se entregan los trabajos fuera del tiempo acordado, “Consecuencias”: 5, “Impact”: medio, “Probability”: 0.5)	El Sistema muestra un mensaje de error alertando que existen datos incorrectos en algunos de los campos.	El Sistema no introduce los datos por existir campos incorrectos lanzando así un error.	
--	---	--	---	--

CP #2: Modificar Riesgos

Breve Descripción

Este caso de prueba nos permite poder modificar un riesgo que ya este registrado en el sistema a través de un cuestionario.

Las pruebas realizadas a este escenario son las siguientes

- Modificar el riesgo que haya sido seleccionado.

Flujo Central

1. El jefe del proyecto accede a la aplicación.
2. El sistema muestra en una interfaz todas las operaciones que el Administrador de Proyecto puede realizar.
3. El jefe del proyecto selecciona la opción Modify Risk.
4. El sistema muestra una interfaz con los riesgos registrados.

Capítulo 3: Prueba y Seguimiento de la Solución Planteada

5. El Administrador de Proyecto selecciona el riesgo que va a ser modificado.
6. El Sistema presentará un formulario con los campos respectivos que facilitara que sean modificados por el Administrador de Proyecto, los campos a modificar son: Nombre, Clasificación, Descripción, Consecuencia, Impacto, Probabilidad, Mitigación, Contingencia, Activadores, Responsables y Resultado de la estrategia de los manejos de los riesgos.
7. El Administrador de Proyecto introduce solamente los datos que van a ser modificados.
8. El sistema guarda los datos en la base de datos.
9. El jefe del proyecto sale de la aplicación.

Condiciones de Ejecución

Debe de existir riesgos registrados en el sistema para que puedan ser modificados.

2.4 Iteraciones

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
El Administrador de Proyecto selecciona el riesgo a modificar y llena correctamente los campos que dese modificar ("Classification": de soporte).		El Sistema modifica correctamente los datos introducidos guardando los cambios realizados.	El Sistema modifica los datos del riesgo seleccionado de forma exitosa.	

CP #3: Eliminar Riesgos

Breve Descripción

Este caso de prueba nos permite eliminar uno o varios riesgos a través de un formulario en donde se muestran todos los riesgos que se han ido registrando hasta el momento en el sistema.

Capítulo 3: Prueba y Seguimiento de la Solución Planteada

Las pruebas realizadas a este escenario son las siguientes

- Eliminar uno o varios riesgos.

Flujo Central

1. El jefe del proyecto accede a la aplicación.
2. El sistema muestra en una interfaz todas las operaciones que el Administrador de Proyecto puede realizar.
3. El jefe del proyecto selecciona la opción Report the Identified Risks.
4. El Sistema presentará una interfaz con un reporte de todos los riesgos introducidos al sistema en donde el jefe del proyecto deberá de seleccionar el o los riesgos que desee que eliminar del sistema.
5. El Administrador de Proyecto selecciona los riesgos que van a ser eliminados.
6. El sistema actualiza la base de datos
7. El jefe del proyecto sale de la aplicación.

Condiciones de Ejecución

- Debe de existir al menos un riesgo registrado en el sistema para que puedan ser eliminado.

2.4 Iteraciones

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
El Administrador de Proyecto selecciona el/los riesgos que desea eliminar.		El Sistema elimina el/los riesgos satisfactoriamente.	El sistema elimina exitosamente el/los riesgos seleccionados.	

Capítulo 3: Prueba y Seguimiento de la Solución Planteada

Caso de Uso: Elaborar Estrategia de Manejo de los Riesgos

Este caso de uso permite al Administrador de Proyecto insertar respuesta de solución a los riesgos más críticos que estén registrados en el sistema.

Las pruebas realizadas en este caso de uso son:

- Insertar Respuesta

CP #1: Insertar Respuesta.

Breve Descripción

Este Caso de Prueba permite comprobar la funcionalidad de insertar una respuesta al riesgo seleccionado.

Flujo Central

1. El Administrador de Proyecto accede a la aplicación.
2. El sistema muestra en una interfaz todas las operaciones que el Administrador de Proyecto puede realizar.
3. El Administrador de Proyecto selecciona la opción a Analyze Critical Risk.
4. El sistema muestra un reporte de los riesgos que no han sido mitigados ordenados de acuerdo a su prioridad.
5. El Administrador de Proyecto selecciona el riesgo que quiere analizar.
6. El Sistema presentará un formulario con los campos respectivos que permite insertar la respuesta a un riesgo, los campos a llenar son: Mitigación, Contingencia, Activadores, Responsables y Respuesta del riesgo.
7. El sistema guarda los datos en la base de datos.
8. El jefe del proyecto sale de la aplicación.

Capítulo 3: Prueba y Seguimiento de la Solución Planteada

Condiciones de Ejecución

- Que el Administrador de Proyecto asegure la completitud de los datos introducidos.

2.4 Iteraciones

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
El jefe del proyecto introduce todos los campos correctos en el sistema:(<i>"Mitigación"</i> : Pedir con anterioridad los trabajos, <i>"Contingencia"</i> : Tomar medidas serias, <i>"Activadores"</i> : Entrega tardía de los trabajos, <i>"Responsables"</i> : Administrador de Proyecto, <i>"Respuesta"</i> : Las medidas tomadas nos facilito la entrega de los trabajos en tiempo").		El Sistema introduce los datos de forma satisfactoria.	El Sistema guarda los datos introducidos y los muestra satisfactoriamente.	
	El jefe del proyecto introduce los datos dejando campos en blanco :(<i>"Mitigación"</i> : <i>"Contingencia"</i> : Tomar medidas serias, <i>"Activadores"</i> : Entrega tardía de los trabajos, <i>"Responsables"</i> : Administrador de Proyecto, <i>"Respuesta"</i> : Las medidas tomadas nos facilito la entrega de los trabajos en tiempo").	El sistema muestra mensajes de error alertando que existen campos vacíos.	El sistema no introduce la información, lanzando de esta forma un error por existir campos vacíos.	

Capítulo 3: Prueba y Seguimiento de la Solución Planteada

	El Administrador de Proyecto introduce datos números en un campo :(<i>Mitigación</i> ”: Pedir con anterioridad los trabajos, <i>Contingencia</i> ”: Tomar medidas serias, <i>Activadores</i> ”: 547, <i>Responsables</i> ”: Administrador de Proyecto, <i>Respuesta</i> ”: Las medidas tomadas nos facilito la entrega de los trabajos en tiempo”).	El sistema muestra un mensaje de error alertando que existen datos incorrectos en algunos de los campos.	El sistema no introduce los datos por existir haber datos incorrectos por lo que se lanza un error.	
--	--	--	---	--

3.4 Manual de Usuario basado en los prototipos funcionales

“Insert Risk”

Para insertar un riesgo debe llenar todos los campos: nombre del riesgo, la clasificación, la descripción, sus consecuencias, junto con el impacto y la probabilidad. Luego de llenar correctamente todos los campos presionar *“Agree”* para guardar los datos. En caso de no estar los datos en formato correcto saldrá un mensaje de alerta explicándole que hay un error que necesita corregirse.

Nota:

Todos los campos tiene que comenzar y contener al menos una letra, menos el impacto y la probabilidad que no pueden estar vacío, solo se debe de escoger una de las opciones que muestran respectivamente.

Report the Identified Risk

Para eliminar un riesgo primero debes de escoger la opción *“Report the Identified Risks”* donde se mostrará un reporte de todos los riesgos registrados en el sistema, posteriormente se seleccionará los riesgos que van a ser eliminados, que puede ser uno o más. Luego de haber seleccionado los riesgos se selecciona el botón *“Eliminate”* permitiendo la eliminación de los riesgos escogidos.

Capítulo 3: Prueba y Seguimiento de la Solución Planteada

Modify Risk

Para modificar un riesgo solo se necesita escoger el que se desea modificar seleccionando la opción “*Modify Risks*”; una vez escogido el riesgo selecciona el botón “*Modify*” donde se mostrará un formulario que solamente se deberá llenar el campo que va a ser modificado y finalmente se presiona “*Send*”.

Nota:

Los campos que aparezcan vacíos una vez que se muestre el formulario no se deberán llenar, pues el riesgo seleccionado todavía no tiene una respuesta de solución.

Analizar Riesgos Críticos.

Para analizar los riesgos críticos primeramente se va a mostrar todos los riesgos con prioridad “Alta” que no han sido analizados, eso se logra seleccionando la opción “*Analyze Critical Risk*”, en donde se deberá de escoger el riesgo que va a ser analizado. Luego se mostrará un formulario con todos los campos a llenar: mitigación, contingencia, activadores, responsable y resultado del análisis. Después de haber llenado correctamente todos los campos presionar el botón “*Agree*”, para guardar los datos. En caso de no estar los datos en formato correcto saldrá un mensaje de alerta explicándole que hay un error que necesita corregirse.

Nota:

Todos los campos tienen que comenzar y contener al menos una letra, menos el responsable que no pueden estar vacío, solo se debe de escoger una de las opciones que se muestran.

3.5 Seguimiento y Control de la Gestión de Riesgos.

Para un efectivo tratamiento de los riesgos en el proyecto, y evitar que estos lleguen a afectar de alguna forma los planes trazados por líderes del mismo, se hace necesario un seguimiento y control de los riesgos una vez estos hayan sido tratados y se les haya dado una solución. Dándole así cumplimiento a las técnicas de GR planteadas en esta investigación (Planeación, identificación, análisis cualitativo, análisis cuantitativo, Planeación de la respuesta a los Riesgos y seguimiento y control de los Riesgos).

Una vez que el riesgo pasa por las primeras técnicas, planeación e identificación, este se analiza tanto cualitativamente como cuantitativamente para darle una prioridad al riesgo y según la prioridad asignarle un impacto que tendría sobre el proyecto, para luego, según estos datos darle una respuesta al riesgo en cuestión, pasando posteriormente a la última técnica de este ciclo que solidifica la GR con un seguimiento y control de los mismos para evitar que vuelva a amenazar los resultados del proyecto.

Capítulo 3: Prueba y Seguimiento de la Solución Planteada

3.6 Resultados presentes y visión futura.

La solución planteada ha arrojado resultados significativos para el perfeccionamiento del proceso de GR, resultados que se optimizaron tras el proceso de pruebas efectuado, específicamente usando el Método de Prueba de Caja Negra, método para el que solo basta conocer las posibles entradas y salidas del programa.

La implementación de esta herramienta, llevada a cabo por el jefe del proyecto, está destinada a establecer estrategias para todo el Polo; diseñada para identificar eventos potenciales que puedan afectar y administrar los riesgos para que proporcionen una razonable seguridad respecto al logro de los objetivos de cada uno de los proyectos.

3.7 Aplicación de la Herramienta de Gestión de Riesgos al SCADA

Luego de implementada y probada la extensión al Trac de Gestión de Riesgos, se procedió a aplicarla al proyecto SCADA Guardián del Alba, de forma tal que se realizaron todos los procesos identificados para la GR, lo que arrojó el siguiente resultado:

- Se logró registrar los riesgos correspondientes a la fase "Construcción 3" del SCADA, así como su correspondiente análisis y respuesta, con el objetivo de darle solución a los riesgos más críticos.

Teniendo de esta forma automatizado un proceso que en un inicio se tenía en hojas de cálculo, de manera que se tiene el punto de partida para darle seguimiento y control a estos riesgos como parte de la Gestión del Proyecto; permitiendo que la información perdure en el tiempo.

Al realizarle una encuesta al Líder del Proyecto SCADA sobre su satisfacción de la Extensión al Trac de GR implementada y probada al proyecto que él representa, se concluyó:

- Correcta implementación de los procesos de Gestión de Riesgos.
- Posibilidad de integrar la extensión al gestor de proyecto Trac, permitiendo obtener más funcionalidades a esta herramienta, garantizando la automatización de otro de los procesos de GP,
- Extender la aplicación de la herramienta a los demás proyectos del Polo de HA.

Capítulo 3: Prueba y Seguimiento de la Solución Planteada

3.8 Conclusiones

Al finalizar el presente capítulo se plantea la visión futura de la aplicación, cumpliendo con el propósito de ejecutar y diseñar el seguimiento y control propuesto para la solución planteada, asegurando la calidad de las mismas. Se elaboran y ejecutan casos de pruebas para los diferentes CU del Sistema, resolviendo las inconformidades lanzadas.

Conclusiones

Al concluir la presente investigación sobre el desarrollo de una herramienta para la GR en el Polo de HA se cumplió con el objetivo general en conjunto con cada uno de los objetivos específicos definidos:

Se elaboró un correcto proceso de GR al proyecto SCADA “Guardián del ALBA”.

Se obtuvo una extensión al Trac, facilitando la automatización de la GR para los proyectos productivos del Polo de HA.

Recomendaciones

1. Instalar la herramienta de GR en los proyectos del Polo de Hardware y Automática, haciéndolo extensivo a los proyectos que utilicen la herramienta Trac como gestor de proyectos.
2. Incorporar nuevas funcionalidades a la aplicación con el objetivo de alcanzar una versión más completa para la GR.

Referencias Bibliográficas

1. Zulueta Veliz, Yeleny. 2007. Maestría Modelo de GR en Proyectos de Desarrollo de Software. 2007.
2. Project Management Institute. 2008. Guide to the Project Management Body of Knowledge. 2008.
3. Gallegos, Javier Del Carpio. 2006. Análisis del riesgo en la administración de proyectos de tecnología de la información.
4. 12MANAGE the executive fasttrack. (2009). Retrieved from http://www.12manage.com/methods_brainstorming_es.htm
5. SEMA GROUP. (2006). Metodología de Análisis y GR de los Sistemas de Información (guía de técnicas).
6. López Crespo, Francisco and Amutio Gómez, Miguel Angel. 2006. Metodología de Análisis y GR de los Sistemas de Información. Madrid: s.n., 2006.
7. Lledo, P. *¿Cómo mitigar los riesgos del proyecto?*
8. GestioPolis. (2007). Obtenido de <http://www.gestiopolis.com/administracion-estrategia/estrategia/toma-de-decisiones-tecnica-del-arbol.htm>
9. Parsiani Shull, N. D. (2006). PROJECT EVALUATION USING FUZZY LOGIC.
10. Microsoft TechNet. 2009. <http://technet.microsoft.com/es-es/library/cc185712.aspx>
11. SIGEA. (2007). SIGEA. Obtenido de http://www.sigea.es/index.php?option=com_content&task=view&id=26&Itemid=26
12. Software in the Public Interest, Inc. 2009. Debian. [Online] 2009. [Cited: 02 16, 2009.]
13. Free Software Supporter. 2009. GNU Operating System. [Online] 2009. <http://www.gnu.org/home.es.html>.
14. *ArgenTIna – Software y Servicios Informáticos*. (2008). Obtenido de <http://www.cbasqa.wordpress.com/2008/09/02/proceso-de-desarrollo-openup/>

Referencias Bibliográficas

15. Python Software Foundation. 2009. Python. [Online] 2009. <http://www.python.org/>.
16. Edgewall Software. Genshi. [Online] <http://genshi.edgewall.org/>.
17. SQLite. [Online] 2009. <http://www.sqlite.org/>.
18. NetBeans UML®. 2009. Project. NetBeans UML® Project. [Online] 2009. <http://uml.netbeans.org/>.
19. *Trac Integreat SCM & Managment*. (2008). Retrieved from <http://www.trac.edgewall.org/>
20. The Eclipse Foundation. Eclipse. [Online] <http://www.eclipse.org>.
21. *MKM publicaciones*. (2007). Retrieved 04 3, 2009, from <http://www.mkmpi.com/mkmpi.php?article176>
22. *Tecnología y Synergix*. (2008). Retrieved 04 3, 2009, from <http://synergix.wordpress.com/2008/07/07/requisito-funcional-y-no-funcional/>

Bibliografía Consultada

1. **Argota Vega, Irina Elena, et al. 2008.** *Planificación de la GR del Proceso: "Comité de Compras" en la Sucursal CUBALSE Matanzas desde la Web.* 2008.
2. **Cedeño Gonzalez, Yusleimi. 2008.** *GR del proyecto Sistema de Gestión Penitenciaria.* 2008.
3. **G. Figueroa, Roberth, J. Solís, Camilo and A. Cabrera, Armando.** *METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES.*
4. **González Duque, Raúl. 2008.** *Python para Todos.* 2008.
5. **Hechavarría Guibert, Lisandra and Barbara May, Ariadna. 2008.** *GR en el proyecto Sistema de Gestión Fiscal.* 2008.
6. **Hernández Alba, Yailien and Fernández Pérez, Leonel Duvier. 2008.** *Guía para la GR a través de RUP.* 2008.
7. **Iliana, Pérez Pupo. 2008.** *GR en la línea Seguridad del SCADA.* 2008.
8. **Intaver Institute. 2009.** INTAVER INSTITUTE. [Online] 2009. <http://www.intaver.com/index-whitepapers.html>.
9. **Lamas. 2007.** *Propuesta para la Gestión de Riesgo en los proyectos productivos de la UCI.* La Habana, s.n., 2007.
10. **Microsoft Corporation. 2009.** MicrosoftTechNet. [Online] 2009. <http://technet.microsoft.com/es-es/library/cc185712.aspx>.
11. **Overseas Gerenciamiento de Riscos. 2009.** OVERSEAS. [Online] 2009. <http://www.overseasbr.com/es/riskmanagement/risksolutons/arm.asp>.
12. **RiskTrak International. 2009.** Experts in Risk Management. [Online] 2009. <http://risktrak.com>.

Anexo

1. Encuentra sobre Riesgos.

Nombre del proyecto al que pertenece: _____ (SCADA - Galba, SIGGE, SCADA - Etecsa, Supervisión Energética - UCI, Supervisión Energética - UNE, TetScada, SCADA - Nico López, SCADA - Hospitales, SCADA - Edificios Inteligentes, Siprogas, SCIA)

Preguntas:

Marque con una X.

1- ¿UD. como Líder de Proyecto Gestiona los riesgos de su proyecto?

Si
 No

En caso de que la respuesta sea "Si" continúe registrando la siguiente información.

2- ¿Qué proceso(s) de la GR realiza?

Planificación de Riesgos
 Identificación de Riesgos
 Análisis Cualitativo de los Riesgos
 Análisis Cuantitativo de los Riesgos
 Planificación de la Respuesta de los Riesgos
 Seguimiento y Control de los Riesgos

3- ¿Utiliza alguna herramienta para Gestionar los Riesgos?

Si
 No

En caso de que la respuesta sea "Si". Especificar cual _____

Glosario de Términos

API: Application Program Interface “Interfaz de programación de aplicaciones.”.

API: Una interfaz de programación de aplicaciones o API (del inglés **A**pplication **P**rogramming **I**nterface) es el conjunto de funciones y procedimientos que ofrece ciertas bibliotecas para ser utilizado por otro software.

Base de datos o banco de datos: Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (electrónico), que ofrece un amplio rango de soluciones al problema de almacenar datos.

Extensión: Extensión o componente es una aplicación que se relaciona con para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de las API.

FODA: Significa fortalezas, oportunidades, debilidades y amenazas, también puede ser usado durante la identificación de riesgos. Ayuda a identificar los riesgos positivos y negativos que aplican a un proyecto.

Gestión de Proyecto: La gestión de proyectos es la disciplina de organizar y administrar recursos de manera tal que se pueda culminar todo el trabajo requerido en el proyecto dentro del alcance, el tiempo, y coste definidos.

Gestión de Riesgo: Es un enfoque estructurado para manejar la incertidumbre relativa a una amenaza, a través de una secuencia de actividades que incluyen identificación y evaluación de riesgo, estrategias de desarrollo para manejarlo y mitigación del riesgo.

GNU/Linux: El resultado de la combinación GNU y Linux es un sistema operativo libre, multiplataforma, multiusuario y multitarea. GNU/Linux es completamente configurable y optimizable, posee un funcionamiento muy rápido ya que es capaz de utilizar todas las posibilidades de hardware del sistema.

Glosario de Términos

Hardware: Corresponde a todas las partes físicas y tangibles de una computadora: sus componentes eléctricos, electrónicos, electromecánicos y mecánicos; sus cables, gabinetes o cajas, periféricos de todo tipo y cualquier otro elemento físico involucrado; contrariamente al soporte lógico e intangible que es llamado software.

Herramienta de Gestión de Proyectos: Son aquellas herramientas que soportan la realización de aplicaciones, actividades o acciones como la generación, codificación o transferencia del datos.

HTML: siglas de Hipertexto Markup Lenguaje (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas web.

HTTP: de Hipertexto Transfer Protocol (Protocolo de transferencia de hipertexto) es el método más común de intercambio de información en la world wide web, el método mediante el cual se transfieren las páginas web a un ordenador.

Informática: es la ciencia del tratamiento automático de la información mediante una computadora

LDAP: Lightweight Directory Access Protocol “”

Lenguaje de Programación: Es aquel elemento dentro de la informática que nos permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes.

LGPL: licencia General Public Licence.

Metodología: Se refiere a los métodos de investigación que se siguen para alcanzar una gama de objetivos en una ciencia, es decir, es el conjunto de métodos que se rigen en una investigación científica.

Multiplataforma: Es un término usado para referirse a los Programas, sistemas operativos, lenguajes de programación u otra clase de software, que puedan funcionar en diversas plataformas.

NetBeans: una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación.

Glosario de Términos

OpenUp: Proceso Unificado Abierto es un marco del proceso del desarrollo del software de código abierto, diseñado para pequeños equipos organizados, quienes quieren tomar una aproximación ágil del desarrollo, que en un cierto plazo, se espera que cubra un amplio sistema de necesidades para los proyectos de desarrollo.

PDVSA: Petróleos de Venezuela S.A.

Programación: En informática es un proceso por el cual se escribe, se prueba, se depura y se mantiene el código fuente de un programa informático. Dentro de la informática, los programas son los elementos que conforman el software, que es el conjunto de las instrucciones que ejecuta el hardware de una computadora para realizar una tarea determinada.

Programador: Persona que diseña, escribe y/o depura programas de ordenador o computadora, es decir, quien diseña la estrategia a seguir, propone las secuencias de instrucciones y/o escribe el código correspondiente en un determinado lenguaje de programación

Proyecto: Es un esfuerzo temporal, único y progresivo, emprendido para crear un producto o un servicio también único.

Riesgo: Es la eventualidad de un daño; que en cualquier momento se puede materializar o no hacerlo nunca.

SCADA: Supervisory Control And Data Acquisition “Sistema supervisor de adquisición de datos”.

Script: Pequeños programas incrustados en las paginas que nos permiten definir interactividades de cualquier tipo.

Server Apache: Apache es programa de servidor HTTP Web de código abierto. Fue desarrollado en 1995 y actualmente es uno de los servidores web más utilizados en la red.

Software: Palabra proveniente del inglés (literalmente: partes blandas o suaves), que en nuestro idioma no posee una traducción adecuada al contexto, por lo cual se utiliza frecuentemente sin traducir y fue admitida por la Real Academia Española (RAE). Aunque no es estrictamente lo mismo, suele sustituirse por expresiones tales como programas (informáticos) o aplicaciones (informáticas).

Sqlite3: Es un sistema de gestión de bases de datos. SQLite sólo necesita una librería (biblioteca) que ocupa poco más de 700KB (tclsqlite3.dll). Desarrollada en C.

Glosario de Términos

Trac: Es un sistema web para la gestión de proyectos y seguimiento de errores.

UML: Lenguaje Unificado de Modelado, es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.

UNIX: sistema operativo originalmente desarrollado en 1969.

XML: Son las siglas de Extensible Markup Lenguaje, Lenguaje de marcas extensible, es un meta-lenguaje. Extensible de etiquetas desarrollado por el World Wide Web Consortium.