

Universidad de las Ciencias Informáticas

Facultad 6



Título:

“Sistema para el análisis cuantitativo de riesgos”.

Trabajo de Diploma para optar por el título de
Ingeniero Informático

Autor(es): Danny Hernández Hernández

Tutor(es): Ing. Geidis Sánchez Michel

Junio 2013

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis que tiene por título: “Sistema para el análisis cuantitativo de riesgos” y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Danny Hernández Hernández

Geidis Sánchez Michel

Firma del Autor

Firma del Tutor

DATOS DE CONTACTO

Tutor: Ing. Geidis Sánchez Michel.

Graduado en la Universidad de las Ciencias Informáticas en julio del 2007. Jefa de departamento de Tecnología Facultad 1.

AGRADECIMIENTOS

Quisiera agradecer a todas aquellas personas que a lo largo de estos años de estudio me han brindado su apoyo incondicional y su sabiduría. Gracias por todos los consejos valiosos y su paciencia, porque siempre estuvieron ahí cuando los necesité.

A Dios por darme la fuerza y la convicción necesaria para completar esta tarea. Gracias por guiarme de manera impecable a través de los muchos obstáculos que tuve en el camino y por mantenerme firme cuando todo parecía perdido. Gracias por tu protección y por las muchas señales a lo largo del camino.

A mis padres, en primer lugar por haberme dado la vida, el regalo más valioso que se pueda tener. Gracias por inculcarme los valores que me han hecho convertirme en la persona que hoy soy. Gracias por enseñarme que el estudio no es una obligación sino un deber.

A mi abuela Lázara Caridad por su apoyo incondicional. Su entereza es digna de admirar y me ha servido de ejemplo invaluable en este proceso.

A Dariam, más que mi primo mi hermano. A mi tía Mercedes por su ánimo y firmeza ante la vida.

A mi tía y segunda madre Mabel a los jimaguas, Rity y Pedro, gracias por siempre estar presentes aunque estén lejos a la vez.

A mi hermana Rosemary y abuela Nancy, Basy, Anthony y Andrea por su cariño y especiales atenciones para con nosotros.

A mi tutora Geidis por su confianza y apoyo desinteresado. En ti no solo tuve una tutora, también una amiga.

A mi padrino Walfrido su esposa Maritza y mis primos Marla y Walfry por su preocupación constante y el apoyo recibido. Un agradecimiento especial a Elena y Ana María por su preocupación constante y saber que siempre podía contar con su ayuda.

A María de los Ángeles Rodríguez Pérez y Lourdes Pérez a las cuales considero parte de mi familia. Gracias por el apoyo incondicional y consejos en momentos que fueron muy difíciles para mí.

A mi amigo Yanniél por su apoyo y estima, gracias por creer en mí.

Al grupo 6CPT (D) por las largas horas que pasamos todos estudiando y brindándonos apoyo los unos a los otros. Demostramos que recorrer este camino se trata de una carrera de resistencia. Gracias por la unión que siempre tuvimos para afrontar las asignaturas más difíciles.

Al colectivo de profesores que a lo largo de estos años han aportado su granito de arena para convertirnos en futuros profesionales.

A mis amigos del tecnológico Yuniel el gordo, Yosnay, Elvis, Yasel y Yosvany el calvo que siempre me preguntaban cómo iban los estudios y la tesis.

A todas estas personas y las que he olvidado, muchísimas gracias.

Danny Hernández Hernández

DEDICATORIA

Dedico el presente trabajo de diploma a mis padres Jorge y Maura, son lo más grande que un ser humano pueda tener.

RESUMEN

La Gestión de Riesgos es una disciplina que requiere gran atención en la actualidad, debido a los altos índices de fracaso en los proyectos de desarrollo de software. Este proceso se compone de varios pasos, donde la identificación de los riesgos constituye la base para realizar el análisis cualitativo y cuantitativo, que permite comprender numéricamente los efectos negativos que estos eventos puedan producir.

La Universidad de las Ciencias Informáticas desarrolla soluciones mediante proyectos productivos. En el Centro de Informatización Universitaria (CENIA), el Análisis Cuantitativo se realiza de forma manual, lo cual no constituye la vía óptima para realizar este proceso. Por este motivo los líderes del centro mencionado dedican tiempo productivo a esta tarea donde las reuniones efectuadas quedan reflejadas solamente en documentos formales. Los líderes del CENIA cuentan con el apoyo de la herramienta para la gestión de proyectos (GESPRO); que permite dar seguimiento y control sobre los riesgos detectados, aunque no brinda la posibilidad de aplicar técnicas numéricas para comprender mejor el impacto de los riesgos.

Por estos elementos el objetivo de este trabajo constituye el desarrollo de una herramienta que permita realizar el Análisis Cuantitativo de los riesgos detectados empleando varias técnicas y la posibilidad de obtener reportes que reflejen los resultados obtenidos de la aplicación de estas técnicas.

Palabras clave: análisis cuantitativo, probabilidad, riesgos

TABLA DE CONTENIDOS

INTRODUCCIÓN.....	9
CAPÍTULO 1: FUNDAMENTO TEÓRICO.....	12
1.1. Riesgo.....	12
1.2. Gestión de riesgos.....	12
1.3. Procesos de la gestión de riesgos.....	14
1.4. Herramientas existentes para la Gestión de riesgos.....	15
1.4.1. Herramientas existentes en la UCI.....	17
1.5. Técnicas para el análisis cuantitativo de riesgos.....	18
1.6. Ambiente de desarrollo.....	23
1.7. Conclusiones.....	30
CAPÍTULO 2: DESCRIPCIÓN DEL SISTEMA.....	31
2.1. Análisis cuantitativo de riesgos en el centro CENIA.....	31
2.2. Lista de Reserva del producto.....	32
2.3. Identificación de los actores.....	35
2.4. Historias de Usuario.....	36
2.5. Plan de iteraciones.....	40
2.6. Plan de entrega.....	41
2.7. Prototipos de interfaz.....	43
2.8. Conclusiones.....	44
CAPÍTULO 3:IMPLEMENTACIÓN.....	45
3.1. Implementación del sistema.....	45
3.2. Empleo de metáforas.....	46
3.3. Tarjetas CRC.....	48
3.4. Patrones de diseño.....	49
3.5. Diseño de la Base de Datos.....	52
3.6. Estándares de codificación.....	54
3.7. Diagrama de componentes.....	55
3.8. Ejemplos de código implementado.....	56
3.9. Diagrama de despliegue.....	58
3.10. Interfaces.....	59
3.11. Conclusiones.....	60
CAPÍTULO 4: PRUEBAS.....	531
4.1. Pruebas.....	61
4.2. Casos De Prueba.....	61
4.3. Resultados Obtenidos.....	73
CONCLUSIONES.....	675
RECOMENDACIONES.....	686
REFERENCIAS BIBLIOGRÁFICAS.....	69
BIBLIOGRAFÍA.....	719
GLOSARIO DE TÉRMINOS.....	82
ANEXOS.....	84

ÍNDICE DE TABLAS Y FIGURAS.

Tabla 1. Cálculo de la evaluación colectiva virtual de los riesgos	20
Tabla 2. Evaluación colectiva sobre los riesgos analizados	21
Tabla 3. Lista de reserva del producto	32
Tabla 4. Relación de requisitos no funcionales del sistema	34
Tabla 5. Relación de usuarios del sistema	35
Tabla 6. HU Gestionar riesgo	36
Tabla 7. HU Gestionar expertos	37
Tabla 8. HU Asignar evaluación lingüística a los riesgos	37
Tabla 9. HU Aplicar evaluación colectiva virtual de riesgos	38
Tabla 10. HU Mostrar lista de riesgos priorizados	38
Tabla 11. HU Gestionar decisión	38
Tabla 12. HU Gestionar acontecimiento	39
Tabla 13. HU Gestionar resultado	39
Tabla 14. HU Obtener camino óptimo	40
Tabla 15. Relación de HU con tiempo estimado de implementación	41
Tabla 16. HU abarcadas por módulos	42
Tabla 17. Relación de entrega de los módulos con sus versiones	43
Tabla 18. Tarjeta CRC clase facultad	48
Tabla 19. Tarjeta CRC clase centro	48
Tabla 20. Tarjeta CRC clase persona	48
Tabla 21. Tarjeta CRC clase proyecto	49
Tabla 22. Tarjeta CRC clase riesgo	49
Tabla 23. Tarjeta CRC clase rol	49
Tabla 24. Descripción de la tabla riesgo	53
Tabla 25. Descripción de la tabla decisión	53
Tabla 26. Descripción de la tabla acontecimiento	53
Tabla 27. Descripción de la tabla resultado	53
Tabla 28. Descripción de la tabla experto	54
Tabla 29. Descripción de la tabla criterio	54
Tabla 30. Caso de Prueba Autenticar Usuario	61
Tabla 31. Caso de Prueba Insertar riesgo	62
Tabla 32. Caso de prueba Mostrar riesgos	62
Tabla 33. Caso de prueba Eliminar riesgo	63
Tabla 34. Caso de prueba Modificar riesgo	63
Tabla 35. Caso de prueba Asignar un riesgo a un proyecto	63
Tabla 36. Caso de prueba insertar experto	64
Tabla 37. Caso de prueba Mostrar expertos	64
Tabla 38. Caso de prueba Eliminar experto	65
Tabla 39. Caso de prueba Modificar experto	65

Tabla 40. Caso de prueba Asignar etiqueta virtual a cada riesgo.....	65
Tabla 41. Caso de prueba Mostrar lista de riesgos priorizados.....	66
Tabla 42. Caso de prueba Insertar decisión.....	66
Tabla 43. Caso de prueba Mostrar decisión.....	67
Tabla 44. Caso de prueba Eliminar decisión.....	67
Tabla 45. Caso de prueba Modificar decisión.....	67
Tabla 46. Caso de prueba insertar acontecimiento.....	68
Tabla 47. Caso de prueba Mostrar acontecimiento.....	68
Tabla 48. Caso de prueba Eliminar acontecimiento.....	69
Tabla 49. Caso de prueba Modificar acontecimiento.....	69
Tabla 50. Caso de prueba Insertar resultado.....	69
Tabla 51. Caso de prueba Mostrar resultados.....	70
Tabla 52. Caso de prueba Eliminar resultado.....	70
Tabla 53. Caso de prueba Modificar resultado.....	70
Tabla 54. Caso de prueba Asignar relaciones entre nodos.....	71
Tabla 55. Caso de prueba Obtener camino óptimo.....	71
Tabla 56. Caso de prueba Calcular exposición al riesgo.....	72
Tabla 57. Caso de prueba Insertar rol.....	72
Tabla 58. Caso de prueba Mostrar rol.....	72
Tabla 59. Caso de prueba Eliminar rol.....	72
Tabla 60. Caso de prueba Modificar rol.....	72
Figura 1. Ejemplo de árbol de decisión.....	22
Figura 2. Prototipo de interfaz para la Autenticación del usuario.....	43
Figura 3. Prototipo de interfaz. Vista principal del sistema.....	43
Figura 4. Prototipo de interfaz. Módulo de administración del sistema.....	44
Figura 5. Diagrama Modelo – Vista - Controlador.....	46
Figura 6. Diagrama de paquetes (Administración).....	47
Figura 7. Clase facultad.....	50
Figura 8. Clases riesgo y categoría.....	50
Figura 9. Clases persona y proyecto.....	51
Figura 10. Clases centro y proyecto.....	51
Figura 11. Modelo físico de datos.....	52
Figura 12. Diagrama de componentes del sistema.....	56
Figura13. Diagrama de despliegue.....	58
Figura 14. Panel de administración del sistema.....	59
Figura 15. Formulario para la gestión de proyectos.....	60
Figura 16. Interfaz para la gestión de riesgos.....	60
Figura 17. Totales de funcionalidades probadas y no conformidades por iteración.....	73

INTRODUCCIÓN

El hombre con su incesante quehacer científico-técnico ha logrado en pocos años revolucionar el entorno en que se desenvuelve debido al desarrollo alcanzado en las Tecnologías de la Información y las Comunicaciones (TIC), permitiendo cambiar muchos aspectos de la vida moderna.

En los últimos años la cantidad de información creada por la actividad del hombre y transmitida por Internet ha alcanzado niveles fuera de la imaginación, hecho que queda constatado en el estudio realizado por la corporación norteamericana Egan Marino C. La cantidad de información generada en 2010 es igual a 100 años de Tweets y a 75 billones de iPads. (1)

Las tecnologías desarrolladas por el hombre, en conjunto con la información generada, ha propiciado grandes avances en cuanto al uso y explotación de este importante activo. Puede afirmarse además que es un vehículo para la transmisión de conocimiento y ha constituido un factor esencial en el avance de la sociedad (2). Como ejemplo importante se puede mencionar el uso de la información en los proyectos de desarrollo de software.

Los altos niveles de información generada, han propiciado el desarrollo de nuevas tecnologías, herramientas y proyectos para satisfacer las necesidades de informatización de la sociedad. Estos últimos pueden variar en cuanto a costos, alcance y plazo de entrega, por lo que es necesario contar con un equipo preparado para hacer frente a los factores mencionados. Aunque datos estadísticos han demostrado que esto no es suficiente para alcanzar los resultados deseados. Según Pressman en 1998, los datos industriales indicaron que el 26 por ciento de los proyectos de software fracasaron y que el 46 por ciento rebasaron sus costos y tiempos de entrega.

La Universidad de las Ciencias Informáticas (UCI) forma parte de la avanzada en cuanto a producción de software en el país. Para emprender esta tarea es necesario contar con el equipamiento tecnológico adecuado y un equipo de profesionales que contribuyan a que los proyectos cumplan con los objetivos trazados.

En la UCI existen varios centros productivos que desarrollan soluciones para clientes nacionales e internacionales. El Centro de Informatización Universitaria (CENIA), perteneciente a la Facultad 1 tiene bajo su dirección varios proyectos. En la actualidad el análisis cuantitativo de los riesgos en el CENIA se realiza de forma manual y por varios especialistas mediante un panel de expertos. La información generada de estos encuentros queda registrada en documentos formales. Los especialistas cuentan

con la herramienta para la gestión de proyectos (GESPRO) la cual permite dar seguimiento y control a los riesgos, aunque no brinda la posibilidad a los usuarios de aplicar una o varias técnicas de análisis cuantitativo.

Teniendo en cuenta estos elementos, se plantea el siguiente **problema a resolver**: ¿Cómo contribuir al análisis cuantitativo de los riesgos detectados?

Dado este **problema** se presenta como **objeto de estudio**: procesos de la gestión de riesgos y como **objetivo general**: Desarrollar un sistema web para realizar el análisis cuantitativo de los riesgos detectados, estableciendo como **campo de acción**: Técnicas para el análisis cuantitativo de riesgos.

Teniendo como **objetivos específicos**:

1. Desarrollar el marco teórico de la investigación mediante el estudio de las técnicas y herramientas para el análisis cuantitativo de riesgos.
2. Diseñar el sistema para el análisis cuantitativo de riesgos.
3. Implementar el sistema para el análisis cuantitativo de riesgos.
4. Realizar pruebas al sistema implementado para comprobar su correcto funcionamiento.

Para cumplir con el objetivo planteado se definen las siguientes **tareas de la investigación**:

1. Análisis de sistemas para el análisis cuantitativo de los riesgos.
2. Análisis de las principales metodologías, tecnologías y herramientas a utilizar para el desarrollo de la solución propuesta.
3. Definición de las características y funcionalidades del sistema a desarrollar.
4. Construir los artefactos requeridos en la ingeniería de software del sistema comprendidos en las fases de la metodología seleccionada.
5. Definición de la arquitectura del sistema como base para el desarrollo de la solución.
6. Implementación de las clases modelos, vistas y controladoras para el desarrollo del sistema.
7. Elaboración de los casos de prueba para el sistema análisis cuantitativo de riesgos.
8. Aplicar las pruebas funcionales al sistema análisis cuantitativo de riesgos.

Métodos teóricos

Análisis-Sintético: se utiliza para el análisis de las técnicas empleadas en el análisis cuantitativo de riesgos.

Modelación: se utiliza para proponer un diseño que contribuya a la solución del problema planteado.

Método empírico

Entrevista: se utiliza para la recopilación de información sobre los principales problemas que presenta el centro CENIA en el análisis cuantitativo de riesgo.

CAPÍTULO 1: FUNDAMENTO TEÓRICO.

En el presente capítulo se aborda los conceptos fundamentales asociados al dominio del problema. Se describen las técnicas para el análisis cuantitativo de los riesgos y se definen la metodología de desarrollo de software, tecnologías y herramientas a utilizar, que permiten: guiar, modelar y desarrollar la propuesta de solución.

1.1. Riesgo

Para tener una mejor comprensión de este término se tendrán en cuenta las siguientes definiciones:

- Es una medida de la desviación de un resultado esperado o deseado que afecta al proyecto y que en la mayoría de los casos tiende a sobrepasar su presupuesto inicial o su fecha planificada de culminación (3).
- De acuerdo al diccionario de la Real Academia de la lengua española, define al riesgo como contingencia o proximidad de un daño.
- Un riesgo de un proyecto según la Guía de los Fundamentos de la Dirección de proyectos (PMBOK) es un evento o condición que, si se produce, tiene un efecto positivo o negativo sobre al menos un objetivo del proyecto, como tiempo, coste, alcance o calidad (es decir, cuando el objetivo de tiempo de un proyecto es cumplir con el cronograma acordado; cuando el objetivo de coste del proyecto es cumplir con el coste acordado).
- El riesgo se relaciona con los acontecimientos futuros, de acuerdo a Pressman e implica elección y la incertidumbre que conlleva. El Proceso Unificado de Rational expone que la gestión de riesgos se encarga de los aspectos desconocidos y lo define como una variable que con su distribución normal puede tomar un valor que ponga en peligro el éxito de un proyecto.

La palabra riesgo encierra diversos significados de acuerdo al criterio de los autores que la definan. Podemos decir que con ella se describe, desde el sentido común, la posibilidad de perder algo (o alguien) o de tener un resultado no deseado, negativo o peligroso (4).

1.2. Gestión de riesgos.

La Gestión de Riesgos (GR) constituye una serie de pasos que ayudan al equipo del software a comprender y a gestionar la incertidumbre. Un proyecto de software puede presentar múltiples problemas. Un riesgo es un problema potencial ya que estos pueden ocurrir o no. Pero sin tener en cuenta el resultado, es necesario identificarlo, evaluar su probabilidad de aparición, estimar su impacto y establecer un plan para gestionarlos, luego de ser identificados. Es por esto que en la actualidad, en las etapas iniciales del desarrollo de un proyecto se trata de identificarlos, aunque este proceso no

debe relegarse solamente a esta etapa. Debe tenerse en cuenta que los riesgos no estarán presentes solamente al iniciarse una etapa. Debido a esto es necesario contar con una metodología o guía que brinde los pasos necesarios para identificarlos, evaluarlos y mitigarlos a lo largo del proceso de desarrollo del producto. A continuación se abordarán varios criterios sobre la GR, haciendo énfasis en el análisis cuantitativo que constituye el objeto de la investigación.

- El Proceso Unificado de Rational comprende dentro de la disciplina administración de proyectos a la GR. De acuerdo a la metodología se deben identificar los riesgos potenciales que puedan amenazar al proyecto los cuales se incorporan a la lista de riesgos y se les prioriza teniendo en cuenta el impacto que pueda tener así como su probabilidad de ocurrencia para luego calcular la exposición al riesgo. La disciplina posee artefactos como: el plan de mitigación de riesgos y estrategias de mitigación de riesgos.
- El PMBOK proporciona a los líderes de proyectos una herramienta efectiva para la gestión de los procesos que se desarrollan en el negocio. Comprende de forma general muchas áreas identificadas y que están presentes en los proyectos, proporcionando artefactos que se relacionan entre sí. En esta guía una de las áreas del conocimiento es la GR en la cual se abordan los aspectos a tener en cuenta para llevar a cabo eficazmente este proceso.
- Métrica 3 es una metodología de planificación, desarrollo y mantenimiento de sistemas de información, contiene actividades y estas a su vez se descomponen en tareas. Dentro del análisis de la viabilidad del sistema, una de las tareas se dedica al estudio de los riesgos, valorándolos en cuanto a su posible impacto en el negocio y tomar medidas para minimizarlos.
- De acuerdo al criterio del Modelo Integrado de Capacidad y Madurez (CMMI) el propósito de la GR es identificar los problemas potenciales antes de que ocurran para darles algún tratamiento. El modelo contempla en su nivel 3 actividades encaminadas a la GR que van desde la Preparación, Identificación, Análisis hasta la Mitigación de riesgos. Específicamente para la tarea de evaluación refiere que a los riesgos deben asignársele parámetros de probabilidad y exposición al riesgo, para establecer una prioridad en su atención.

Después de haber analizado las metodologías anteriormente expuestas, se puede afirmar que existen semejanzas entre estos ya que llevan a cabo tareas de identificación, análisis y seguimiento de los riesgos detectados. El presente trabajo hará énfasis en el Análisis Cuantitativo, aunque todos los procesos llevan a cabo esta tarea. Se seleccionó al PMBOK como guía de esta tarea específica pues

emplea métodos más profundos para conocer las posibles implicaciones que puede traer consigo los riesgos.

1.3. Procesos de la gestión de riesgos.

La GR se compone de un orden lógico con el propósito de aumentar la probabilidad y el impacto de los eventos positivos y disminuir la probabilidad y el impacto de los eventos adversos para el proyecto. Este proceso está compuesto por una serie de pasos que detallan los elementos a tener en cuenta. Abarca elementos importantes para el control de costes, tiempo de entregas, además de gestión de los recursos humanos. Por ser una guía que incluye disímiles procesos, se ha convertido en un estándar a tener en cuenta para la gestión de proyectos. Los procesos definidos utilizan elementos en común o entradas, para producir salidas empleando estos artefactos, los cuales son procesados mediante herramientas que se traduce en la aplicación de mecanismos para el procesamiento de la información. Los elementos resultantes pueden ser utilizados por otras fases como entradas o sea, se solapan unos con otros.

Para el desarrollo de este trabajo es importante conocer los elementos que conforman o incluyen la gestión de riesgos. Específicamente para el control de riesgos, proporciona un compendio de pasos para la gestión eficaz de estos, lo cual incluye la planificación, identificación, análisis cualitativo, análisis cuantitativo, planificación de la respuesta y control de riesgos. El análisis cualitativo es el proceso que permite evaluar los riesgos además de evaluar las causas probables de su ocurrencia.

Para realizar este análisis en un proyecto deben haberse previamente identificado las variables a cuantificar. Los expertos y personas asociadas al proyecto deben cooperar en este esfuerzo, de esta forma no solo se identifican otros posibles riesgos, también se comparten los criterios o números de los impactos que puedan tener cada riesgo sobre los objetivos. La palabra “cuantitativo” encierra un significado numérico y se refiere a cantidades.

El análisis cuantitativo consiste en la evaluación de los riesgos identificados. El proceso de análisis cuantitativo analiza el efecto de los riesgos y les asigna una calificación numérica. Los elementos de evaluación cuantitativa más comunes son: la probabilidad de ocurrencia y su impacto. Para la aplicación de estas técnicas deben previamente haberse identificado los riesgos y agruparlos en diferentes categorías.

Categorías de riesgos.

El establecimiento de categorías para los riesgos permite identificar específicamente a qué área del proyecto puede afectar y de acuerdo a este criterio establecer un tratamiento personalizado a la potencial amenaza. De acuerdo al criterio de autores especializados en ingeniería de software y

gestión de proyectos, han definido diferentes categorías para los riesgos. Los criterios de Pressman y Sommerville tienen puntos de coincidencia, debido a que en el proceso de desarrollo pueden existir riesgos asociados al proyecto, negocio, técnicos, estimación, requisitos y con el personal involucrado en el proyecto. Aunque también pueden existir aquellos riesgos conocidos y los impredecibles, los primeros surgen de la experiencia adquirida en el desarrollo de anteriores proyectos, mientras que los últimos son más difíciles de identificar.

Por otra parte el PMBOK propone una estructura compuesta por 4 grupos generales de riesgos y un total de 18 subcategorías. Esta estructura puede emplearse como referencia en el momento de realizar el análisis cualitativo de los riesgos pues permite o contribuye a una identificación más rápida y eficiente de los riesgos, este listado se conoce por el nombre de estructura de desglose de riesgos (EDR), que puede ser adaptada a las características del proyecto en cuestión.

1.4. Herramientas existentes para la gestión de riesgos.

La era de la tecnología ocupa espacios cada vez mayores en la vida cotidiana. El desarrollo tecnológico alcanzado por el hombre le ha permitido desarrollar numerosas herramientas para facilitar el trabajo que en tiempos anteriores tenía que ser realizado de forma manual. Las herramientas que sirven de soporte a la gestión de proyectos contribuyen a mejorar los procesos que se desarrollan dentro de las organizaciones. Este aspecto cobra gran importancia, dado que muchos proyectos fracasan por la combinación de múltiples factores. A continuación se describen algunas soluciones para realizar análisis cuantitativo de riesgos a nivel internacional y en la UCI. Se destacan sus principales características, así como ventajas y desventajas.

Microsoft Project.

Herramienta desarrollada por la empresa norteamericana Microsoft. Aunque este software ha sido etiquetado como miembro de la familia Microsoft Office hasta el momento no ha sido incluido en ninguna de las ediciones de Office (5). Cuenta con varias funcionalidades, permite el modelado de Diagramas de Gantt para la gestión de las actividades, posibilita el resaltado de cada actividad en diversos colores, permitiendo identificar de forma más sencilla aquellas que son críticas para el proyecto. Se pueden modelar además diagramas de ruta crítica empleados para determinar aquellas actividades con mayor peso en el proyecto.

Como principales desventajas hay que destacar que es software privativo, hay que pagar por su uso y no está disponible para plataformas abiertas como Linux. Por otra parte no posee la funcionalidad de realizar análisis cuantitativo empleando métodos como los árboles de decisiones.

Soft Expert (SE) Risk.

Cuenta con amplias funcionalidades para la gestión de proyectos. Los usuarios finales pueden interactuar a través de una interfaz web. Permite dar seguimiento a los riesgos, organizándolos por categorías. Otra funcionalidad para la gestión de riesgos se ejecuta mediante árboles de decisiones, donde el usuario puede realizar organigramas con diferentes alternativas que se puedan presentar. Puede ser adaptada en dependencia de las necesidades de gestión, lo cual incluye proyectos, procesos, estrategias, seguridad del consumidor, activos y peligros. Para el análisis cuantitativo el sistema presenta un formulario donde establece los datos necesarios para realizar la evaluación del riesgo (6).

El análisis cuantitativo, en esta herramienta se encuentra limitado a la inserción y edición de fórmulas sencillas. Aunque permite el modelado de árboles de decisión, hay que pagar para hacer uso de sus funcionalidades.

Primavera Risk Analysis.

Es una herramienta específicamente orientada a la gestión de riesgos en proyectos. Propiedad de la empresa norteamericana Oracle, posee otra solución que permite gestionar proyectos, ofreciendo variadas funcionalidades para este fin. Es una solución de análisis de riesgo de todo el ciclo de vida que integra los costos y la gestión de riesgos del programa. Es una aplicación que utiliza el método de Montecarlo y el método de la ruta crítica para simular miles de escenarios en los que un proyecto puede variar, reportando las distribuciones probabilísticas de costos y tiempo de proyectos generados (6). Primavera Risk Analysis proporciona un medio integral para determinar niveles de confianza para el éxito del proyecto, junto con técnicas rápidas y sencillas para planes de respuesta de contingencia y de riesgo (7). A pesar de que simula escenarios posibles para la gestión de riesgos, la principal desventaja radica en el pago de una licencia para hacer uso de este software.

RiskyProject.

Es una herramienta de escritorio que permite desarrollar planificaciones y análisis cuantitativo de riesgos, se integra con Microsoft Project, Primavera Risk y otras aplicaciones de gestión de proyectos. Risky Project permite registrar riesgos, priorizarlos, definir matrices de riesgos, análisis de Montecarlo (8).

- Entre sus principales funcionalidades destacan:
- Registrar riesgos de una forma sencilla;
- Asignar riesgos a tareas o al proyecto de forma global
- Personalizar categorías de riesgos

- Asignar riesgos a recursos
- Visualizar los resultados del análisis de riesgos en diagramas de Gantt
- Visualizar matrices de probabilidad-impacto;
- Análisis probabilístico del Valor Ganado;
- Análisis de sensibilidad (diagramas de Tornado)
- Análisis de Montecarlo
- Diagramas waterfall de riesgos
- Gestión de planes de respuesta y de mitigación
- Múltiples distribuciones de probabilidad (beta, gamma, normal, laplace, etc...) (8)

A pesar que permite aplicar el método Montecarlo para el análisis cuantitativo de riesgos, es una herramienta privativa y su costo de adquisición es de \$999 dólares norteamericanos.

@Risk para Project.

Es una herramienta de escritorio orientada específicamente al análisis cuantitativo de riesgos. Puede integrarse a Microsoft Project permitiendo tener una completa suite tanto para la gestión de proyectos, así como el análisis cuantitativo. Permite la creación de diagramas de Gantt para establecer la ruta crítica de las actividades, así como la separación por categorías de los riesgos detectados. En cuanto a la simulación y muestreo, posee las siguientes capacidades:

- El modo de Demo actualiza las gráficas y los reportes en cada iteración.
- Existen ocho generadores de números aleatorios disponibles.
- El número máximo de iteraciones para el detenido automático es ahora 50,000.
- Una nueva ventana del progreso de la simulación reporta estadísticas adicionales de cada corrida y permite cambiar los parámetros de la simulación durante la corrida.
- Existe un nuevo monitor de alto rendimiento para el @RISK Accelerator. (9)

A pesar de poseer un conjunto de funcionalidades para el análisis cuantitativo, es una herramienta privativa, por lo que hay que pagar por su uso.

1.4.1. Herramientas existentes para la gestión de riesgos en la UCI.

La UCI cuenta con un número significativo de proyectos productivos, los cuales no están exentos de riesgos a lo largo del proceso de desarrollo. Para gestionarlos y cuantificar los riesgos se han desarrollado varias soluciones. A continuación se hace un análisis de algunas de estas.

Sistema para el análisis cuantitativo de riesgos de los proyectos productivos del Centro de Informática Industrial (CENDIN).

Solución propuesta en un Trabajo de Diploma. Desarrollada para ser adaptada a las necesidades de los proyectos productivos pertenecientes al CENDIN. En el Trabajo de Diploma se hace un análisis de algunas técnicas empleadas para el análisis cuantitativo de riesgos, tomándose como más factibles el modelo de simulación conocido como Montecarlo y la técnica de recopilación y representación de datos (Distribuciones de probabilidad).

Extensión al Redmine: Análisis cuantitativo de riesgos en proyectos de software.

Solución propuesta en un Trabajo de Diploma. Fue desarrollada con el objetivo de integrarla a la herramienta de gestión de proyectos Redmine. En esta propuesta de forma similar a la anterior se hace un análisis de algunas técnicas empleadas para el análisis cuantitativo de riesgos. De las técnicas analizadas se aplica o implementa el método de Montecarlo.

Las herramientas analizadas para la gestión de proyectos, cubren la posibilidad de emplear métodos basados en la simulación e incluyen la técnica árboles de decisión. En algunos casos puede emplearse solamente uno de estos métodos. Como principal desventaja hay que destacar que para hacer uso de las funcionalidades incluidas se establece un pago por la adquisición de licencias. En el caso de las soluciones que existen en la UCI se hace un estudio de algunas técnicas para el análisis cuantitativo de riesgos. En ambos casos se toma el método de Montecarlo como modelo a implementar. Debido a los elementos anteriormente expuestos, se hace necesaria la realización de un sistema que contemple otras técnicas para el análisis cuantitativo de riesgos, incluyendo el Modelo Lingüístico Virtual, método propuesto para el CENIA por el MsC. Osiris Pérez Moya.

1.5. Técnicas para el análisis cuantitativo de riesgos.

El análisis cualitativo es una fase que se inserta dentro de la gestión de los riesgos. Esta práctica permite la recopilación de los riesgos que pueden preverse a lo largo del proceso de desarrollo. No solo es necesario conocer los riesgos para trazar diversas estrategias y controlarlos, también es importante el empleo de técnicas numéricas para medir su impacto en el negocio.

Panel de Expertos.

Se refiere a la participación de personas especializadas en un determinado tema. En estas reuniones, emiten criterios sobre la temática que se está tratando con la particularidad de que pueden coincidir o

no. Los especialistas dialogan en un lenguaje informal, opinando sobre una parte del tema general, suponiendo que cada participante se especialice en un área del tema. En el caso de la Gestión de riesgos puede aplicarse no solo a la fase de identificación de riesgos. Como parte del análisis cuantitativo los especialistas pueden emitir valores sobre los riesgos identificados, los cuales pueden ser la probabilidad (P) y el impacto (I). Pressman plantea un método matemático para conocer la exposición al riesgo (E). Para cuantificar un riesgo de acuerdo a este modelo es necesario conocer la probabilidad de ocurrencia y el coste o impacto sobre el proyecto. La probabilidad es emitida de acuerdo al criterio de expertos o personal vinculado directamente al proceso de desarrollo, teniendo los valores de probabilidad se expresan como puntos porcentuales. La variable impacto se refiere al peso económico que puede tener el riesgo en cuestión. La relación entre estas variables es luego expresada mediante la relación: $E = P \times I$. A partir de los resultados de exposición al riesgo obtenido se pueden establecer comparaciones entre los diferentes proyectos para conocer aquellos que poseen altos índices de este valor.

Modelo Lingüístico Virtual.

Es un modelo propone el uso de variables lingüísticas, las cuales adquieren diversos valores discretos definidos, mediante criterios de los especialistas involucrados en el proceso. Para realizar este proceso se define un conjunto de siete valoraciones o alternativas lingüísticas. En forma de resumen se presentan a continuación el conjunto de pasos a seguir para aplicar este método.

- 1- Definir los expertos que emitirán criterios lingüísticos sobre la exposición a un riesgo. Este número se expresa mediante la variable e_j . De esta forma el conjunto queda: **$E = \{e_1, \dots, e_k\}$**
- 2- Los expertos emiten o definen el conjunto de términos lingüísticos para expresar el grado de ocurrencia o gravedad de un evento o alternativa x_j . De esta forma el conjunto para la selección de la preferencia queda: **$S = \{S_1 = \text{Muy Alta (MA)}, S_2 = \text{Bastante Alta (BA)}, S_3 = \text{Alta(A)}, S_4 = \text{Moderada (M)}, S_5 = \text{Baja (B)}, S_6 = \text{Muy Baja (MB)}, S_7 = \text{No Afecta (NA)}\}$**

En este caso como se aplica al análisis cuantitativo de riesgos, los expertos emiten criterios para valorar la probabilidad (P) y el impacto (I) del riesgo en cuestión. Ambos criterios pueden tener el mismo valor lingüístico en una misma alternativa. Por lo que puede decirse que $P = I$.

- 3- Utilizando el operador lingüístico virtual para la multiplicación lingüística virtual.

$$S_\alpha \times S_\beta = S_\beta \times S_\alpha = S_{\alpha\beta}$$

Se calcula la media aritmética que tiene como objetivo obtener un consenso de los criterios lingüísticos emitidos por los expertos o participantes en el proceso para cada riesgo. La media aritmética se calcula mediante la fórmula:

$$\bar{r} = \frac{1}{n} \sum_{i=1}^n S_{\alpha_i}^n \otimes S_{\beta_i}^n$$

Donde:

\bar{r} Es el valor de la preferencia sobre el riesgo n.

n Es la cantidad de riesgos a analizar.

$S_{\alpha_i}^n$ Es el valor del impacto del riesgo n emitido por la persona i.

$S_{\beta_i}^n$ Es el valor de la probabilidad del riesgo n emitido por la persona i

En la Tabla 1 se resumen las operaciones descritas anteriormente:

Tabla 1. Cálculo de la evaluación colectiva virtual de los riesgos (10)

persona	RIESGO 1		RIESGO 2		RIESGO n	
	Probabilidad	Impacto	Probabilidad	Impacto	Probabilidad	Impacto
e_1	$S_{\beta_1}^1$	$S_{\alpha_1}^1$	$S_{\beta_1}^2$	$S_{\alpha_1}^2$
e_2	$S_{\beta_2}^1$	$S_{\alpha_2}^1$	$S_{\beta_2}^2$	$S_{\alpha_2}^2$
e_i	$S_{\beta_1}^n$	$S_{\alpha_1}^n$
Evaluación colectiva virtual de riesgos					$\bar{r}_n = \frac{1}{n} \sum_{i=1}^n S_{\alpha_i}^n \otimes S_{\beta_i}^n$	

4- Una vez obtenidos los valores numéricos correspondientes a cada riesgo, se ordena descendientemente. De esta forma se prioriza su atención.

A continuación se muestra un ejemplo del funcionamiento de la técnica del análisis lingüístico virtual.

A un grupo de tres especialistas se les solicita su criterio acerca de probabilidad e impacto sobre tres riesgos detectados en un proyecto. Para este proceso se emplean las etiquetas virtuales mencionadas anteriormente. De esta forma el conjunto de datos quedaría conformado de la siguiente forma:

- riesgos - $r = r_1, r_2, r_3$
- personas - $e = e_1, e_2, e_3$
- etiquetas virtuales - $S = \{S_1 = \text{Muy Alta (MA)}, S_2 = \text{Bastante Alta (BA)}, S_3 = \text{Alta(A)}, S_4 = \text{Moderada (M)}, S_5 = \text{Baja (B)}, S_6 = \text{Muy Baja (MB)}, S_7 = \text{No Afecta (NA)}\}$

Una vez que los especialistas emiten los valores de probabilidad e impacto sobre los riesgos a analizar, se procede a obtener el valor de las preferencias colectivas. Un especialista elige, sobre un riesgo las etiquetas: Muy Alta (MA) y Moderada (M) para los valores de probabilidad e impacto respectivamente. Para obtener la preferencia en términos numéricos, se multiplican los números correspondientes a la posición que ocupan estas ocupan en el conjunto de etiquetas virtuales. En este caso la etiqueta Muy Alta (MA) tiene un valor igual a S_1 , la etiqueta Moderada (M) tiene un valor igual a S_4 . El resultado entonces sería igual a S_4 , el proceso continúa hasta que cada especialista ha emitido sus criterios sobre cada riesgo. La Tabla 2 muestra los resultados obtenidos.

Tabla 2. Evaluación colectiva sobre los riesgos analizados.

persona	riesgo 1		riesgo 2		riesgo 3	
	P	I	P	I	P	I
e_1	M	MA	A	M	MA	M
e_2	M	BA	BA	MA	BA	A
e_3	A	BA	A	BA	M	A
$\bar{r}_n =$	$= \frac{1}{3}(S_4 + S_8 + S_6)$ $= \frac{1}{3}(S_{18})$ $= S_6$		$= \frac{1}{3}(S_{12} + S_2 + S_6)$ $= \frac{1}{3}(S_{20})$ $= S_{6,66}$		$= \frac{1}{3}(S_4 + S_6 + S_{12})$ $= \frac{1}{3}(S_{22})$ $= S_{7,33}$	

Una vez calculada la preferencia colectiva sobre los riesgos, se procede a su ordenamiento. Para este caso queda de la siguiente forma: riesgo 3, riesgo 2 y riesgo 1.

Árboles de decisión.

Los árboles de decisión se utilizan cuando estamos frente a un riesgo en el cual pueden presentarse diferentes alternativas de decisión o eventos. Los eventos pueden ocurrir o no de acuerdo a cada alternativa de decisión. Este método permite identificar los posibles cursos de acción que puedan presentarse ante un riesgo determinado y el objetivo que persigue es la elección de la mejor opción viable. La aplicación de este método supone un conjunto de beneficios pues permite analizar: el problema de forma precisa y evaluar múltiples situaciones, analizar las consecuencias en el momento de tomar una decisión y la selección de aquella que brinde mejores resultados basándose en la información obtenida.

Los árboles de decisión se componen de:

- Decisiones: Se representan mediante un cuadrado y resultan el punto de partida de las diferentes acciones. El árbol puede contener más decisiones partiendo de una primaria. Las decisiones tienen asignadas un costo de implementación.
- Acontecimientos: Se representan mediante un círculo y constituyen los diferentes cursos de acción que puede tener una decisión. Cada acontecimiento tiene asignado una probabilidad de ocurrencia, la suma de estos valores tiene que ser igual a uno.
- Resultados: Se representan mediante un rectángulo y constituyen el resultado de tomar un curso de acción determinado.

Para realizar los cálculos pertinentes se tendrán en cuenta la probabilidad para cada curso de acción (acontecimientos) y el resultado que traería consigo cada elección. Con estos elementos se procede a calcular el valor esperado para cada nodo. La relación entre estas variables se representa mediante: $E(X) = \sum_{i=1}^m x_i p(x_i)$, la fórmula representa la multiplicación de cada resultado con su respectivo valor de probabilidad, este cálculo se realiza para cada resultado asociado a un acontecimiento, para luego obtener el valor esperado de un nodo. Una vez realizado este procedimiento para cada acontecimiento del árbol, se procede a restar el valor esperado del costo de implementación de cada decisión, para obtener el beneficio de cada curso de acción. Para obtener el curso de acción óptimo del árbol, se selecciona aquel que mayor beneficio aporte o sea el de mayor valor numérico.

A continuación se muestra un ejemplo del funcionamiento de la técnica de árboles de decisión.

El árbol para este caso partirá de una decisión primaria (D_p), la cual está relacionada con dos acontecimientos (A₁ y A₂) y estos últimos a su vez con dos resultados cada uno. La Figura 1 muestra el árbol de decisión con sus nodos relacionados.

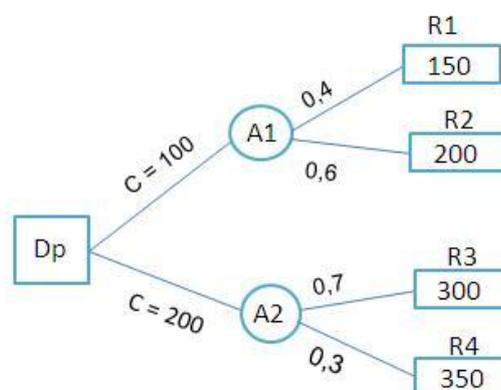


Figura 1. Ejemplo de árbol de decisión.

El proceso de resolución comienza desde atrás hacia adelante.

Valor esperado para el nodo A₁

Se multiplican los valores de los resultados R₁ y R₂ por sus respectivas probabilidades asociadas, para luego sumar los valores obtenidos.

$$VE = (150 \times 0,4) + (200 \times 0,6)$$

$$VE = 180$$

Valor esperado para el nodo A₂

$$VE = (300 \times 0,7) + (350 \times 0,3)$$

$$VE = 315$$

Luego de haber obtenido estos valores se procede a obtener el beneficio de seguir un curso de acción específico, que consiste en restar el valor esperado para cada nodo del costo (C) de implementar o seguir un curso de acción.

Beneficio para la decisión primaria (D_p) al nodo acontecimiento A₁

$$B = VE - C$$

$$B = 180 - 100 = 80$$

Beneficio para la decisión primaria (D_p) al nodo acontecimiento A₂

$$B = VE - C$$

$$B = 315 - 200 = 115$$

Con los resultados obtenidos, se compara el que mayor beneficio aporte y será la ruta a seguir en el árbol. En este caso la ruta que aporta un mayor beneficio es partir de la decisión primaria D_p y seguir hacia el nodo acontecimiento A₂.

1.6. Ambiente de desarrollo.

Para el desarrollo de una propuesta de solución a la problemática planteada se analizaron varias metodologías de desarrollo, así como las herramientas y tecnologías a utilizar para la implementación y modelado del sistema.

Metodología: Programación Extrema (XP)

Creada por Kent Beck, toma pasos de otras metodologías para realizar un desarrollo más agradable y sencillo. Tiene como base la simplicidad y la satisfacción del cliente, para lograrlo se deben tener en cuenta cuatro valores fundamentales:

Comunicación: Es importante que haya comunicación constante con el cliente y dentro de todo el equipo de trabajo, de esto dependerá que el desarrollo se lleve a cabo de una manera sencilla, entendible y que se entregue al cliente lo que necesita.

Simplicidad: En la XP se refiere que ante todo y sin importar qué funcionalidad requiera el usuario en su sistema, éste debe ser fácil. El diseño debe ser sencillo y amigable al usuario, el código debe ser simple y entendible, programando sólo lo necesario y lo que se utilizará.

Retroalimentación: El cliente forma parte del equipo de desarrollo, al cual se le consultan los posibles cambios a realizar y se le muestran los avances o historias de usuario que se implementen. El cliente es una fuente importante de retroalimentación.

Coraje: Se refiere a la valentía que se debe tener al modificar o eliminar el código que se realizó con tanto esfuerzo; el desarrollador debe saber cuándo el código que desarrolló no es útil en el sistema y, por lo mismo, debe ser eliminado. También se refiere a tener la persistencia para resolver los errores en la programación.

Propone un número reducido de roles, dentro de los cuales el cliente es uno de ellos. A lo largo del proceso de desarrollo se producen pequeñas partes del sistema que se han ido implementando. El código generado es compartido para todo el equipo, cualquier miembro puede hacer cambios, y es modificado a lo largo de las iteraciones. Una particularidad de esta metodología es que propone la programación en pares, consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo (11). Todo proceso de desarrollo de software lleva consigo una etapa de planificación. Una parte importante de esta etapa en XP son las historias de usuario, las cuales constituyen un artefacto que recoge las necesidades del usuario, que se traduce en requisitos tanto funcionales como no funcionales que poseerá el sistema.

Metodología: SCRUM

Esta metodología, de forma similar a XP, recoge las necesidades del usuario en un documento con una prioridad asignada para su implementación. Sigue un conjunto de iteraciones, las cuales son planeadas previamente en reuniones del equipo de trabajo. En estas reuniones, el equipo de trabajo decide cuáles serán los elementos prioritarios a incluir en la iteración en cuestión. El chequeo del trabajo realizado se realiza periódicamente, de esta forma se nutre el proceso de desarrollo con los resultados obtenidos. SCRUM también se utiliza para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costes se disparan o la calidad no es aceptable (11).

Metodología: OpenUp

Es una metodología basada en el Proceso Unificado de Rational, pero no incorpora muchos elementos de este proceso. Adopta principios que distinguen a las metodologías ágiles. Es un proceso de desarrollo iterativo, incremental y guiado por casos de uso. Por otra parte OpenUp/Basic es aplicable a equipos de desarrollo pequeños y posee tres características distintivas. Mínimo porque solo incluye el contenido del proceso fundamental. Completo porque puede ser manifestado como proceso entero para construir un sistema. Extensible porque puede ser utilizado como base para agregar o adaptar más procesos (11). El proceso de desarrollo está dividido en cuatro fases. En la fase inicial o Inicio se determinan y planifican elementos como: visión del proyecto, costes, requisitos críticos y se explora una posible solución inicial. La fase de elaboración va encaminada a una descripción más detallada de los requisitos del sistema y la definición de la línea base de la arquitectura. La fase de construcción tiene como objetivo fundamental el desarrollo iterativo del sistema hasta que esté lista una versión de prueba para hacer su transición. En la fase de transición, se le aplican pruebas a la versión obtenida de las iteraciones de la fase anterior para validar que cumpla con las expectativas del usuario.

Las metodologías tradicionales, se adaptan a las necesidades de grandes proyectos. Su aplicación genera gran cantidad de artefactos y no contempla al cliente como parte del equipo de desarrollo. Debido a estos elementos se decide optar por la metodología ágil XP. Entre las razones para esta decisión están: poco tiempo para el desarrollo de la investigación, reducido número de integrantes del equipo de desarrollo y cantidad mínima y necesaria de artefactos para el análisis y diseño del sistema.

Lenguaje de Modelado Unificado: UML 2.0

Un lenguaje de modelado es un conjunto de símbolos y elementos gráficos usados para representar mediante estos, la modelación de un sistema. Los lenguajes de modelado permiten no solo crear abstracciones del sistema que se quiere modelar, sino también del lenguaje de programación que se use. El desarrollo de UML tiene sus inicios en el año 1994 y es fruto del esfuerzo de: Grady Booch, Ivar Jacobson y James Rumbaugh. UML es un conjunto estandarizado de notaciones gráficas y modelos para ayudar a describir el diseño de sistemas de software, especialmente aquellos que utilizan el paradigma de programación orientado a objetos. Permite la abstracción del modelado del sistema independientemente del lenguaje de programación que se pretenda utilizar para el desarrollo de un proyecto, además de modelar una amplia variedad de diagramas utilizados en la actualidad para el desarrollo de sistemas (12).

Herramienta CASE: Visual Paradigm 8.0.

Las herramientas CASE (Computer Assisted Software Engineer por sus siglas en inglés) incluyen un conjunto de funcionalidades que facilitan la optimización de un producto ofreciendo apoyo permanente a los analistas, ingenieros de software y desarrolladores (13). Visual Paradigm es una herramienta CASE que permite el modelado de sistemas, con los respectivos diagramas que necesitemos y provistos por UML. Permite la generación de código a partir de diagramas, así como reportes a diversos formatos. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, implementación y pruebas. Ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor coste (14).

Entorno de desarrollo integrado: Netbeans 7.3.

Los entornos de desarrollo integrado ofrecen a los usuarios finales un conjunto de funcionalidades encaminadas a la construcción de aplicaciones de diversos tipos, además de ejecución, compilación y aplicación de pruebas al código fuente. Tienen como objetivo mejorar la productividad del desarrollador ofreciéndole un conjunto de herramientas totalmente cohesionadas entre sí, a través de un interfaz gráfica de usuario (15). La edición para PHP de este popular entorno de desarrollo integrado ofrece a los desarrolladores un conjunto de funcionalidades para construir aplicaciones web de forma sencilla y eficiente. Es una plataforma de desarrollo totalmente gratuita en la que se pueden desarrollar proyectos para varios lenguajes de programación. Una de las grandes ventajas que incluye esta versión es la de poder utilizar HTML5, CSS3 y JavaScript (16). Entre otras ventajas que ofrece este IDE se pueden mencionar:

- Completamiento de código PHP así como su resaltado.
- Depuración de código, de esta forma es posible conocer el comportamiento de las diferentes variables durante la ejecución de un proyecto.
- Producto multiplataforma, se encuentra disponible para: Windows, GNU/Linux y MacOSX.
- Soporte para el control de versiones.
- Producto libre y sin restricciones de uso.

Framework de desarrollo: CodeIgniter 2.1

Es un framework o entorno de trabajo para los desarrolladores que utilizan el lenguaje de programación PHP y puede ser descargado y utilizado sin restricciones de licencia. Permite la creación de aplicaciones web y cuenta con un conjunto de librerías para agilizar este proceso, este aspecto realza su utilidad debido a las demandas actuales de proyectos complejos por parte de los clientes. CodeIgniter implementa el modelo-vista-controlador que separa la lógica del negocio, la interfaz de usuario y los datos del sistema, ayudando programar de forma ordenada. En comparación con otros entornos de trabajo, posee características que lo distinguen, como: su rapidez, extensibilidad y compatibilidad con PHP 4.0 y 5.0. Permite a los desarrolladores web mejorar la forma de trabajar y también agregar mayor velocidad a la hora de crear una página web (17).

Entre las características generales, se encuentran las siguientes:

- **Documentación:** Puede encontrarse en forma de tutorial, que es más sencilla de asimilar para usuarios inexpertos.
- **Instalación sencilla:** Debido a que es una aplicación desarrollada en PHP, solamente necesitamos de un servidor web debidamente configurado para hacer uso de las funcionalidades que brinda.
- **Librerías:** El conjunto de librerías incluidas posibilitan la creación de no solo contenido dinámico, además de tareas comunes e importantes en el proceso de desarrollo como acceso a bases de datos y validación de formularios.

Lenguaje de programación: PHP 5.3

Se trata de un lenguaje interpretado y de alto nivel que se ejecuta del lado del servidor. Es generalmente utilizado para la creación de sitios web y goza de gran popularidad actualmente. Sus inicios datan del año 1994 y es creado por el programador Rasmus Lerdorf. A lo largo de los años ha ido evolucionando y agregando funciones para hacer posible la interacción con nuevas tecnologías y es actualizado continuamente para ampliar sus capacidades. El código PHP puede ser embebido en páginas HTML para agregarle dinamismo al contenido incluido. Las páginas web dinámicas son aquellas que permiten la interacción con los clientes. Las bases de datos cubren la necesidad actual de almacenar y recuperar información en el momento que sea necesario, PHP entre sus características a destacar, es que ofrece soporte para diversos tipos de bases de datos, tanto de renombre comercial como libres (Oracle, InterBase, MySQL, PostgreSQL, entre otras). Es software

libre, por lo que no hay que preocuparse por la compra de licencias ni pagos de ningún tipo para hacer uso de las bondades que brinda. Posee una amplia comunidad activa, debido a esto es posible encontrar ayuda rápida y fácilmente a través de foros y documentación actualizada. Permite la interacción con diversos protocolos de red como: SNMP, IMAP y LDAP. Esta última interfaz brinda la posibilidad de proveer autenticación a usuarios pertenecientes a un dominio y obtención de datos de estos. Es utilizado en aplicaciones Web-relacionadas por algunas de las organizaciones más prominentes tales como Mitsubishi, Redhat, Der Spiegel, MP3-Lycos, Ericsson y NASA (18).

Librería ExtJs 4.1

Es una librería para JavaScript que facilita el desarrollo de aplicaciones web. Sus inicios datan del año 2006, creado por el desarrollador Jack Slocum como una extensión a la librería YUI (Yahoo User Interface por sus siglas en inglés). Soporta diversos navegadores actuales como: Firefox, Internet Explorer, Opera y Safari. Ofrece una amplia colección de elementos de interfaz gráfica para ser utilizados y adaptados a las características de un nuevo proyecto, incluye tablas, gráficos, campos numéricos y de fecha con menú desplegable. Estas bondades permiten además de reducir el tiempo de desarrollo y elevar las prestaciones gráficas de la aplicación. Como beneficios de forma general se pueden mencionar los siguientes:

- Existe un balance entre Cliente – Servidor. La carga de procesamiento se distribuye, permitiendo que el servidor, al tener menor carga, pueda manejar más clientes al mismo tiempo.
- Comunicación asíncrona. En este tipo de aplicación el motor de render puede comunicarse con el servidor sin necesidad de estar sujeta a un clic o una acción del usuario, dándole la libertad de cargar información sin que el cliente se dé cuenta.
- Eficiencia de la red. El tráfico de red puede disminuir al permitir que la aplicación elija que información desea transmitir al servidor y viceversa, sin embargo la aplicación que haga uso de la pre-carga de datos puede que revierta este beneficio por el incremento del tráfico (19).

Servidor de aplicaciones Web: Apache 2.2.21.

Un servidor es un programa ejecutado en una computadora con el fin de responder a las solicitudes de múltiples clientes, posibilitando el envío de textos, figuras, formularios y elementos multimedia. Estas aplicaciones tienen en común el uso del protocolo HTTP (Protocolo de Transferencia de Hipertexto). Apache es un servidor de código abierto para aplicaciones web, el cual fue desarrollado por la Fundación Apache. El hecho de que su código sea abierto implica que pueda ser modificado y

distribuido sin restricciones de ningún tipo, para diversos fines que también pueden ser comerciales. Utiliza el protocolo de transferencia HTTP para la gestión de las transacciones que se llevan a cabo entre el cliente y el servidor. Soporta bases de datos así como el protocolo de seguridad HTTPS para la protección de la información. Se encuentra disponible para GNU/Linux y Windows. Apache es una muestra, al igual que el sistema operativo Linux (un Unix desarrollado inicialmente para PC), de que el trabajo voluntario y cooperativo dentro de Internet es capaz de producir aplicaciones de calidad profesional difíciles de igualar (20). Hay que destacar que es altamente configurable, rápido y eficiente, además de contar con otras características como su fiabilidad y extensibilidad ya que se pueden agregar módulos para ampliar sus capacidades ya sea para generar contenido dinámico empleando diversos lenguajes de programación, monitorear el rendimiento del servidor, atender peticiones encriptadas, limitación del ancho de banda y creación de servidores virtuales.

Sistema Gestor de Base de Datos: PostgreSQL 9.1

Los SGBD surgen de la necesidad de manipular el creciente volumen de información en formato digital. Los SGBD permiten la creación de bases de datos así como su manipulación. En estos sistemas se proporciona un conjunto coordinado de programas, procedimientos y lenguajes que permiten a los usuarios realizar sus tareas habituales con los datos. PostgreSQL es un SGBD creado por la universidad de Berkeley. Es una herramienta de código abierto a pesar de estar licenciada por la institución mencionada. El código puede ser leído, modificado, copiado y distribuido. Las diferentes versiones son mantenidas por una amplia comunidad de desarrolladores voluntarios y posee una amplia documentación. Permite el almacenamiento de elementos multimedia como imágenes y videos. Es capaz de manejar grandes volúmenes de datos, característica importante debido al aumento de información digital. Soporta casi toda la sintaxis de SQL y es: extensible, confiable y multiplataforma. Posee entre otras características con las siguientes:

- Cuenta con un rico conjunto de tipos de datos, permitiendo además su extensión mediante tipos y operadores definidos y programados por el usuario.
- Su administración se basa en usuarios y privilegios.
- Es altamente confiable en cuanto a estabilidad se refiere.
- Control de concurrencia multiversión, lo que mejora sensiblemente las operaciones de bloqueo y transacciones en sistemas multiusuario.
- Soporte para vistas, claves foráneas, integridad referencial, disparadores, procedimientos almacenados y casi todos los tipos y operadores soportados en SQL92 y SQL99.

- Implementación de algunas extensiones de orientación a objetos. En PostgreSQL es posible definir un nuevo tipo de tabla a partir de otra previamente definida (21).

1.7. Conclusiones.

Con el estudio de sistemas que posibilitan la gestión de riesgos así como las técnicas de análisis cuantitativo empleadas, se llega a la conclusión de han servido para acelerar el proceso de gestión de riesgos en proyectos. Contribuyen a dar una respuesta anticipada a los eventos adversos que hayan sido detectados. Algunos de estos sistemas tienen la limitante de enmarcarse solamente en la gestión de riesgos, lo cual no constituye el objetivo de esta investigación. Para hacer uso de las soluciones a nivel mundial, se establece un pago por la adquisición de las respectivas licencias. Este último aspecto, interfiere con las políticas de migración hacia el software libre que sigue la UCI y el país. Otro aspecto importante a tener en cuenta es que estas soluciones no se adaptan a las necesidades específicas de la UCI, además de haber sido implementadas utilizando tecnologías propietarias. A partir de estos elementos expuestos, surge la necesidad de realizar un sistema que responda a las necesidades del centro CENIA. Para realizar dicho trabajo se utilizarán las herramientas y tecnologías siguientes:

- ✓ Metodología de desarrollo XP debido al poco tiempo con que se cuenta para el desarrollo de la solución y poseer los artefactos necesarios para el desarrollo de la solución.
- ✓ Lenguaje de programación PHP versión 5.3 por su popularidad para el desarrollo de aplicaciones para la web.
- ✓ Framework de desarrollo para PHP CodeIgniter versión 2.1 porque es una herramienta libre y facilita el trabajo de programación en el lenguaje PHP.
- ✓ Visual Paradigm versión 8.0 como herramienta de modelado ya que cubre los diagramas necesarios para el modelado de la solución.
- ✓ Como entorno de desarrollo integrado Netbeans versión 7.1, es una herramienta libre y sin restricciones de uso además de estar disponible para varias plataformas.
- ✓ Como sistema Gestor de Bases de Datos se utilizará PostgreSQL versión 9.1 por ser un SGBD libre y confiable.

CAPÍTULO 2: DESCRIPCIÓN DEL SISTEMA

En el presente capítulo se describen las características del sistema a implementar. Comprende las historias de usuario y lista de reserva del producto, que resultan de gran importancia su correcta identificación, pues representan las necesidades de los usuarios. Se identifican además las funcionalidades que compondrán al sistema y que serán la base para su implementación. También se realizan los diagramas de casos de uso, así como las personas que se beneficiarán directamente de su uso.

2.1. Análisis cuantitativo de riesgos en el centro CENIA.

Actualmente el análisis cuantitativo de riesgos en el CENIA se realiza de forma manual. De esta manera el proceso puede tomar más tiempo en llevarse a cabo, lo que conlleva a emplear este valioso recurso en una tarea que puede ser objeto de automatización. El resultado de las sesiones de trabajo queda registrado solamente en documentos formales, por lo que consultar estos resultados puede volverse un proceso tedioso y que conlleve también a pérdida de tiempo.

Objeto de automatización.

Debido a los elementos planteados anteriormente, existen procesos que pueden ser objeto de automatización ya que la forma en que se ejecutan actualmente no satisface las necesidades del centro. El objetivo de este trabajo es facilitar estas tareas que hoy se realizan de forma manual, por lo que se automatizará todo lo relacionado con el tratamiento de los riesgos (inserción en el sistema), gestión de los usuarios, análisis a aplicar (técnicas cuantitativas), así como los reportes para facilitar el manejo de los resultados.

Propuesta de sistema.

Se propone la implementación de un sistema que brinde una serie de funcionalidades que se adapten a las necesidades del CENIA. El sistema debe ser capaz de gestionar los diferentes riesgos que se detecten, siendo estos la base para la aplicación de las diferentes técnicas de análisis cuantitativo. Las técnicas mencionadas anteriormente se utilizan para diversos fines, permitiendo priorizar riesgos y establecer el valor económico de estos. El sistema contará con las respectivas secciones para la aplicación de estas técnicas. No solo contará con esta posibilidad, además contará con la capacidad de presentar mediante informes los resultados obtenidos. Estos informes les brindarán la posibilidad a los usuarios de consultarlos. Se dividirá el sistema en los siguientes módulos:

Módulo de administración.

El módulo contiene las funcionalidades necesarias para gestionar los usuarios que tendrán acceso al sistema. Se podrá a través de esta vista, agregar un nuevo usuario, asignándole su respectivo rol en el sistema, modificar los datos y rol de un usuario existente y eliminar un usuario registrado. El sistema también permitirá listar o mostrar los usuarios registrados en el sistema. También se podrá gestionar los centros, proyectos y nuevas facultades a los que se les quiera registrar en el sistema.

Módulo de análisis.

El módulo contiene las técnicas de análisis cuantitativo que el usuario dispone para aplicar. Hay que tener en cuenta que para realizar esta acción deberá estar registrado en el sistema al menos un riesgo. Luego de haber completado el paso anterior el usuario puede seleccionar la correspondiente técnica que desee aplicar. El usuario dispone además de la opción de guardar los resultados obtenidos.

Módulo de reportes.

El módulo ofrece las funcionalidades al usuario de consultar los resultados de los análisis aplicados. Los reportes en formato HTML, podrán ser consultados y filtrados por facultad, centro, proyecto, proyectos por facultad así como los resultados obtenidos de la aplicación de las técnicas de análisis cuantitativo.

2.2. Lista de reserva del producto.

La identificación de los requisitos permite tener una correcta visión del producto que se quiere construir. Estos elementos se clasifican en dos grupos: requisitos Funcionales y no funcionales. La ingeniería de requisitos proporciona el mecanismo apropiado para entender lo que el cliente quiere, analizar las necesidades, evaluar la factibilidad, negociar una solución razonable, especificar la solución sin ambigüedades, validar la especificación y administrar los requisitos. A continuación en la siguiente tabla 2 y tabla 3 se especifican los diferentes requisitos con los que debe cumplir el sistema.

Tabla 3. Lista de reserva del producto.

Código	Descripción	Prioridad
Funcionalidades del Sistema		
1	Autenticar Usuario	Media
2	Adicionar Usuario	Media
3	Mostrar Usuarios	Media
4	Eliminar Usuarios	Media

5	Adicionar Facultad	Media
6	Modificar Facultad	Media
7	Eliminar Facultad	Media
8	Mostrar Facultades	Media
9	Adicionar Proyecto	Media
10	Modificar Proyecto	Media
11	Eliminar Proyecto	Media
12	Mostrar Proyectos	Media
13	Adicionar Centro	Media
14	Modificar Centro	Media
15	Eliminar Centro	Media
16	Mostrar Centros	Media
17	Adicionar Riesgo	Alta
18	Modificar Riesgo	Alta
19	Eliminar Riesgo	Alta
20	Mostrar Riesgos	Alta
21	Adicionar Riesgo a proyecto	Alta
22	Eliminar Riesgo de proyecto	Alta
23	Adicionar Categoría	Alta
24	Modificar Categoría	Alta
25	Eliminar Categoría	Alta
26	Mostrar Categorías	Alta
27	Seleccionar técnica de análisis	Media
28	Iniciar técnica de análisis Modelo Lingüístico Virtual	Alta
29	Adicionar experto	Alta
30	Modificar Experto	Alta
31	Eliminar Experto	Alta
32	Mostrar Expertos	Alta
33	Asignar evaluación lingüística a los riesgos	Alta
34	Aplicar evaluación colectiva virtual de riesgos	Alta
35	Mostrar lista de riesgos priorizados	Alta
36	Iniciar técnica "Árboles de decisión"	Media
37	Adicionar decisión	Alta
38	Modificar decisión	Alta
39	Eliminar decisión	Alta
40	Mostrar decisiones	Alta

41	Adicionar Acontecimiento	Alta
42	Modificar Acontecimiento	Alta
43	Eliminar acontecimiento	Alta
44	Mostrar acontecimientos	Alta
45	Adicionar resultado	Alta
46	Modificar resultado	Alta
47	Eliminar resultado	Alta
48	Mostrar resultados	Alta
49	Asignar relaciones entre nodos	Alta
50	Obtener camino óptimo	Alta
51	Iniciar técnica de análisis Panel de expertos	Media
52	Calcular exposición al riesgo	Alta
53	Obtener exposición total de riesgos y promedio	Alta
54	Mostrar resultados	Alta
55	Adicionar rol	Alta
56	Modificar rol	Alta
57	Eliminar rol	Alta
58	Actualizar rol	Alta
59	Adicionar permiso	Alta
60	Eliminar permiso	Alta
61	Generar reporte por facultad	Baja
62	Generar reporte por centro	Baja
63	Generar reporte por proyecto	Baja
64	Generar reporte por técnica de análisis	Baja

Tabla 4. Relación de requisitos no funcionales del sistema.

	Requisitos no Funcionales(RNF)
RNF 1	Confiability
	El sistema debe ser preciso con la información que maneja y le proporciona al usuario. Haciendo énfasis en los resultados de los análisis que ejecutará, para evitar errores que puedan incidir negativamente en la toma de decisiones.
RNF 2	Seguridad
RNF 2.1	La seguridad de la Base de Datos será gestionada mediante la asignación de roles para mantener la integridad de los datos en función del nivel de acceso asignado
RNF 2.2	El sistema poseerá mecanismos de seguridad que garanticen los niveles de acceso de acuerdo al rol asignado por el usuario administrador.
RNF 3	Restricciones de diseño e implementación

RNF 3.1	Para la implementación del componente se utilizará el lenguaje de programación PHP
RNF 3.2	Se hará uso de la librería Extjs de JavaScript para la creación de los elementos visuales del sistema.
RNF 3.3	Como framework para PHP se utilizará la herramienta CodeIgniter
RNF 3.4	El IDE a utilizar será NetBeans 7.1
RNF 3.5	Para el modelado del sistema se hará uso de la herramienta CASE Visual Paradigm 8.0
RNF 4	Apariencia o interfaz externa
RNF 4.1	La interfaz del sistema deberá ser amigable a los usuarios finales. Incluirá solo la información y elementos necesarios que hagan intuitivo su uso. Deberá contar con una adecuada combinación de color y tipografía.
RNF 4.2	Los módulos se muestran con sus respectivos nombres.
RNF 5	Software(Cliente)
RNF 5.1	Navegador Web Mozilla Firefox o Google Chrome
RNF 5.2	Sistema Operativo Windows XP o GNU/Linux
RNF 5.3	Software(Servidor)
RNF 5.4	Sistema Operativo GNU/Linux, Windows XP o versión superior
RNF 5.5	Servidor web Apache versión 2.0 o superior
RNF 6	Hardware(Cliente)
RNF 6.1	Requerimientos Mínimos: Procesador Intel Pentium III, con 512 de Ram
RNF 6.2	Conexión de red
RNF 6.3	Hardware(Servidor)
RNF 6.4	Requerimientos Mínimos: Procesador Intel Pentium Dual-Core, con 2GB de Ram
RNF 6.5	Contar con al menos 20 GB de espacio libre en el servidor
RNF 6.6	Conexión de red

2.3. Identificación de los actores.

El sistema será utilizado por especialistas pertenecientes al Centro de Informatización Universitaria. Cuyos dos roles son: usuario y administrador. A continuación en la tabla 4 se explica los privilegios asociados a cada rol.

Tabla 5. Relación de usuarios del sistema.

Actor	Descripción
Usuario	Accede al sistema para hacer uso de las técnicas de análisis cuantitativo además de consultar los diferentes reportes.

Administrador	Es el encargado de realizar funciones administrativas: administrar usuarios, gestionar riesgos, proyectos y posee permisos para hacer uso de las técnicas de análisis cuantitativo y consultar los reportes.
---------------	--

2.4. Historias de Usuario (HU).

Las HU constituyen el artefacto empleado por XP para especificar los requisitos funcionales que poseerá la aplicación o sistema a implementar. De forma análoga pueden asociarse a los casos de uso, pertenecientes al Proceso Unificado de Desarrollo (RUP). En estas se reflejan las necesidades del usuario en un lenguaje claro y en pocas líneas de texto. También se les asigna un número identificativo, una prioridad en el negocio (alta, media, baja) y la iteración en la que se implementará. A continuación se describen algunas HU con alta prioridad para el sistema, para consultar el resto de las HU remitirse al [Anexo 1](#).

Tabla 6. HU Gestionar riesgo.

Historia de usuario.	
No.:6	Nombre: Gestionar Riesgo.
Usuarios: Administrador.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 1	Iteración asignada: 1
Descripción: Permite al administrador del sistema, adicionar nuevos riesgos así como eliminar y mostrar los diferentes riesgos registrados en el sistema. El sistema mostrará una interfaz en la cual introduce los datos (nombre, descripción y categoría). Cuando se ingresa un nuevo riesgo el sistema mostrará un listado con todos los riesgos y se permitirá actualizar los datos.	
Observaciones: No se permitirá introducir datos que ya existan en el sistema.	

Tabla 7. HU Gestionar expertos.

Historia de usuario.	
No.:12	Nombre: Gestionar expertos.
Usuarios: Administrador, Usuario.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 1	Iteración asignada: 1
<p>Descripción: Permite al administrador del sistema y al usuario, adicionar nuevos expertos así como eliminar y mostrar los diferentes expertos registrados en el sistema. El sistema mostrará una interfaz en la cual introduce los datos (nombre y descripción). Cuando se ingresa un nuevo experto el sistema mostrará un listado con todos los que han sido agregados. Se permitirá actualizar los datos introducidos.</p>	
<p>Observaciones: No se permitirá introducir datos que ya existan en el sistema.</p>	

Tabla 8. HU Asignar evaluación lingüística a los riesgos.

Historia de usuario.	
No.:13	Nombre: Asignar evaluación lingüística a los riesgos.
Usuarios: Administrador, Usuario.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 1	Iteración asignada: 1
<p>Descripción: La HU permite tanto al administrador del sistema como al usuario asignar las etiquetas lingüísticas virtuales (MA, BA, A, M, B, MB, NA) a los riesgos que se van a analizar. El sistema mostrará una interfaz que permita al usuario asignar las etiquetas a los valores de probabilidad e impacto para cada riesgo a analizar.</p>	
<p>Observaciones: Se permitirá actualizar las etiquetas que hayan sido asignadas.</p>	

Tabla 9. HU Aplicar evaluación colectiva virtual de riesgos.

Historia de usuario.	
No.:14	Nombre: Aplicar evaluación colectiva virtual de riesgos.
Usuarios: Administrador, Usuario.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 2	Iteración asignada: 1
Descripción: Permite tanto al administrador del sistema como al usuario, aplicar la técnica una vez definidos los expertos en el proceso además de que se hayan asignado las etiquetas virtuales sobre cada riesgo, emitido por los expertos. El usuario selecciona la opción "Evaluar" para obtener el resultado.	
Observaciones:	

Tabla 10. HU Mostrar lista de riesgos priorizados.

Historia de usuario.	
No.:15	Nombre: Mostrar lista de riesgos priorizados
Usuarios: Administrador, Usuario.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 1	Iteración asignada: 1
Descripción: Permite que al usuario obtener una lista de riesgos priorizados. El sistema mostrará una interfaz con aquellos riesgos que han sido priorizados luego de aplicar la técnica.	
Observaciones: Deben haber asignado los expertos así como su criterio para obtener este resultado.	

Tabla 11. HU Gestionar decisión.

Historia de usuario.	
No.:17	Nombre: Gestionar decisión.

Usuarios: Administrador, Usuario.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 1	Iteración asignada: 2
Descripción: Permite al administrador como al usuario del sistema, adicionar nuevas decisiones así como eliminar y mostrar las diferentes decisiones registradas en el sistema. El sistema mostrará una interfaz en la cual introduce los datos (etiqueta). Cuando se ingresa una nueva decisión, el sistema mostrará un listado con todas aquellas que han sido agregadas. Se permitirá actualizar los datos introducidos.	
Observaciones: No se permitirá introducir valores que ya existan en el sistema.	

Tabla 12. HU Gestionar acontecimiento.

Historia de usuario.	
No.:18	Nombre: Gestionar acontecimiento.
Usuarios: Administrador, Usuario.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 1	Iteración asignada: 2
Descripción: Permite al administrador y al usuario del sistema adicionar nuevos acontecimientos así como eliminar y mostrar los diferentes acontecimientos registradas en el sistema. El sistema mostrará una interfaz en la cual introduce los datos (etiqueta). Cuando se ingresa un nuevo acontecimiento, el sistema mostrará un listado con todos aquellos que han sido agregados. Se permitirá actualizar los datos introducidos.	
Observaciones: No se permitirá introducir valores duplicados.	

Tabla 13. HU Gestionar resultado.

Historia de usuario.	
No.:19	Nombre: Gestionar resultado.

Usuarios: Administrador, Usuario.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 1	Iteración asignada: 2
Descripción: Permite al administrador y al usuario del sistema adicionar nuevos resultados así como eliminar y mostrar los diferentes resultados registradas en el sistema. El sistema mostrará una interfaz en la cual introduce los datos (etiqueta, valor). Cuando se ingresa un nuevo resultado, el sistema mostrará un listado con todos aquellos que han sido agregados. Se permitirá actualizar los datos introducidos.	
Observaciones: No se permitirá introducir valores que ya existan en el sistema.	

Tabla 14. HU Obtener camino óptimo.

Historia de usuario.	
No.:21	Nombre: Obtener camino óptimo.
Usuarios: Administrador, Usuario.	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Medio
Puntos de estimación: 3	Iteración asignada: 2
Descripción: Permite al administrador y al usuario del sistema aplicar la técnica una vez definidos los elementos que componen el árbol. El usuario selecciona la opción "Evaluar" para obtener el resultado. El sistema muestra una interfaz en la que se refleja al camino óptimo a seguir en el árbol, así como el beneficio del camino y el costo de implementar la decisión.	
Observaciones: Deben haberse completado los elementos que componen al árbol.	

2.5. Plan de iteraciones.

El plan de iteraciones es un artefacto que definen los desarrolladores mediante una reunión formal. Estos deciden cómo y en qué orden realizar las tareas. Se confecciona sobre la base de las historias de usuario priorizadas o que poseen un mayor impacto sobre el negocio. Estas historias son divididas

en tareas con el objetivo de evitar complejidades para la implementación. Para esta investigación, el plan de iteraciones se proyectará de la siguiente forma:

Tabla 15. Relación de HU con tiempo estimado de implementación.

Iteración	Nro. HU	HU	Duración Estimada
1	1	Autenticar usuario	2 semanas
	2	Gestionar usuario	
	3	Gestionar facultad	
	4	Gestionar proyecto	
	5	Gestionar centro	
	6	Gestionar riesgo	
	7	Adicionar Riesgo a proyecto	
	8	Eliminar Riesgo de proyecto	
2	9	Gestionar categoría	3 semanas
	10	Seleccionar técnica de análisis	
	11	Iniciar técnica de análisis Modelo Lingüístico Virtual	
	12	Gestionar expertos	
	13	Asignar evaluación lingüística a los riesgos	
	14	Aplicar evaluación colectiva virtual de riesgos	
	15	Mostrar lista de riesgos priorizados	
	16	Iniciar técnica "Árboles de decisión"	
	17	Gestionar decisión	
	18	Gestionar acontecimiento	
	19	Gestionar resultado	
	20	Asignar relaciones entre nodos	
	21	Obtener camino óptimo	
3	22	Iniciar técnica de análisis Panel de expertos	2 semanas
	23	Calcular exposición al riesgo	
	24	Obtener exposición total de riesgos y promedio	
	25	Mostrar resultados	
	26	Gestionar rol	
	27	Adicionar permiso	
	28	Eliminar permiso	
	29	Generar reportes por facultad	
	30	Generar reportes por centro	
	31	Generar reportes por proyecto	
	32	Generar reporte por técnica de análisis	

2.6. Plan de entrega

A continuación se relaciona las HU por cada módulo que contiene la aplicación.

Tabla 16. HU abarcadas por módulos.

Módulo	Historia de Usuario	
Administración	Autenticar usuario	
	Gestionar usuario	
	Gestionar facultad	
	Gestionar proyecto	
	Gestionar centro	
	Gestionar riesgo	
	Adicionar Riesgo a proyecto	
	Eliminar Riesgo de proyecto	
	Gestionar categoría	
	Gestionar rol	
	Adicionar permiso	
		Eliminar permiso
	Análisis	Seleccionar técnica de análisis
Iniciar técnica de análisis Modelo Lingüístico Virtual		
Gestionar experto		
Asignar evaluación lingüística a los riesgos		
Aplicar evaluación colectiva virtual de riesgos		
Mostrar lista de riesgos priorizados		
Iniciar técnica "Árboles de decisión"		
Gestionar decisión		
Gestionar acontecimiento		
Gestionar resultado		
Asignar relaciones entre nodos		
Obtener camino óptimo		
Iniciar técnica de análisis Panel de expertos		
Calcular exposición al riesgo		
Obtener exposición total de riesgos y promedio		
Reportes	Mostrar resultados	
	Generar reportes por facultad	
	Generar reportes por centro	
	Generar reportes por proyecto	

A continuación en la Tabla 16 se muestran los planes de entrega de los módulos del sistema con la respectiva versión de los diferentes componentes.

Tabla 17. Relación de entrega de los módulos con sus versiones.

Módulo	Final 1ra Iterac.	Final 2da Iterac.	Final 3ra Iterac.
Administración	0.1	0.2	1.0
Análisis	0.1	0.2	1.0
Reportes	-	-	1.0

2.7. Prototipos de interfaz.

En el presente epígrafe se muestran algunos prototipos de interfaz.

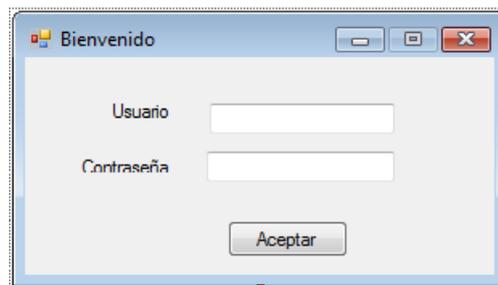


Figura 2. Prototipo de interfaz para la Autenticación del usuario.

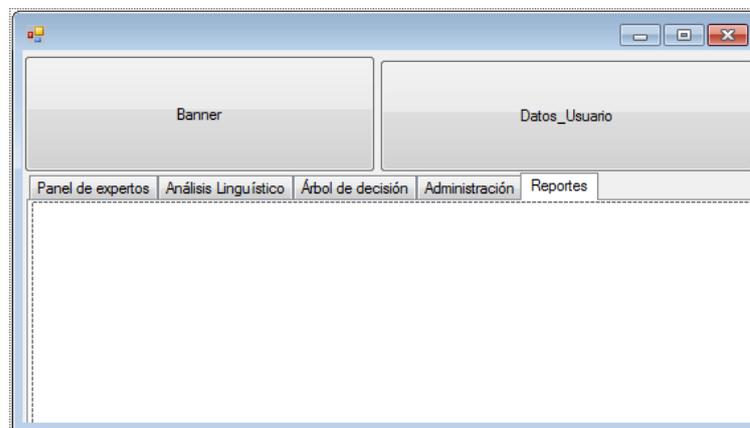


Figura 3. Prototipo de interfaz. Vista principal del sistema.

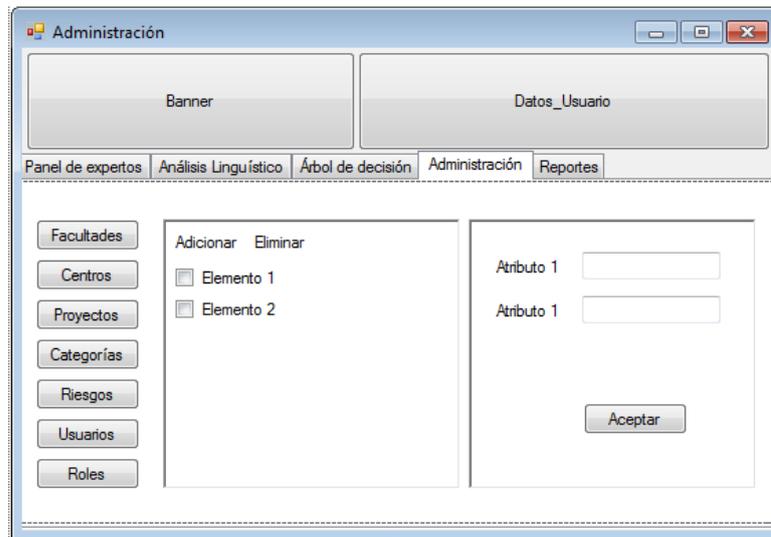


Figura 4. Prototipo de interfaz. Módulo de administración del sistema.

2.8. Conclusiones.

En este capítulo quedó descrito el sistema que se desea desarrollar. Se analizó el flujo actual de los procesos para entender las necesidades del cliente. En base a estos elementos se propuso la división del sistema en los diferentes módulos (administración, análisis y reportes) que lo compondrá. Se obtuvieron las funcionalidades con las que debe contar el sistema las cuales constituyen la base de la implementación del sistema. Se describió mediante un artefacto la relación de HU con la iteración correspondiente que serán implementadas y la estimación correspondiente en semanas. Se describieron además algunas HU las cuales juegan un papel fundamental en desarrollo del sistema, teniendo en cuenta que se quiere obtener un producto final de buena calidad.

CAPÍTULO 3: IMPLEMENTACIÓN.

En el presente capítulo se exponen diversos elementos de la implementación del sistema, como resultado de este proceso se obtienen las tarjetas CRC. Se define la arquitectura del sistema, los patrones de diseño y los estándares de codificación empleados. Se diseñan los casos de prueba para comprobar las diferentes funcionalidades del sistema.

3.1 Implementación del sistema.

Para la implementación del sistema se utilizó el framework CodeIgniter para el lenguaje de programación PHP el cual se basa en el patrón de arquitectura Modelo-Vista-Controlador que se divide en tres capas:

Modelo: Es una representación específica del dominio de la información sobre la cual funciona la aplicación. Es un componente totalmente independiente el cual se encarga de manejar los datos y sus transformaciones. Se encarga de obtener los datos almacenados en la base de datos para que sean posteriormente representados en la vista apropiada.

Vista: La vista representa el modelo en un formato adecuado para interactuar con el usuario. Transforma el modelo en una página web que permite al usuario interactuar con ella. Se encarga de la presentación visual de los datos captados por el modelo para después mostrarlos al usuario. Interactúa con el modelo y muestra los datos al usuario.

Controlador: Los controladores reciben la entrada o sea los eventos producidos por el usuario mediante el uso del teclado o el mouse a través de las vistas. Los eventos son traducidos para servir las demandas del modelo o las vistas. Cuando se realiza algún cambio, entra en acción bien puede ser por cambios en el modelo o en la vista.

La aplicación de este patrón de arquitectura presenta varias ventajas:

- Reduce el esfuerzo de programación necesario en la implementación de sistemas.
- El modelo, la vista y los controladores se tratan como entidades separadas lo cual hace que un cambio producido en el modelo se refleje automáticamente en las vistas.
- Se pueden construir varias vistas sin necesidad de modificar el modelo.

Primeramente el usuario interactúa con la interfaz (Vista) de alguna forma. El controlador recibe la notificación de la acción solicitada y gestiona el evento. El controlador accede al modelo actualizándolo o buscando la información requerida. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario y mostrar los datos del modelo para generar la interfaz apropiada.

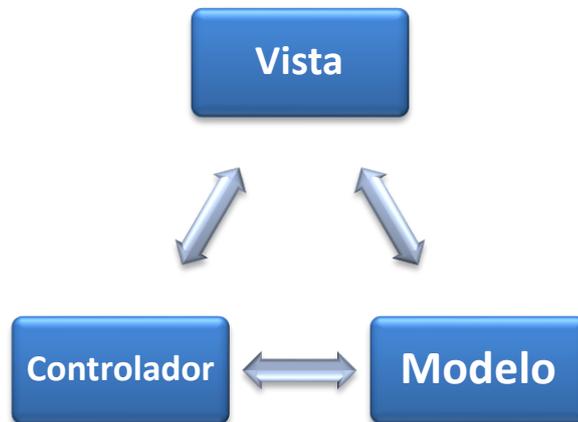


Figura 5. Diagrama Modelo – Vista – Controlador.

3.2. Empleo de metáforas.

Una metáfora es una historia que el equipo puede emitir en conjunto con el cliente acerca de cómo debería funcionar el sistema, de esta forma el equipo obtiene una visión común sobre lo que desea alcanzar. Esto ayuda a obtener la solución más simple que pueda ser implementada y probada posteriormente. Es por esto que el sistema se define mediante un conjunto de metáforas compartidas entre el cliente y el equipo de desarrollo. A continuación se presentan algunas metáforas que se emplearon como base para la implementación del sistema.

- La interfaz del sistema debería ser como las aplicaciones de escritorio.
- Las opciones del sistema deben estar en diferentes apartados o ventanas.
- El sistema mostrará datos en forma de gráficos.

El sistema debe definirse de forma simple y sin entrar en grandes detalles para que el diseñador realice un primer entregable del sistema. El diagrama de paquetes proporciona un modelo de los diferentes elementos que componen al sistema así como sus relaciones, además de permitir un diseño inicial sencillo del sistema. La Figura 4 muestra el diagrama de paquetes asociado al módulo de administración del sistema.

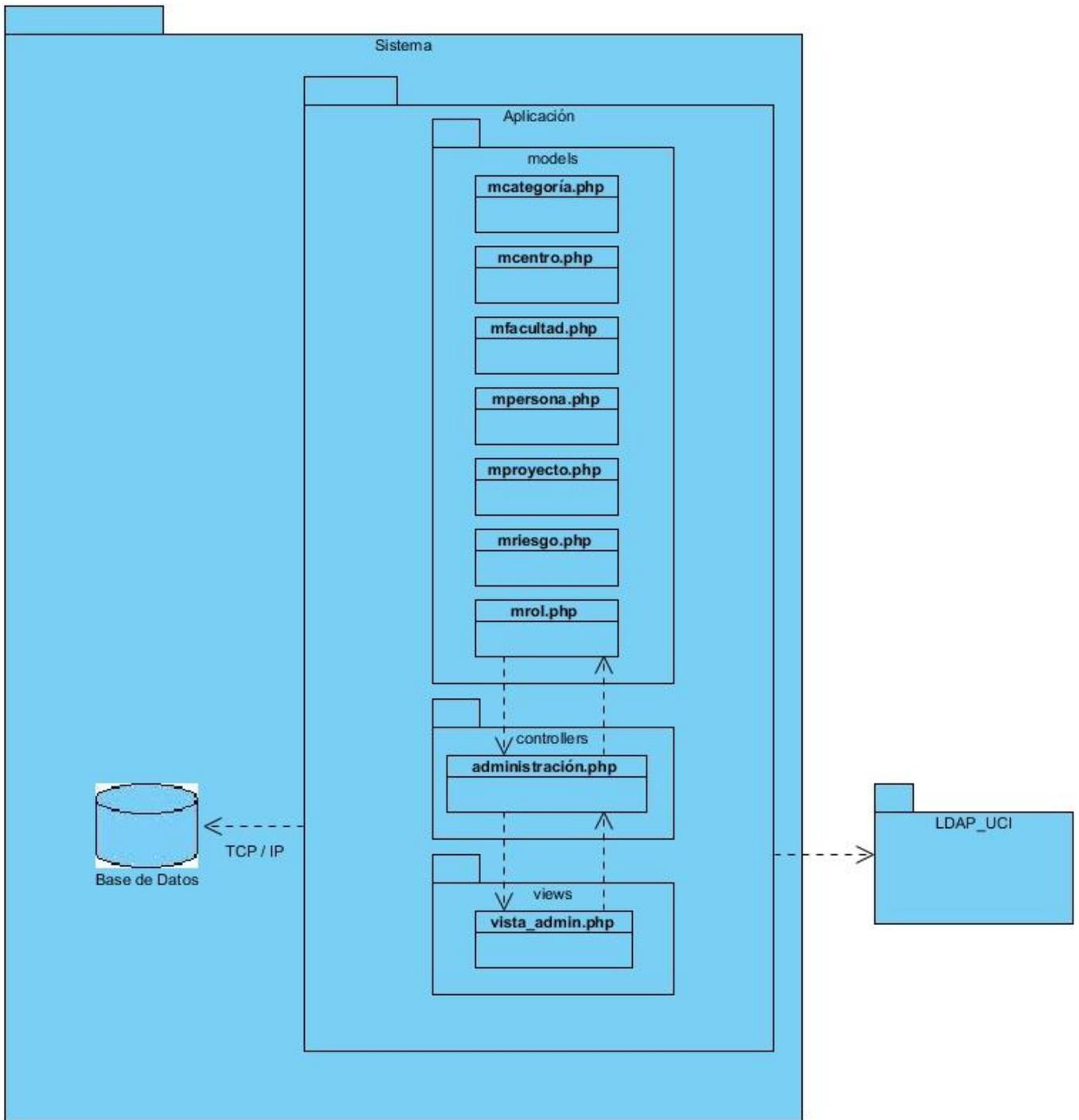


Figura 6. Diagrama de paquetes (Administración)

3.3 Tarjetas CRC.

Las tarjetas CRC constituyen una metodología creada por Kent Beck que define la relación Clase – Responsabilidad – Colaboración y ayuda a pensar en términos de objetos. En primer lugar deben identificarse las clases que compondrán al sistema, para comenzar este proceso se utiliza la técnica de tormenta de ideas (Brainstorm) mediante la cual el equipo de desarrollo aporta ideas para localizar todas las posibles clases. Una vez identificadas todas las clases se procede a refinarlas o sea dejar solo aquellas que sean básicas para el sistema que son las que poseen responsabilidades y participan activamente en el sistema. Las tarjetas CRC se crean a partir de la lista de clases básicas identificadas y constituyen una primera aproximación a los objetos que luego se van a utilizar en el desarrollo del sistema. El resto de las tarjetas CRC puede ser consultado en el [Anexo 2](#).

Tabla 18. Tarjeta CRC clase facultad.

facultad	
Funcionalidades	Colaboración
Mostrar facultades	
Adicionar facultad	
Eliminar facultad	
Actualizar facultad	

Tabla 19. Tarjeta CRC clase centro.

centro	
Funcionalidades	Colaboración
Mostrar centros	facultad
Adicionar centro	
Eliminar centro	
Actualizar centro	

Tabla 20. Tarjeta CRC clase persona.

persona	
Funcionalidades	Colaboración
Mostrar personas	proyecto

Adicionar persona	rol
Eliminar persona	

Tabla 21. Tarjeta CRC clase proyecto.

proyecto	
Funcionalidades	Colaboración
Mostrar proyectos	centro
Adicionar proyecto	
Eliminar proyecto	
Actualizar proyecto	

Tabla 22. Tarjeta CRC clase riesgo.

riesgo	
Funcionalidades	Colaboración
Mostrar riesgos	categoría
Adicionar riesgo	
Eliminar riesgo	
Actualizar riesgo	

Tabla 23. Tarjeta CRC clase rol.

rol	
Funcionalidades	Colaboración
Mostrar roles	
Adicionar rol	
Eliminar rol	
Actualizar rol	

3.4 Patrones de diseño.

Patrones GRASP (Patrones para la asignación de responsabilidad).

Experto: Se basa en asignar una responsabilidad al experto en información, se refiere a la clase que cuenta con la información o atributos necesarios para cumplir con la responsabilidad o responsabilidades para la que fue implementada. De esta forma se conserva la encapsulación y se

logra una mayor cohesión. En el sistema existen clases que cuentan con la información necesaria para cumplir con sus responsabilidades, tal es el caso de Facultad, Rol y Riesgo. A continuación se muestra la clase “Facultad”.

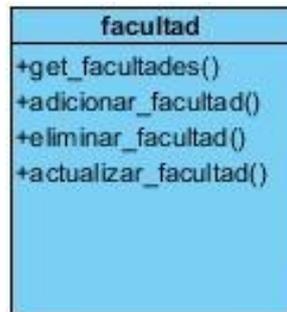


Figura 7. Clase facultad.

Alta Cohesión: Se refiere a cuán relacionadas y enfocadas están las relaciones de una clase, por lo que una clase con Alta cohesión tiene responsabilidades estrechamente relacionadas y poco complejas. Tal es el caso en el sistema de la clase Riesgo la cual está formada por varias funcionalidades que están estrechamente relacionadas y colabora con la clase Categoría para realizar diferentes operaciones. A continuación se muestran estas dos clases.

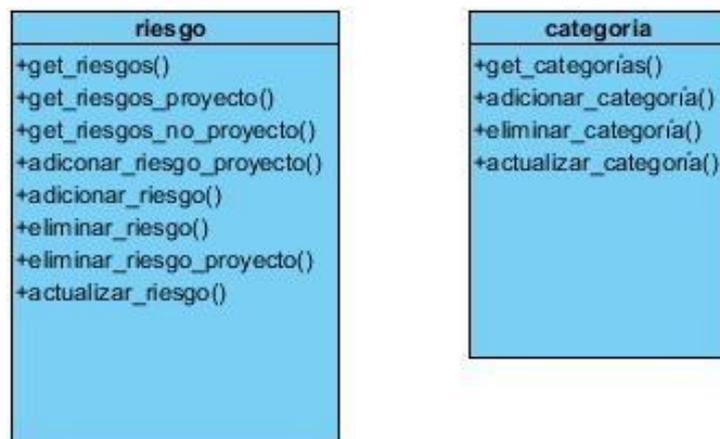


Figura 8. Clases riesgo y categoría.

Bajo Acoplamiento: Este patrón se refiere a cuán fuertemente está relacionada una clase con otra, de forma tal que si se produce una modificación entre sus relaciones, se afecten en menor medida el resto de las clases, por lo que una clase con bajo acoplamiento no depende de muchas otras para realizar su función en el negocio. Ejemplo de ello lo constituyen las clases Persona y Proyecto que a pesar de estar relacionadas no tienen una fuerte dependencia.

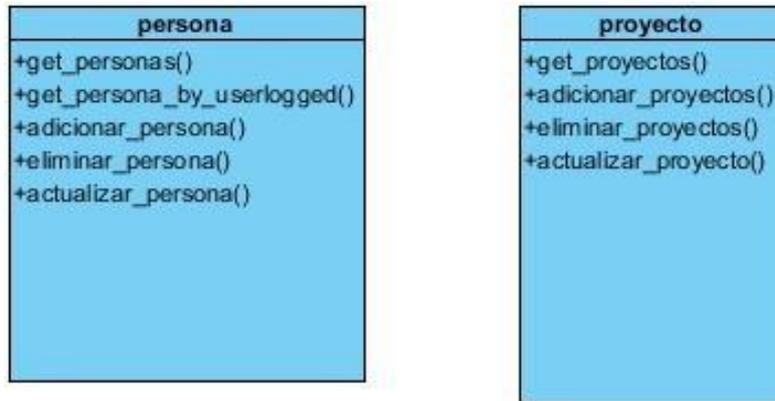


Figura 9. Clases persona y proyecto.

Controlador: Este patrón sugiere el uso de clases controladoras para separar la lógica de negocios de la capa de presentación. Las peticiones en las aplicaciones web son manejadas por clases de este tipo, por lo que no existe una sola clase controladora que se encargue de manejar todas las peticiones. El framework CodeIgniter destina un directorio específico para guardar las clases que realizarán estas funciones. A continuación se muestran las clases “persona” y “proyecto”.

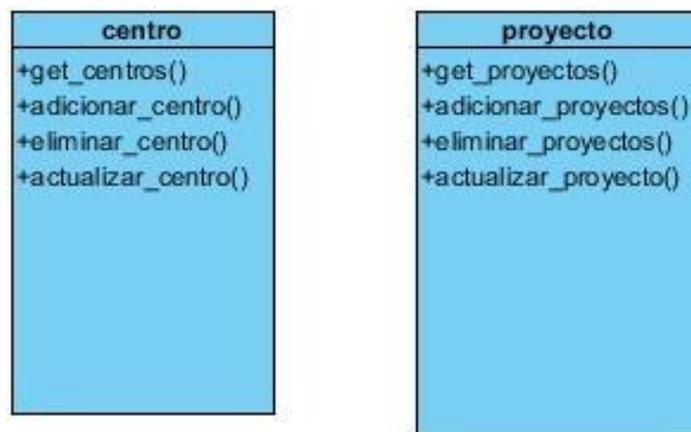


Figura 10. Clases centro y proyecto.

Patrón instancia única.

La motivación de este patrón es que una cierta clase tenga una única instancia y esta sea accesible a través de una variable global. El framework CodeIgniter para realizar el acceso a los datos almacenados en una Base de Datos utiliza la clase “database” la cual posee parámetros configurables para adaptarlos a nuestras necesidades. Para realizar la consulta a los datos se realiza mediante la variable \$db.

3.5 Diseño de la Base de Datos.

Representa los datos usados por la aplicación, es utilizado para la persistencia de la información necesaria. Para el presente sistema se identificaron las siguientes tablas, así como sus relaciones y atributos.

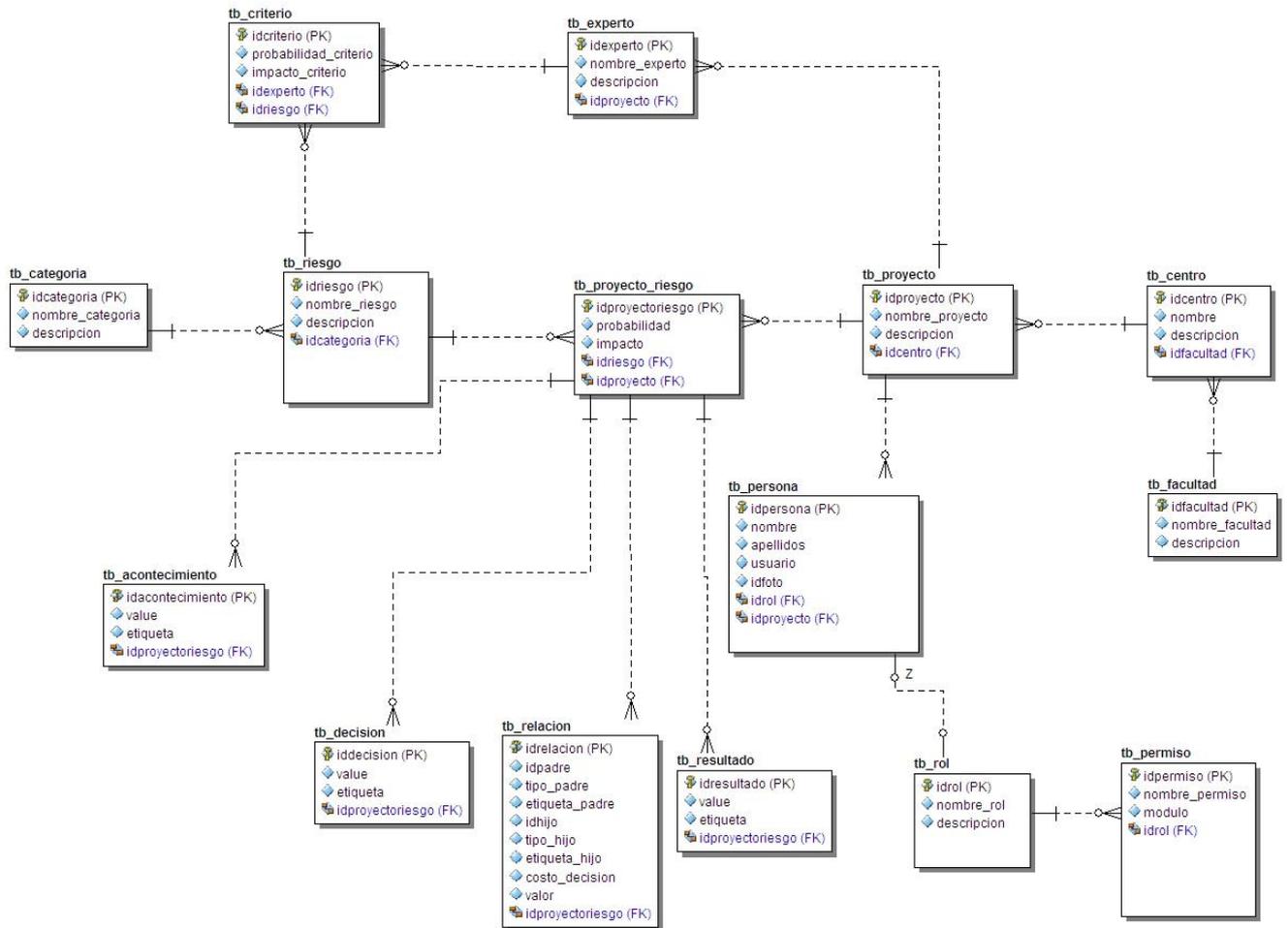


Figura 11. Modelo físico de datos.

Descripción de las tablas.

A continuación se describen algunas tablas identificadas (riesgo, decisión, acontecimiento, resultado, experto y criterio) para la persistencia de los datos. Para consultar la descripción de las restantes tablas, remitirse al [Anexo 3](#).

Tabla riesgo

Tabla 24. Descripción de la tabla riesgo

Nombre: riesgo		
Descripción: En esta tabla se recoge toda la información referente a los riesgos registrados en el sistema		
Atributo	Tipo	Descripción
Idriesgo	serial	Atributo que identifica al riesgo
nombre_riesgo	varchar	En este atributo se representa al riesgo
descripcion	text	Se brinda una descripción acerca del riesgo
idcategoria	Integer	Atributo que identifica la categoría del riesgo

Tabla decisión

Tabla 25. Descripción de la tabla decisión.

Nombre: decisión		
Descripción: En esta tabla se recoge toda la información referente a las decisiones pertenecientes al árbol de decisión		
Atributo	Tipo	Descripción
iddecision	serial	Atributo que identifica a una decisión
idriesgoproyecto	Integer	Identificador para un proyecto en un riesgo
value	double precision	Valor numérico asociado a una decisión
etiqueta	varchar	Atributo de texto para representar la decisión

Tabla acontecimiento

Tabla 26. Descripción de la tabla acontecimiento.

Nombre: acontecimiento		
Descripción: En esta tabla se recoge toda la información referente a los acontecimientos pertenecientes al árbol de decisión		
Atributo	Tipo	Descripción
idacontecimiento		Atributo que identifica a un acontecimiento
idriesgoproyecto	Integer	Identificador para un proyecto en un riesgo
value	double precision	Valor numérico asociado a un acontecimiento
etiqueta	varchar	Atributo de texto para representar el acontecimiento

Tabla resultado

Tabla 27. Descripción de la tabla resultado.

Nombre: resultado		
Descripción: En esta tabla se recoge toda la información referente a los resultados que pueden		

existir en el árbol de decisión		
Atributo	Tipo	Descripción
idresultado	serial	Atributo que identifica a un resultado
idriesgoproyecto	Integer	Identificador para un proyecto en un riesgo
value	double precision	Valor numérico asociado a un acontecimiento
etiqueta	varchar	Atributo de texto para representar el acontecimiento

Tabla experto

Tabla 28. Descripción de la tabla experto.

Nombre: experto		
Descripción: En esta tabla se recoge toda la información referente a los expertos que emiten criterios sobre los riesgos		
Atributo	Tipo	Descripción
idexperto	Serial	Atributo que identifica a un experto
nombre_experto	varchar	Nombre que identifica al experto en el sistema
descripción	Text	Descripción de los datos de un experto
idproyecto	Serial	Identificador para un proyecto en un riesgo

Tabla criterio

Tabla 29. Descripción de la tabla criterio.

Nombre: criterio		
Descripción: En esta tabla se recoge toda la información referente al criterio emitido por una persona sobre un riesgo.		
Atributo	Tipo	Descripción
idcriterio	serial	Atributo que identifica a un criterio
probabilidad_criterio	text	Atributo para la probabilidad sobre un riesgo
Impacto_criterio	text	Atributo para el impacto sobre un riesgo
idexperto	integer	Atributo para identificar a un experto

3.6 Estándares de codificación.

Debido a que la metodología XP considera al código como propiedad colectiva, es necesario tener en cuenta los siguientes estándares para llegar a un entendimiento colectivo por parte del equipo de trabajo. Se tuvieron en cuenta los siguientes convenios que se describen a continuación:

- Las clases controladoras están representadas mediante una sola palabra (facultad, riesgo, rol).

- Los formularios comienzan con la palabra “form” seguido del guión bajo y el nombre de la clase a el cual hace referencia (form_centro).
- En el caso de las clases que pertenecen al Modelo comienzan con la letra “m” (mcentro, mproyecto).
- El nombre de las clases que pertenecen a las vistas del sistema comienza con la palabra “view” seguido del guión bajo y el nombre de la vista a la cual hace referencia (view_admin).
- El nombre de las funciones comienza con letra minúscula. En caso de que el nombre contenga más de una palabra, estas estarán separadas por un guion bajo.

3.7 Diagrama de componentes.

Los diagramas de componentes representan la estructura del modelo de implementación dividido en subsistemas y a su vez relaciones entre estos elementos. Este diagrama agrupa los elementos de software que componen al sistema (archivos, paquetes, bibliotecas, etc). El framework CodeIgniter implementa el patrón MVC, por lo que las clases que componen al sistema se localizan en distintos directorios o carpetas. En este caso el sistema se encuentra compuesto por una serie de clases controladoras que se encargan de recibir las peticiones del usuario, además de contener las funciones (adicionar, eliminar, mostrar y actualizar) para desempeñar su papel en el sistema. Estas clases a su vez están estrechamente relacionadas con sus respectivos modelos, que contienen las funcionalidades para interactuar con la base de datos. Las vistas interactúan directamente con las clases controladoras y estas fueron desarrolladas empleando los diferentes elementos visuales (formularios, gráficos, paneles, etc...) que ofrece la librería ExtJS. A continuación se presenta en la Figura 6 los elementos que componen al sistema, evidenciando el uso del patrón MVC y el framework CodeIgniter.

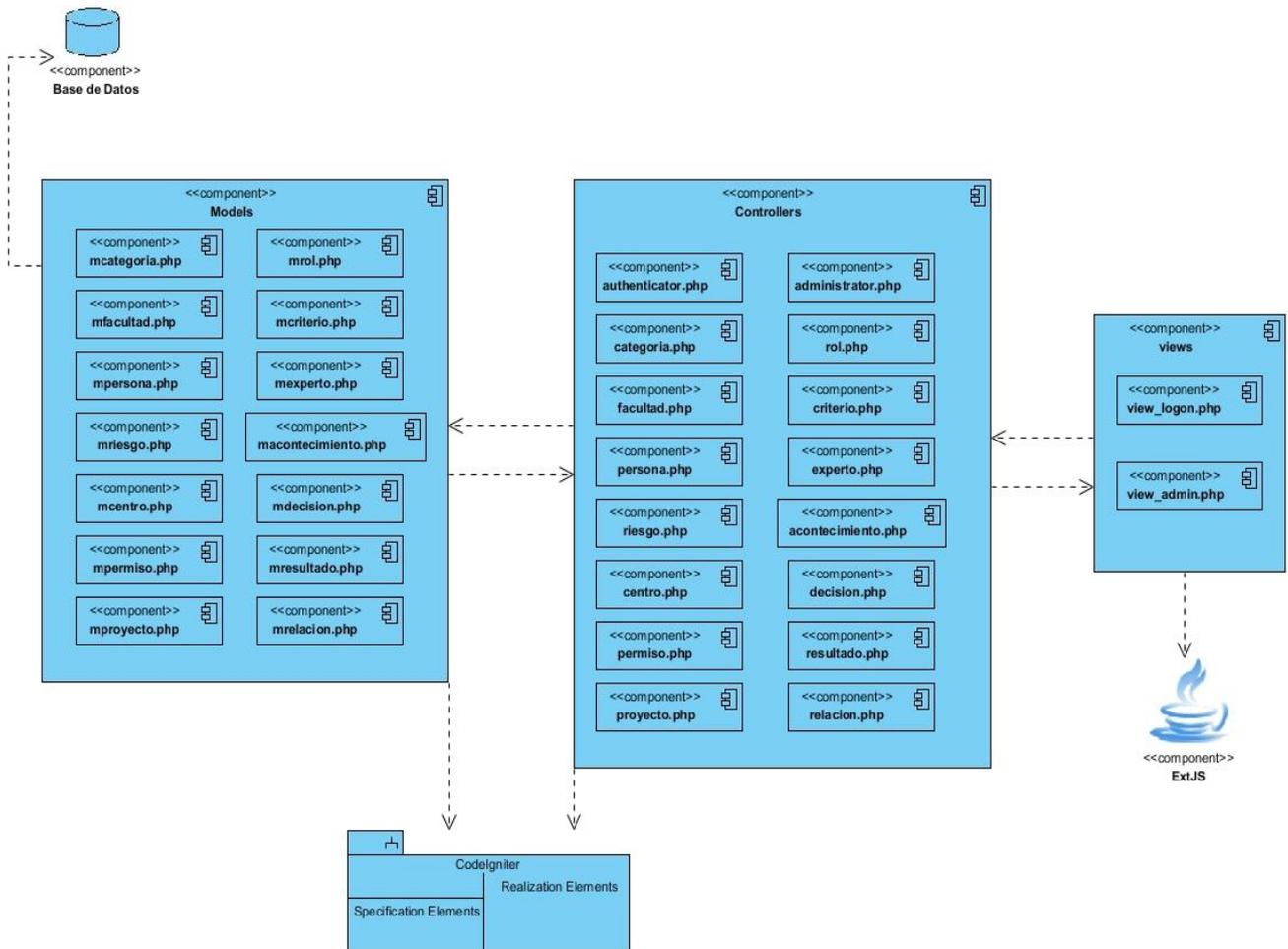


Figura 12. Diagrama de componentes del sistema.

3.8. Ejemplos de código implementado.

A continuación se presentan fragmentos de código correspondientes al sistema implementado.

En el caso del árbol de decisión se utilizaron una serie de condiciones para verificar su validez y de esta forma proceder a obtener el curso de acción óptimo.

Se parte de una decisión primaria, por lo que debe existir al menos un nodo de este tipo, aunque el árbol puede contener otros nodos de decisión.

```

if (get class($tree) == "Decision" && count($tree->getHijos()) < 1) {
    $status = false;
    $visit ['bad'] = $tree->getEtiqueta ();
    $visit['error_message'] = "Las Decisiones deben tener al menos un hijo"; }

```

Los nodos de tipo acontecimiento tienen al menos dos hijos y la suma de los valores de probabilidad para cada caso tiene que ser igual a uno. Esto se valida con el siguiente fragmento de código.

```
if (get_class($tree) == "Acontecimiento" && count($tree->getHijos()) < 2) {
    $status = false;
    $visit['bad'] = $tree->getEtiqueta();
    $visit['error_message'] = "Los Acontecimientos deben tener al menos dos hijos";
}
if (get_class($tree) == "Acontecimiento" && $tree->getSumaProbHijos() === 1) {
    $status = false;
    $visit['bad'] = $tree->getEtiqueta();
    $visit['error_message'] = "Los hijos de un Acontecimiento deben sumar 1 en su probabilidad";
}
```

Los nodos de tipo resultado no pueden tener asociados a ningún hijo.

```
if (get_class($tree) == "Resultado") {
    $status = false;
    $visit['bad'] = $tree->getEtiqueta();
    $visit['error_message'] = "Los resultados no pueden tener hijos";}
```

En el caso del análisis lingüístico virtual, hay que validar que todos los expertos hayan emitido su criterio sobre los riesgos analizados. En caso de que la cantidad de criterios sea diferente de la cantidad de expertos multiplicado por la cantidad de riesgos, no se procede a evaluar, esto significa que no se han establecido todos los criterios sobre los riesgos, luego se construyen las etiquetas con sus respectivos números identificativos. Esto se logra con el siguiente fragmento de código.

```
if (count($data_criterios) == count($data_expertos) * count($data_riesgos)){
    $escala = array ('MA' => 1, 'BA' => 2, 'A' => 3, 'M' => 4, 'B' => 5, 'MB' => 6, 'NA' => 7);
```

El valor de la preferencia colectiva se obtiene de la multiplicación de la escala seleccionada para los valores de probabilidad e impacto, este proceso es iterativo, por lo que se realizará hasta que se obtenga la suma de todos los criterios emitidos por los expertos. Esto se logra mediante el siguiente fragmento de código.

```
for ($r = 0; $r < count($data_riesgos); $r++) {
    $riesgo = $data_riesgos[$r];
    $data_criterios_riesgos = $this->mcriterio->obtenerTodosCriteriosByRiesgo ($idproyecto,
    $riesgo['idriesgo']);
```

```

$suma = 0;
for ($k = 0; $k < count($data_criterios_riesgos); $k++)
{
    $suma += $escala[$data_criterios_riesgos[$k]['probabilidad_criterio']] *
$escala[$data_criterios_riesgos[$k]['impacto_criterio']];
}

```

El valor de la variable \$suma es dividido entre la cantidad de riesgos, de acuerdo a la fórmula de evaluación colectiva. Los resultados se guardan en la variable \$result y son ordenados de forma ascendente mediante la función sort, para finalmente invertir los valores empleando la función array_reverse, obteniéndose los valores ordenados descendientemente para priorizar la atención a los riesgos.

3.9 Diagrama de despliegue.

El diagrama de despliegue muestra interactúan los diferentes nodos que componen al sistema. En este caso el usuario realiza una petición mediante la PC_Cliente al servidor web que ejecuta las funciones que tiene implementado el sistema, accede a la base de datos ya sea para buscar, adicionar o eliminar datos existentes y envía la respuesta de vuelta al cliente. En la Figura No.7 se muestra este diagrama.

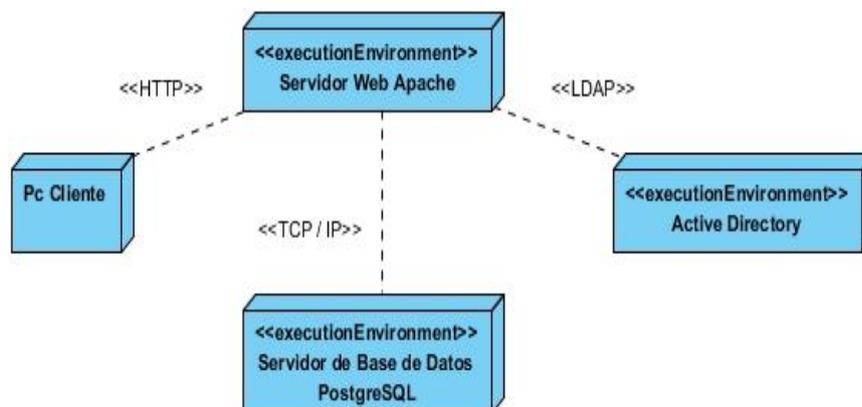


Figura No.13 Diagrama de despliegue.

Nombre del nodo: Pc_Cliente.

Este nodo hace referencia a cualquier estación de trabajo (PC_Cliente) desde la cual se accede al sistema vía web. El servidor es el que se encarga de procesar todas las peticiones del usuario, por lo que los requisitos de hardware en este caso son mínimos.

Nombre del nodo: Servidor_Web_Apache

Este nodo hace referencia al servidor de aplicaciones web Apache en el cual se aloja la aplicación, el cual se encarga de manejar todas las peticiones de los clientes.

Nombre del nodo: Servidor de Base de Datos.

Este nodo hace referencia al servidor PostgreSQL 9.1 que dará soporte a la información necesaria del sistema.

Nombre del nodo: LDAP UCI.

El sistema utiliza este nodo para la autenticación de los usuarios.

3.10 Interfaces.

A continuación se muestran algunas interfaces que componen al sistema. La Figura 5 muestra las opciones que compone al panel de administración. La Figura 6 muestra los proyectos y riesgos que se han agregado al sistema. Estas opciones cuentan con las funcionalidades de agregar y eliminar elementos. El resto de las interfaces de este apartado (Facultades, Personas, Roles y Permisos) cuentan con una interfaz similar.



Figura 14. Panel de administración del sistema.

Panel de gestión

Adicionar Proyecto Eliminar Proyecto

<input type="checkbox"/>	Proyecto	Centro	Descripción
<input type="checkbox"/>	1 Hospitales	CESIM	Proyecto para venezuela
<input type="checkbox"/>	2 Alas-His	CESIM	Alas-His
<input type="checkbox"/>	3 PSGT: Solucione...	DATEC	dasdasdads
<input type="checkbox"/>	4 CEDAE	CISED	Proyecto para Bolivia
<input type="checkbox"/>	5 SAS	CESIM	Proyecto para mexico

Adicionar Proyecto

Nombre:

Descripción:

Centro:

Aceptar

Figura 15. Formulario para la gestión de proyectos.

Panel de gestión

Adicionar Riesgo Eliminar Riesgo

<input type="checkbox"/>	Riesgo	Descripción
<input type="checkbox"/>	1 Hardware fallido	Puede ser que el hardware falle en el transcurso del proyecto
<input type="checkbox"/>	2 Lluvia	La lluvia impide que los trabajadores asistan al laboratorio
<input type="checkbox"/>	3 Requisitos mal le...	Los requisitos no fueron correctamente identificados por el analista

Adicionar Riesgo

Nombre:

Descripción:

Categoría:

Aceptar

Figura 16. Interfaz para la gestión de riesgos.

3.11 Conclusiones.

En este capítulo se definió la arquitectura que soporta el sistema. Se han representado diversos artefactos que definen al sistema y sirvieron como base para su posterior implementación. Se confeccionó la base de datos que almacena la información relevante del sistema, además de la descripción y uso de los patrones de diseño empleados.

CAPÍTULO 4: VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

4.1 Pruebas.

Cuando se quiere contar con un producto con la calidad requerida es necesario aplicar pruebas que corroboren su correcto funcionamiento a lo largo del proceso de desarrollo o una vez finalizado este. La validación de un sistema se refiere al proceso de evaluación de sus componentes y las funcionalidades identificadas en la fase de análisis. Las pruebas se ejecutan dentro de este proceso y con circunstancias previamente identificadas para posteriormente documentar los resultados obtenidos. De forma particular un caso de prueba busca comprobar una pequeña funcionalidad del sistema y observar su comportamiento.

4.2 Casos de Prueba.

Una parte importante de la metodología XP son las pruebas, mediante las cuales los desarrolladores prueban constantemente el sistema para detectar las posibles fallas que puedan ocurrir. Esta práctica contribuye no solo a elevar la calidad del sistema que se quiere construir, además de que el proceso de implementación se realiza sobre funcionalidades con un correcto funcionamiento. Los casos de prueba se ejecutan sobre las historias de usuario, el principal objetivo es comprobar que las HU han sido correctamente implementadas. Las técnicas de caja negra se realizan sobre la interfaz del sistema y sin conocer la lógica de este. El sistema recibe un conjunto de entradas y produce salidas, a partir de las cuales se comprobará si el comportamiento es el esperado o no. A continuación se presentan un conjunto de casos de prueba destinados a comprobar el funcionamiento del sistema.

Iteración 1: La presente iteración de pruebas tiene como objetivo comprobar la correcta implementación de varias HU y detectar posibles no conformidades, la relación de HU involucradas en el proceso es la siguiente: Gestionar riesgo, Adicionar riesgo a proyecto, Gestionar expertos, Asignar evaluación lingüística a los riesgos y mostrar lista de riesgos priorizados.

Tabla 30. Caso de prueba Autenticar Usuario.

Caso de Prueba	
Código: HU1_P1	Historia de Usuario: Autenticar Usuario
Nombre: Autenticar Usuario	
Descripción: Prueba de funcionalidad para la autenticación de usuario.	

Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su respectivo rol.
Pasos de ejecución: El usuario procede a la inserción de los datos necesarios especificando su usuario y contraseña y presiona sobre el botón “Aceptar”.
Resultado esperado: El usuario accede correctamente al sistema con el nivel de autorización de acuerdo al rol asignado.
Evaluación de la prueba: Satisfactoria.

Tabla 31. Caso de prueba Insertar riesgo.

Caso de Prueba	
Código: HU6_P2	Historia de Usuario: Gestionar riesgo
Nombre: Insertar riesgo	
Descripción: Prueba de funcionalidad para la inserción de un riesgo.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario procede a la inserción de los datos necesarios especificando el identificador del riesgo, su descripción y la categoría a asignar y se ejecuta la acción.	
Resultado esperado: El riesgo es insertado correctamente.	
Evaluación de la prueba: Satisfactoria.	

Tabla 32. Caso de prueba Mostrar riesgos.

Caso de Prueba	
Código: HU6_P3	Historia de Usuario: Gestionar riesgo
Nombre: Mostrar riesgos	
Descripción: Prueba de funcionalidad para mostrar los riesgos registrados en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario verifica que el riesgo insertado se muestre en una lista con aquellos que hayan sido insertados previamente.	
Resultado esperado: El riesgo es mostrado correctamente.	
Evaluación de la prueba: Satisfactoria.	

Tabla 33. Caso de prueba Eliminar riesgo.

Caso de Prueba	
Código: HU6_P4	Historia de Usuario: Gestionar riesgo
Nombre: Eliminar riesgo	
Descripción: Prueba de funcionalidad para eliminar un riesgo registrado en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario selecciona un riesgo y procede a ejecutar la acción eliminar riesgo.	
Resultado esperado: El riesgo es eliminado correctamente.	
Evaluación de la prueba: Satisfactoria.	

Tabla 34. Caso de prueba Modificar riesgo.

Caso de Prueba	
Código: HU6_P5	Historia de Usuario: Gestionar riesgo
Nombre: Modificar riesgo	
Descripción: Prueba de funcionalidad para modificar un riesgo registrado en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario selecciona un riesgo y procede a ejecutar la acción modificar riesgo.	
Resultado esperado: El riesgo es modificado correctamente.	
Evaluación de la prueba: Satisfactoria.	

Tabla 35. Caso de prueba Asignar un riesgo a un proyecto.

Caso de Prueba	
Código: HU7_P6	Historia de Usuario: Adicionar riesgo a proyecto
Nombre: Asignar un riesgo a un proyecto.	
Descripción: Prueba de funcionalidad para asignar un riesgo a un proyecto registrado en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	

Pasos de ejecución: El usuario selecciona un proyecto y seguidamente ejecuta la acción asignar riesgo a proyecto. Procede a introducir los valores de probabilidad e impacto y ejecuta la acción de asignación.
Resultado esperado: El riesgo es asignado correctamente a un proyecto.
Evaluación de la prueba: Satisfactoria.

Tabla 36. Caso de prueba Insertar experto.

Caso de Prueba	
Código: HU12_P7	Historia de Usuario: Gestionar expertos
Nombre: Insertar experto.	
Descripción: Prueba de funcionalidad para insertar un nuevo experto en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario procede a la inserción de los datos necesarios (nombre, descripción) y ejecuta la acción.	
Resultado esperado: El experto es insertado correctamente en el sistema.	
Evaluación de la prueba: Satisfactoria.	

Tabla 37. Caso de prueba Mostrar expertos.

Caso de Prueba	
Código: HU12_P8	Historia de Usuario: Gestionar expertos
Nombre: Mostrar expertos.	
Descripción: Prueba de funcionalidad para mostrar los expertos registrados en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario verifica que el experto insertado se muestre en una lista con aquellos que hayan sido insertados previamente.	
Resultado esperado: El experto es mostrado correctamente en el sistema.	
Evaluación de la prueba: Satisfactoria.	

Tabla 38. Caso de prueba Eliminar experto.

Caso de Prueba	
Código: HU12_P9	Historia de Usuario: Gestionar expertos
Nombre: Eliminar experto.	
Descripción: Prueba de funcionalidad para eliminar un experto del sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario selecciona un experto y procede a ejecutar la acción eliminar experto.	
Resultado esperado: El experto es eliminado correctamente del sistema.	
Evaluación de la prueba: Satisfactoria.	

Tabla 39. Caso de prueba Modificar experto.

Caso de Prueba	
Código: HU12_P10	Historia de Usuario: Gestionar expertos
Nombre: Modificar experto.	
Descripción: Prueba de funcionalidad para modificar un experto en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario selecciona un experto y procede a ejecutar la acción modificar.	
Resultado esperado: El experto es modificado correctamente del sistema.	
Evaluación de la prueba: Satisfactoria.	

Tabla 40. Caso de prueba Asignar etiqueta virtual a cada riesgo.

Caso de Prueba	
Código: HU13_P11	Historia de Usuario: Asignar evaluación lingüística a los riesgos
Nombre: Asignar etiqueta virtual a cada riesgo	
Descripción: Prueba de funcionalidad para asignar valores de probabilidad e impacto sobre etiquetas lingüísticas virtuales a cada riesgo, emitido por cada experto.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	

Pasos de ejecución: El usuario selecciona un experto y procede a asignar a cada riesgo un conjunto de etiquetas lingüísticas virtuales por cada riesgo que se analizará. . El usuario procede a ejecutar la acción de asignación.
Resultado esperado: Se asignan correctamente las etiquetas para cada riesgo emitidas por cada experto.
Evaluación de la prueba: Satisfactoria.

Tabla 41. Caso de prueba Mostrar lista de riesgos priorizados.

Caso de Prueba	
Código: HU15_P12	Historia de Usuario: Mostrar lista de riesgos priorizados.
Nombre: Evaluación obtenida de la aplicación del método Modelo Lingüístico Virtual	
Descripción: Prueba de funcionalidad para comprobar que la lista de riesgos priorizados sea correcta.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente y debe haber registrado los expertos participantes en el proceso así como su criterio sobre los riesgos empleando las etiquetas virtuales.	
Pasos de ejecución: El usuario ejecuta la acción de evaluación y se muestra la lista con los riesgos priorizados así como su categoría asignada.	
Resultado esperado: Se muestran correctamente priorizados los riesgos de acuerdo a la fórmula de evaluación colectiva de riesgos.	
Evaluación de la prueba: Satisfactoria.	

Iteración 2: La presente iteración de pruebas tiene como objetivo comprobar la correcta implementación de varias HU y detectar posibles no conformidades, la relación de HU involucradas en el proceso es la siguiente: Gestionar decisión, Gestionar acontecimiento, Gestionar resultado, Asignar relaciones entre nodos, Obtener camino óptimo.

Tabla 42. Caso de prueba Insertar decisión.

Caso de Prueba	
Código: HU17_P13	Historia de Usuario: Gestionar decisión
Nombre: Insertar decisión.	
Descripción: Prueba de funcionalidad para insertar un nuevo acontecimiento en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario procede a la inserción de los datos necesarios y ejecuta la acción.	

Resultado esperado: El acontecimiento es adicionado correctamente al sistema.
Evaluación de la prueba: Satisfactoria.

Tabla 43. Caso de prueba Mostrar decisión.

Caso de Prueba	
Código: HU17_P14	Historia de Usuario: Gestionar decisión
Nombre: Mostrar decisión.	
Descripción: Prueba de funcionalidad para mostrar las decisiones insertadas en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario verifica que la decisión insertada se muestre en una lista con aquellas que hayan sido insertados previamente.	
Resultado esperado: La decisión es correctamente mostrada en una lista.	
Evaluación de la prueba: Satisfactoria.	

Tabla 44. Caso de prueba Eliminar decisión.

Caso de Prueba	
Código: HU17_P15	Historia de Usuario: Gestionar decisión
Nombre: Eliminar decisión.	
Descripción: Prueba de funcionalidad para eliminar una decisión insertada en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario selecciona una decisión y procede a ejecutar la acción eliminar decisión.	
Resultado esperado: La decisión es correctamente eliminada del sistema.	
Evaluación de la prueba: Satisfactoria.	

Tabla 45. Caso de prueba Modificar decisión.

Caso de Prueba	
Código: HU17_P16	Historia de Usuario: Gestionar decisión
Nombre: Modificar decisión.	

Descripción: Prueba de funcionalidad para eliminar una decisión insertada en el sistema.
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.
Pasos de ejecución: El usuario selecciona una decisión y procede a ejecutar la acción modificar decisión.
Resultado esperado: La decisión es correctamente modificada.
Evaluación de la prueba: Satisfactoria.

Tabla 46. Caso de prueba insertar acontecimiento.

Caso de Prueba	
Código: HU18_P17	Historia de Usuario: Gestionar acontecimiento
Nombre: Insertar acontecimiento.	
Descripción: Prueba de funcionalidad para insertar un acontecimiento en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario procede a la inserción de los datos necesarios y ejecuta la acción.	
Resultado esperado: El acontecimiento es correctamente insertado en el sistema	
Evaluación de la prueba: Satisfactoria.	

Tabla 47. Caso de prueba Mostrar acontecimiento.

Caso de Prueba	
Código: HU18_P18	Historia de Usuario: Gestionar acontecimiento
Nombre: Mostrar acontecimiento.	
Descripción: Prueba de funcionalidad para mostrar los acontecimientos insertados en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario verifica que el acontecimiento insertado se muestre en una lista con aquellos que hayan sido insertados previamente.	
Resultado esperado: El acontecimiento es correctamente mostrado en una lista.	
Evaluación de la prueba: Satisfactoria.	

Tabla 48. Caso de prueba Eliminar acontecimiento.

Caso de Prueba	
Código: HU18_P19	Historia de Usuario: Gestionar acontecimiento.
Nombre: Eliminar acontecimiento.	
Descripción: Prueba de funcionalidad para eliminar un acontecimiento del sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario selecciona un acontecimiento y procede a ejecutar la acción eliminar acontecimiento.	
Resultado esperado: El acontecimiento es correctamente mostrado en una lista.	
Evaluación de la prueba: Satisfactoria.	

Tabla 49. Caso de prueba Modificar acontecimiento.

Caso de Prueba	
Código: HU18_P20	Historia de Usuario: Gestionar acontecimiento.
Nombre: Modificar acontecimiento.	
Descripción: Prueba de funcionalidad para modificar un acontecimiento del sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario selecciona un acontecimiento y procede a ejecutar la acción modificar acontecimiento.	
Resultado esperado: El acontecimiento es correctamente modificado.	
Evaluación de la prueba: Satisfactoria.	

Tabla 50. Caso de prueba insertar resultado.

Caso de Prueba	
Código: HU19_P21	Historia de Usuario: Gestionar resultado.
Nombre: Insertar resultado.	
Descripción: Prueba de funcionalidad para insertar un resultado en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario procede a la inserción de los datos necesarios y ejecuta la acción.	

Resultado esperado: El resultado es correctamente insertado.
Evaluación de la prueba: Satisfactoria.

Tabla 51. Caso de prueba Mostrar resultados.

Caso de Prueba	
Código: HU19_P22	Historia de Usuario: Gestionar resultado.
Nombre: Mostrar resultados.	
Descripción: Prueba de funcionalidad para mostrar los resultados registrados en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario verifica que el resultado insertado se muestre en una lista con aquellos que hayan sido insertados previamente.	
Resultado esperado: El resultado es correctamente insertado.	
Evaluación de la prueba: Satisfactoria.	

Tabla 52. Caso de prueba Eliminar resultado.

Caso de Prueba	
Código: HU19_P23	Historia de Usuario: Gestionar resultado.
Nombre: Eliminar resultado.	
Descripción: Prueba de funcionalidad para eliminar un resultado registrado en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario selecciona un resultado y procede a ejecutar la acción eliminar resultado.	
Resultado esperado: El resultado es correctamente insertado.	
Evaluación de la prueba: Satisfactoria.	

Tabla 53. Caso de prueba Modificar resultado.

Caso de Prueba	
Código: HU19_P24	Historia de Usuario: Gestionar resultado.
Nombre: Modificar resultado.	

Descripción: Prueba de funcionalidad para modificar un resultado registrado en el sistema.
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.
Pasos de ejecución: El usuario selecciona un resultado y procede a ejecutar la acción modificar resultado.
Resultado esperado: El resultado es correctamente modificado.
Evaluación de la prueba: Satisfactoria.

Tabla 54. Caso de prueba Asignar relaciones entre nodos.

Caso de Prueba	
Código: HU20_P25	Historia de Usuario: Asignar relaciones entre nodos.
Nombre: Asignación de relaciones entre nodos	
Descripción: Prueba de funcionalidad asignar relaciones entre los nodos que componen el árbol de decisión.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario procede a seleccionar el tipo de asociación que desea establecer entre los diferentes elementos.	
Resultado esperado: La relación es correctamente asignada y mostrada.	
Evaluación de la prueba: Satisfactoria.	

Tabla 55. Caso de prueba Obtener camino óptimo.

Caso de Prueba	
Código: HU21_P26	Historia de Usuario: Obtener camino óptimo
Nombre: Camino óptimo.	
Descripción: Prueba de funcionalidad obtener camino óptimo.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario procede a evaluar el árbol de decisión y se ejecuta la acción de evaluación.	
Resultado esperado: Se muestra correctamente el camino óptimo a seguir de acuerdo al algoritmo de resolución del árbol.	
Evaluación de la prueba: Satisfactoria.	

Iteración 3: La presente iteración de pruebas tiene como objetivo comprobar la correcta implementación de varias HU y detectar posibles no conformidades, la relación de HU involucradas en el proceso es la siguiente: Calcular exposición al riesgo y Gestionar rol.

Tabla 56. Caso de prueba Calcular exposición al riesgo.

Caso de Prueba	
Código: HU23_P27	Historia de Usuario: Calcular exposición al riesgo
Nombre: Exposición al riesgo	
Descripción: Prueba de funcionalidad para obtener el resultado de la exposición al riesgo	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario procede seleccionar la técnica "Panel de expertos", selecciona el proyecto del cual desea obtener los valores de exposición al riesgo	
Resultado esperado: Se muestra correctamente los valores de exposición al riesgo por cada proyecto que ha sido seleccionado.	
Evaluación de la prueba: Satisfactoria.	

Tabla 57. Caso de prueba insertar rol.

Caso de Prueba	
Código: HU26_P27	Historia de Usuario: Gestionar rol
Nombre: Insertar rol	
Descripción: Prueba de funcionalidad para insertar un rol en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario procede a la inserción de los datos necesarios y ejecuta la acción.	
Resultado esperado: Se muestra correctamente los valores de exposición al riesgo por cada proyecto que ha sido seleccionado.	
Evaluación de la prueba: Satisfactoria.	

Tabla 58. Caso de prueba Mostrar rol.

Caso de Prueba	
Código: HU26_P28	Historia de Usuario: Gestionar rol
Nombre: Mostrar rol	
Descripción: Prueba de funcionalidad para mostrar los roles registrados en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario verifica que el rol insertado en el sistema se muestre en una lista con aquellos que han sido insertados previamente.	
Resultado esperado: El rol es correctamente mostrado en una lista.	
Evaluación de la prueba: Satisfactoria.	

Tabla 59. Caso de prueba Eliminar rol.

Caso de Prueba	
Código: HU26_P29	Historia de Usuario: Gestionar rol
Nombre: Eliminar rol	

Descripción: Prueba de funcionalidad para eliminar un rol registrado en el sistema.
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.
Pasos de ejecución: El usuario selecciona un rol y procede a ejecutar la acción “eliminar rol”
Resultado esperado: El rol es correctamente eliminado del sistema.
Evaluación de la prueba: Satisfactoria.

Tabla 60. Caso de prueba Modificar rol.

Caso de Prueba	
Código: HU26_P30	Historia de Usuario: Gestionar rol
Nombre: Modificar rol	
Descripción: Prueba de funcionalidad para modificar un rol registrado en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario selecciona un rol y procede a ejecutar la acción “modificar rol”	
Resultado esperado: El rol es correctamente modificado en el sistema.	
Evaluación de la prueba: Satisfactoria.	

4.3 Resultados obtenidos.

En la metodología XP, la aplicación de pruebas se realiza de forma iterativa para comprobar la correcta implementación de las historias de usuario. Se aplicaron un total de 31 casos de prueba que se distribuyeron en 3 iteraciones. A continuación se refleja en la Figura 16, la cantidad total de funcionalidades probadas y las no conformidades obtenidas por cada iteración.

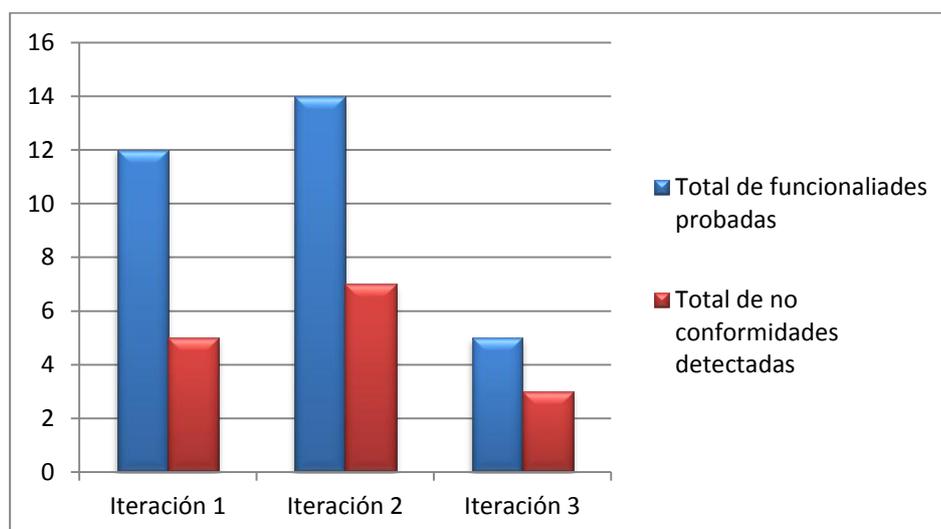


Figura 17. Totales de funcionalidades probadas y no conformidades detectadas por iteración.

A continuación se desglosa para cada iteración el tipo de error detectado:

Iteración 1.

Se comprobaron un total de 12 funcionalidades, detectándose un total de 5 no conformidades. Los errores detectados se desglosan de la siguiente forma:

- 1 error ortográfico.
- 3 errores de validaciones de campos.
- 1 error de interfaz.

Iteración 2.

Se comprobaron un total de 14 funcionalidades, detectándose un total de 7 no conformidades. Los errores detectados se desglosan de la siguiente forma:

- 1 error ortográfico.
- 5 errores de validación.
- 1 error de interfaz.

Iteración 3.

Se comprobaron un total de 5 funcionalidades, detectándose un total de 3 no conformidades. Los errores detectados se desglosan de la siguiente forma:

- 1 error ortográfico.
- 2 errores de validación.

CONCLUSIONES

Con la realización de este trabajo se ha cumplido con los objetivos propuestos. Se llegó además a las siguientes conclusiones:

Durante el desarrollo de la investigación se realizó un estudio de diversos procesos para conocer su criterio acerca del análisis cuantitativo de riesgos, así como diversas soluciones existentes para llevar a cabo esta tarea. Se demostró la necesidad de desarrollar un sistema que fuera capaz de automatizar este proceso para dar solución a la problemática planteada.

La aplicación se desarrolló siguiendo las directrices de la metodología XP y se identificaron las funcionalidades del sistema que sirvieron como base para la implementación. Este proceso se llevó a cabo con las herramientas seleccionadas. Como parte del proceso de implementación se obtuvo un sistema capaz de brindar a los usuarios las técnicas para analizar riesgos que pueden ser potencialmente peligrosos para los proyecto.

Al sistema obtenido se le aplicaron pruebas para comprobar su correcto funcionamiento y de esta forma comprobar que cumple con las expectativas del cliente.

RECOMENDACIONES

Como resultado del proceso de investigación e implementación de la solución propuesta han surgido ideas que son recomendables tener en cuenta para un posterior perfeccionamiento del sistema.

- ✓ Continuar el desarrollo del sistema con el objetivo de incorporar nuevas funcionalidades que contribuyan a su perfeccionamiento.
- ✓ Incluir una funcionalidad al sistema que permita dar seguimiento a los riesgos analizados.

REFERENCIAS BIBLIOGRÁFICAS

1. **Mercados y Marcas IT.** [En línea] 2010. [Citado el: 08 de 01 de 2013.] http://www.colombianproductions.com/mymit/joom1515/index.php?option=com_content&view=article&id=153:estudio-de-emc-proyector-gran-crecimiento-de-lainformacion-en-10-anos&catid=38:noticias&Itemid=55.
2. **Alfonso Sánchez, Ileana.** [En línea] 2001. [Citado el: 13 de 01 de 2013.] http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352001000300007. ISSN 1561-2880.
3. *Riesgo & incertidumbre en proyectos de desarrollo de software.* **Pérez Moya, Osiris.** 1, 10 de Febrero de 2011, Serie Científica de la Universidad de las Ciencias Informáticas, Vol. 4, pág. 12. ISSN.
4. *Definiciones acerca del riesgo y sus implicaciones.* **Echemendía Tocabens, Belkis .** 3, La Habana : s.n., 2011, Vol. 49. ISSN 1561-3003.
5. **Cuní, David.** Aplicaciones Empresariales. [En línea] 28 de 03 de 2012. [Citado el: 13 de 03 de 2013.] <http://www.aplicacionesempresariales.com/desventajas-de-microsoft-proyector.html>.
6. **zumbado Mora, Edward.** *Propuesta de herramienta para la gestión de riesgos en proyectos informáticos en Dinámica Consultores Internacional S.A.* [Documento] San José, Costa Rica : s.n., 2009.
7. **Oracle.** Primavera Enterprise Project Portfolio Management. [En línea] [Citado el: 2013 de Marzo de 15.] <http://www.oracle.com/lad/products/applications/primavera/overview/index.html>.
8. **Gedpro.** expertos en gestión de proyectos. [En línea] [Citado el: 2013 de Marzo de 15.] <http://www.gedpro.com/Software/RiskyProject.aspx>.
9. **Palisade Corporation.** Características nuevas en @Risk 5.0. [En línea] [Citado el: 2013 de Marzo de 15.] <http://www.palisade-lta.com/risk/caracteristicasnuevas.asp>.
10. **Pérez Moya, Osiris.** *Proceso para gestionar riesgos en los proyectos de desarrollo de software del Centro de Informatización Universitaria.* La Habana : s.n., 2013. pág. 47.
11. **Heredía Ruiz, Javier, Álvarez Almanza, Lilián y Linares Pons, Naryana.** *Comparación y tendencias entre metodologías ágiles y formales. Metodología utilizada en el Centro de Informatización para la Gestión de Entidades.* [Documento] La Habana : s.n., 2011.
12. **Fowler, Martin.** *UML Distilled: A Brief Guide to the Standard Object Modeling Language.* 3. s.l. : ADDISON-WESLEY, 2003. ISBN: 0-321-19368-7.
13. **Alarcón, Andrea y Sandoval, Erika.** *Herramientas Case para ingeniería de Requisitos.* [Documento] 2008.
14. **Cabrera González, Lianet y Pompa Torres, Enrique Roberto.** *Extensión de Visual Paradigm for UML para el desarrollo dirigido por modelos de aplicaciones de gestión de información.* [Documento] La Habana : s.n., 2012.
15. **Fernández Luna, Juan Manuel.** Breve descripción de los principales entornos integrados. [En línea] Septiembre de 2006. [Citado el: 20 de Marzo de 2013.] http://leo.ugr.es/J2ME/INTRO/intro_11.htm.
16. **López, Juan .** Netbeans 7.3. [En línea] 28 de Febrero de 2013. [Citado el: 20 de Marzo de 2013.] <http://www.topeinfo.es/noticias/netbeans-7-3/>.

17. **Ramírez, Mariano.** Que es Codeigniter y porque usarlo como framework de desarrollo. . [En línea] 20 de Febrero de 2011. [Citado el: 01 de Abril de 2013.] <http://sosinformatico.blogspot.com/2011/02/que-es-codeigniter-y-porque-usarlo-como.html>.
18. **Heredia Santos, Herminio** . [En línea] 23 de Mayo de 2001. [Citado el: 2013 de Marzo de 20.] <http://www.maestrosdelweb.com/editorial/phpintro/>.
19. **Sánchez Rosa, Juan Eladio.** Desarrollo en Web-Blog sobre desarrollo de aplicaciones web en Java, Python y JavaScript. [En línea] 22 de Octubre de 2008. [Citado el: 01 de Abril de 2013.] <http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/>.
20. **Ciberaula.** Una introducción a Apache. [En línea] 2010. [Citado el: 20 de Marzo de 2013.] http://linux.ciberaula.com/articulo/linux_apache_intro.
21. **Dataprix.** Características de PostgreSQL. [En línea] [Citado el: 20 de Marzo de 2013.] <http://www.dataprix.com/12-prestaciones>.

Bibliografía

Tapia Moreno, Franciso Javier. *Estadística Aplicada a las Licenciaturas: Administración, Contaduría e Informática Administrativa.* [Documento] Sonora : s.n., 2012.

Bonet, Adelina. [En línea] 12 de Junio de 2012. [Citado el: 15 de 01 de 2013.] <http://calidadtic.blogspot.com/2012/06/2-herramientas-para-la-toma-de.html>.

Amaro Calderón, Sarah Dámaris y Valverde Rebaza, Jorge Carlos. *Metodologías Ágiles.* 2007.
Rumbaugh, James, Jacobson, Ivar y Booch, Grady. *The Unified Modeling Language.* s.l. : ADDISON-WESLEY, 1999.

Gosling, James y McGilton , Henry. Sun Microsystems. [En línea] Mayo de 1996. [Citado el: 08 de 01 de 2013.] <http://www.oracle.com/technetwork/java/intro-141743.html>.

Teleinformatics Technology Group. Eclipse Process Framework. [En línea] [Citado el: 24 de 01 de 2013.] http://epf.eclipse.org/wikis/openupsp/openup_basic/deliveryprocesses/openup_basic_process_0uyGoMlgEdmt3adZL5Dmdw_desc.html.

Oracle Corporation and/or its affiliates. Portal del IDE Java de Código Abierto. [En línea] 2012. [Citado el: 28 de 01 de 2013.] <http://netbeans.org/features/ide/index.html>.

Sommerville, Ian. *Ingeniería del Software, 7ma edición.* s.l. : Addison Wesley, 2005. ISBN: 84-7829-074-5.

SoftExpert. SE Risk Gestión de Riesgos y Controles. [En línea] [Citado el: 2013 de Marzo de 15.] <http://www.softexpert.es/gestion-riesgos-controles.php>.

J Date, C. *Introducción a los Sistemas de Bases de Datos.*

Heredia Santos, Herminio . Maestros del Web. [En línea] 23 de Mayo de 2001. [Citado el: 2013 de Marzo de 20.] <http://www.maestrosdelweb.com/editorial/phpintro/>.

Carrero, Angel. Programación en Castellano. [En línea] 2013. [Citado el: 20 de Marzo de 2013.] http://www.programacion.com/articulo/funciones_y_caracteristicas_de_php_que_todo_programador_deberia_de_conocer_509.

Oracle Corporation and/or its affiliates. Netbeans IDE Features. *PHP Development.* [En línea] 2013. [Citado el: 20 de Marzo de 2013.] <http://netbeans.org/features/php/>.

Equipo ExtJS México. ExtJs Nuestra comunidad en español. [En línea] 27 de Noviembre de 2011. [Citado el: 31 de Marzo de 2013.] <http://www.extjs.mx/2011/11/27/post-con-cursos-generales/>.

Rodríguez Domínguez, Rosa María. *Un Nuevo Modelo para Procesos de Computación con Palabras en Toma de Decisión Lingüística.* Jaén : s.n., 2010.

Cárdenas, Karina. Ingeniería de Software. *Administración de riesgos.* [En línea] 16 de Marzo de 2011. [Citado el: 2013 de Abril de 10.] <http://sw-ufps.blogspot.com/2011/05/administracion-de-riesgos.html>.

Project Management Institute. [En línea] 2013. [Citado el: 2013 de Abril de 10.] <http://marketplace.pmi.org/Pages/ProductDetail.aspx?GMPProduct=00101186201>.

Universidad Virtual Anáhuac. [En línea] [Citado el: 2013 de Abril de 10.] <http://uva.anahuac.mx/content/catalogo/diplanes/modulos/mod2/simulacion.htm>.

Fiorito, Fabián . Universidad del CEMA. [En línea] [Citado el: 10 de Abril de 2013.] http://www.ucema.edu.ar/u/ffiorito/Workshop_Simulacion_CEMA_2009_Teoria.pdf.

Sánchez Barreiro, Pablo . *Ingeniería del Software II.* [Documento]

Morales Díaz, Leonel Vinicio. [En línea] 2008. [Citado el: 22 de Abril de 2013.] <http://ingenieriasimple.com/problemas/EjemploArbolesDecision.pdf>.

Felipe Valdés, Pilar, y otros, y otros. *Guía de estudio Asignatura: Investigación de Operaciones.* Ciudad Habana : s.n., 2008. pág. 21.

Universidad Veracruzana. Universo. *el periódico de los universitarios.* [En línea] 11 de Junio de 2011. [Citado el: 26 de Abril de 2013.] http://www.uv.mx/universo/486/infgral/infgral_15.html.

Un acercamiento a la trazabilidad en el desarrollo ágil de software. **González Ferro, Osmany.** 2, Habana : Ediciones Futuro, 28 de Septiembre de 2011, Revista Cubana de Ciencias Informáticas, Vol. 5, pág. 13. ISSN: 1994-1536.

Retos en la gestión de los riesgos en proyectos de software. **Zulueta Véliz, Yeleny, Hernández González, Anaisa y Despaigne Herrera , Eder.** 3-4, Habana : s.n., Julio-Diciembre de 2008, Revista Cubana de Ciencias Informáticas, Vol. 2, pág. 34.

Métricas de riesgos para los proyectos productivos de la Universidad de las Ciencias Informáticas. **Hernández Díaz, Neysis, Cuza García Rachel Seijó, Betsy y Seijó, Rachel.** 2, La Habana : Ediciones Futuro, 10 de Septiembre de 2012, Revista Cubana de Ciencias Informáticas, Vol. 6, pág. 12. ISSN: 1994-1536.

Una Aplicación del Método de Monte Carlo en el Análisis de Riesgo de Proyectos: Su automatización a través de una planilla de cálculo. **Périsse, Marcelo Claudio y Pepe, María Laura.** 04, Buenos Aires : s.n., 11 de Agosto de 2006, Técnica Administrativa, Vol. 05. ISSN 1666-1680.

Pressman, Roger. *Ingeniería del software, Un enfoque práctico.* 6ta. s.l. : McGraw-Hill.

Rafael, Juan. Modelo Vista Controlador. [En línea] 22 de Julio de 2008. [Citado el: 03 de Mayo de 2013.] <http://www.neleste.com/modelo-vista-controlador/>.

Patrón Modelo Vista Controlador. **Díaz González, Yanette y Fernández Romero, Yenisleidy.** [ed.] Departamento de Telemática del Instituto Superior Politécnico José Antonio Echeverría. 1, Enero-Abril de 2012, Revista Digital de las tecnologías de la información y las comunicaciones, Vol. 11. ISSN 1729-3804.

Hassan, Samer. Tarjetas CRC. [En línea] 2013. [Citado el: 04 de Mayo de 2013.] <https://cv2.sim.ucm.es/moodle/mod/resource/view.php?id=497139>.

Ellis Lab. Guía del Usuario de CodeIgniter Versión 2.1.3. [En línea] [Citado el: 03 de Mayo de 2013.] http://escodeigniter.com/guia_usuario/database/active_record.html.

García Luis, Luis Miguel. Patrón DAO y Patrón Active Record. [En línea] 12 de Junio de 2012. [Citado el: 2013 de Mayo de 30.] <http://unpocodejava.wordpress.com/2012/06/12/patron-dao-y-patron-active-record/>.

Moreno Vozmediano, Alfredo. Usando el patrón Active Record con CodeIgniter. [En línea] 16 de Enero de 2013. [Citado el: 30 de Abril de 2013.] <http://iesceliacf.org/blog/?p=46>.

Khalili, Mehdi. ORM anti-patterns - Part 1: Active Record. [En línea] 09 de Diciembre de 2010. [Citado el: 30 de Abril de 2013.] <http://www.mehdi-khalili.com/orm-anti-patterns-part-1-active-record>.

Code Igniter - Guía del Usuario en Español . 2011.

Beck, Kent . *Extreme Programming Explained*. First Edition. 1999. pág. 224. ISBN: 0201616416.

Achour, Mehdi , Betz , Friedhelm y Dovgal , Antony . *Manual de PHP*. [ed.] Philip Olson . 2010.

The PostgreSQL Global Development Group. PostgreSQL 9.1.9 Documentation. [En línea] 2013. [Citado el: 20 de mayo de 2013.] <http://www.postgresql.org/docs/9.1/static/tutorial-table.html>.

López, Félix. Parte II El lenguaje SQL. *Creación de tablas*. [En línea] 2013. [Citado el: 20 de Mayo de 2013.] http://pgsqtutorial.readthedocs.org/en/9.1.0/part_ii/creating_tables.html.

Rodríguez Domínguez, Rosa María. *Un nuevo modelo para procesos de Computación con palabras en Toma de Decisión Lingüística*. Departamento de Informática, Universidad de Jaén. Jaén : s.n., 2010. págs. 19-34, Diploma de estudios avanzados.