

Universidad de las Ciencias Informáticas

Facultad 3



Propuesta de Modelo para el Análisis Cuantitativo de Riesgos basado en técnicas de Softcomputing.

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

Autores: Danelys Brito González

Maylen Bon Pérez.

Tutor: Dr.C Pedro Y. Piñero Pérez.

Ing. Henrik Pestano Pino

MSc. Yadenis Piñero Pérez

Junio 2009

“Mientras que es inútil intentar eliminar el riesgo y cuestionable el poder minimizarlo, es esencial que los riesgos que se tomen sean los riesgos adecuados.”

Peter Drucker

Agradecimientos

A nuestro tutor Pedro, por sus consejos, paciencia y colaboración.

A Henrik por su preocupación y estar siempre pendiente.

A nuestros familiares por todo su apoyo y fe en nosotras.

A nuestras amistades por soportarnos durante todos estos años.

A René Lazo por su complicidad incondicional.

A Henry Raúl por auxiliarnos en los cortes.

A Rachel Olivera Hernández por su ayuda en la implementación.

A la UCI por formarnos como profesionales.

A todos los que de una manera u otra contribuyeron con el desarrollo del presente trabajo.

De Danelys:

A mi madre, por siempre estar ahí y apoyarme, y porque gracias a ti estoy aquí. Por apoyarme en la lucha de mis propios sueños.

A mi abuela Virgen, mi segunda madre, y a toda mi familia.

A Maylén, mi compañera de tesis y fiel amiga.

A Hirán, por apoyarme y quererme parte de este tiempo.

A los profes Pascual, Yoandris, Yoán, Águeda, por socorrerme a lo largo de la carrera.

A todos aquellos que en algún momento preguntaron ¿Cómo va la tesis?

A todos los que han contribuido con la materialización de este sueño.

De Maylén:

A mis padres por ser mi apoyo y por darme todo su cariño.

A mis abuelitos por ser los mejores y tener tanta fe en mí.

A mi hermana Yoelsys por sus consejos y locuras que compensan mi carácter.

A mi cuñado Roberto por estar al pendiente de mí durante todos estos años.

A mi sobrino Alain a quien tanto quiero.

A mis tías y primos por estar siempre ahí, en especial a mi tía Addys por ser un ejemplo de amor y entrega.

A mis amigos por estar siempre cuando los necesito, en especial a Jenny por ayudarme a madurar.

A Danelys por su amistad y por su colaboración en el desarrollo del presente trabajo.

A nuestros tutores por su paciencia y apoyo.

A todos los que de una forma u otra han contribuido a mi formación.

Dedicatoria

*De Danelys:
A mi familia.*

*De Maylén:
A mi Familia.*

Resumen

En este trabajo se propone un modelo para el análisis cuantitativo de riesgos, con alcance en el desarrollo de proyectos de producción de software. Para ello se realizó un estudio del estado del arte de los principales centros de pensamiento, modelos y herramientas existentes; analizándose las ventajas y desventajas de cada uno de ellos. El modelo que se propone define: personal involucrado, actividades, técnicas, y artefactos que se generan en el proceso. Se propone además como una de las técnicas, el uso de algoritmos basados en técnicas de softcomputing que permiten calcular el impacto y la probabilidad de los riesgos. La presente investigación surgió por la necesidad de contar en los proyectos productivos de las UCI con un proceso de análisis cuantitativo de riesgos que permitiera al equipo de desarrollo, analizar el impacto y la probabilidad de los riesgos, y tomar las decisiones correctas.

Palabras Claves

Riesgo, Análisis Cuantitativo, Probabilidad, Impacto, Softcomputing.

Índice de Tablas

Tabla 1 Método de evaluación de las variables independientes	5
Tabla 2 Probabilidad de ocurrencia de los riesgos	23
Tabla 3 Impacto esperado en el costo	24
Tabla 4 Matriz de probabilidad e impacto	24
Tabla 5 Fórmula para calcular la exposición.....	25
Tabla 6 Representación del riesgo.....	25
Tabla 7 Ejemplo de Base de Riesgos	65
Tabla 8 Responsabilidad de las clases	89
Tabla 9 Complejidad de las clases.....	89
Tabla 10 Reutilización de las clases	89
Tabla 11 Evaluación de variables.....	95
Tabla 12 Creación de clústeres	95
Tabla 13 Creación de particiones	96
Tabla 14 Generación de reglas borrosas	96
Tabla 15 Refinación de la base de reglas candidatas	96

Índice de Figuras

Figura 1 CRM.....	16
Figura 2 Análisis de Riesgos. Modelo de Boehm	22
Figura 3 Entradas, Herramientas y Salidas del Análisis Cuantitativo	27
Figura 4 Tendencias de los principales modelos	32
Figura 5 Modelo para Análisis Cuantitativo de Riesgos	42
Figura 6 Entradas para la construcción de la base de riesgos.....	46
Figura 7 Representación de la base de riesgos.....	47
Figura 8 Creación de reglas borrosas. Algoritmo MLRUL	51
Figura 9 Construcción de Sistema de Inferencia Borroso	61
Figura 10 Sistema borroso. Topología de 5 capas, orientada a nodos	61
Figura 11 Artefactos de Salida	62
Figura 12 Estructura de la base de casos.....	67
Figura 13 Base de Casos, estructura de los parámetros	68
Figura 14 Base de Casos, valores de los parámetros	69
Figura 15 Fichero de Reglas Borrosas.....	77
Figura 16 Fichero de Reglas Borrosas.....	78
Figura 17 PATDSI.....	79
Figura 18 Diagrama de clases más representativas.....	83
Figura 19 Diagrama de clases: Discretizadores	84
Figura 20 Diagrama de clases: Variables Lingüísticas	85
Figura 21 Diagrama de clases: Funciones de Membresía	86
Figura 22 Procedimientos de las clases.....	88
Figura 23 Profundidad de herencia de las clases.....	89
Figura 24 Diagrama de Componentes	90
Figura 25 Resultado de las pruebas de unidad	91
Figura 26 Herramienta EAR. Unidad Administrativa.....	110
Figura 27 Herramienta EAR	110
Figura 28 Herramienta EAR. Valoración de las amenazas	111
Figura 29 Herramienta EAR. Riesgo acumulado	112
Figura 30 Herramienta @Risk.....	113
Figura 31 Herramienta MB Risk Management.....	114

Tabla de contenido

<i>Agradecimientos</i>	III
<i>Dedicatoria</i>	V
Resumen.....	VI
Índice de Tablas	VII
Índice de Figuras	VIII
Introducción.....	2
Capítulo 1: Estudio del Estado del Arte sobre Análisis Cuantitativo de Riesgos	9
1.1. Introducción.....	9
1.2. Conceptos asociados al dominio del problema.....	11
1.2.1. <i>Definición de riesgo:</i>	11
1.2.2. <i>Definición de gestión de riesgos:</i>	12
1.2.3. <i>Definición de análisis cuantitativo de riesgos:</i>	12
1.2.4. <i>Definición de softcomputing:</i>	12
1.2.5. <i>Definición de impacto del riesgo:</i>	13
1.2.6. <i>Definición de probabilidad de ocurrencia del riesgo:</i>	13
1.3. Centros de Pensamiento	13
1.3.1. <i>Society for Risk Analysis (SRA)</i>	14
1.3.2. <i>International Project Management Association (IPMA)</i>	14
1.3.3. <i>Project Management Institute (PMI)</i>	14
1.3.4. <i>Software Engineering Institute (SEI).</i>	15
1.3.5. <i>Microsoft Corporation</i>	15
1.3.6. <i>Institute of Electrical and Electronics Engineers (IEEE)</i>	15
1.3.7. <i>International Standard Organization (ISO)</i>	15
1.3.8. <i>Risk World</i>	16
1.4. Modelos	16
1.4.1. <i>Continuous Risk Management (CRM)</i>	16
1.4.2. <i>Capability Maturity Model Integration (CMMI)</i>	17
1.4.3. <i>Software Process Improvement and Capability Determination (SPICE)</i>	17
1.4.4. <i>Information Technology / Software Life Cycle Processes (Tecnología de Información/Procesos de ciclo de vida de software)</i>	18
1.4.5. <i>Rational Unified Process (RUP)</i>	19

1.4.6.	<i>Acquisition Risk Management (ARM)</i>	19
1.4.7.	<i>Project Management o Dirección Integrada de Proyectos (DIP)</i>	19
1.4.8.	<i>Risk-Driven Software Process Improvement (DriveSPI)</i>	20
1.4.9.	<i>Modelo de Procesos para la Industria de Software (MoProSoft)</i>	20
1.4.10.	<i>Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información (MAGERIT)</i>	20
1.4.11.	<i>Metodología Eurométodo</i>	21
1.4.12.	<i>Team Risk Management (TRM)</i>	21
1.5.	Modelos Basados en la Experiencia	22
1.5.1.	<i>El Modelo de Boehm</i>	22
1.5.2.	<i>Modelo de Hall</i>	23
1.6.	Modelos Matemáticos	23
1.6.1.	<i>Microsoft Solution Framework (MSF)</i>	23
1.6.2.	<i>Project Management Body of Knowledge (PMBOK)</i>	26
1.6.3.	<i>Modelo de Pressman</i>	27
1.6.4.	<i>Software Risk Evaluation (SRE)</i>	30
1.7.	Modelos Basados en la Simulación.....	31
1.7.1.	<i>PMBOK</i>	31
1.7.2.	<i>Microsoft Solution Framework (MSF)</i>	31
1.7.3.	<i>Principales Tendencias</i>	31
1.8.	Análisis Cuantitativo de Riesgos en la UCI	32
1.9.	Herramientas.....	36
1.9.1.	<i>EAR/PILAR</i>	36
1.9.2.	<i>Technical Risk Identification and Mitigation System (TRIMS)</i>	37
1.9.3.	<i>@Risk</i>	37
1.9.4.	<i>Chinchón</i>	38
1.9.5.	<i>Acertus, Securac`s Enterprise Risk Management (ERM)</i>	38
1.9.6.	<i>MB Risk Management (MBRM)</i>	39
1.9.7.	<i>SCRAM99</i>	39
1.9.8.	<i>Valoración sobre las herramientas actuales</i>	39
1.10.	Conclusiones Parciales	39
Capítulo 2:	Modelo para el Análisis Cuantitativo de Riesgos	41
2.1.	Introducción.	41

2.2.	Características del Modelo	41
2.1.1.	<i>Alcance</i>	43
2.1.2.	<i>Principios</i>	43
2.1.3.	<i>Premisas</i>	45
2.1.4.	<i>Construcción de la Base de Riesgos</i>	46
2.1.5.	<i>Construcción de Base de Casos de Entrenamiento</i>	48
2.1.6.	<i>Generación de Reglas Borrosas</i>	49
2.1.7.	<i>Construcción de Sistema de Inferencia Borroso</i>	59
2.1.8.	<i>Generación de Artefactos Resultantes del Análisis</i>	62
2.3.	Conclusiones Parciales	63
Capítulo 3:	Validación del Modelo	65
3.1.	Introducción.....	65
3.2.	Implantación del modelo.....	65
3.2.1.	<i>Construcción de la Base de Riesgos</i>	65
3.2.2.	<i>Construcción de Base de Casos de Entrenamiento</i>	66
3.2.3.	<i>Generación de Reglas Borrosas</i>	76
3.2.4.	<i>Características de la Implementación</i>	78
3.2.5.	<i>Generación de Artefactos Resultantes del Análisis</i>	93
3.3.	Análisis del modelo	94
3.3.1.	<i>Evaluación de las variables independientes</i>	95
3.3.2.	<i>Análisis de Complejidad</i>	95
3.4.	Conclusiones Parciales	97
Conclusiones.....		98
Recomendaciones		99
Bibliografía		100
Anexos		108
Anexo 1:	Taxonomía del desarrollo de software	108
Anexo 2:	Herramienta EAR/PILAR	110
Anexo 3:	Herramienta @Risk	113
Anexo 4:	Herramienta MBRM.....	114
Anexo 5:	Glosario de Términos de Inteligencia Artificial	115
Anexo 6:	Registro de Riesgos	118
Anexo 7:	Lista Priorizada de Riesgos Cuantificados	122

Anexo 8: Ejemplo de Limpieza de Datos126
Anexo 9: Declaración de Revisión128

Introducción

Un riesgo es una condición futura o circunstancia que puede tener un impacto negativo (amenaza) o positivo (oportunidad) por lo menos en uno de los objetivos del proyecto: costos (cumplir con los costos planificados al principio del proyecto), tiempo (obtener cada versión del software en el tiempo acordado), alcance (cumplir con las características del producto que se acordaron) y calidad (cumplir con los requerimientos de calidad pactados) (PMI, 2004). Es la posibilidad de sufrir un daño o pérdida que aunque no hay certeza de que ocurra, es potencialmente perjudicial y que de concretarse se transforma en un problema.

Un riesgo es una variable del proyecto de software, que pone en peligro o impide el éxito del proyecto. En la gestión de riesgos las etapas fundamentales varían según la tendencia o el autor pero en general se consideran: planeación, identificación, análisis, mitigación, control y seguimiento. (PMI, 2004)

En la etapa de planificación, es donde se decide qué estrategia se utilizará para la gestión de riesgos. En esta etapa los directivos del proyecto deciden cómo afrontar y guiar la gestión de riesgos respetando las políticas o disposiciones de la empresa, utilizando una guía para la gestión de riesgos. Aquí definen qué elementos considerarán de cada riesgo, cómo se van a medir, qué criterios se utilizarán, con qué documentos se trabajará y qué formato y consideraciones deben tener. (PMI, 2004)

En la etapa de identificación los elementos que influyen son: la capacidad de los implicados de detectar un riesgo, la experiencia en gestión de riesgos o en gestión de proyectos en general, pues algunos gerentes de proyectos experimentados crean su propia lista de riesgos siempre presentes que resulta útil en esta etapa, la claridad y el grado de definición en los procesos de gestión de riesgos, el nivel de concientización con su utilidad para el desarrollo satisfactorio del proyecto. (PMI, 2004)

Una vez identificados los riesgos, se pasa a la etapa de análisis, el mismo se puede hacer cualitativo y cuantitativo. El análisis cualitativo es el proceso de medir el impacto de los riesgos identificados de manera no exacta. Los riesgos se analizan de acuerdo a su probabilidad de ocurrencia y el impacto que traerá para el proyecto, y pueden clasificarse en: alto, medio o bajo. Estos valores o niveles (alto, medio, bajo) son definidos en el proceso de planeación e indican cierto grado de relevancia o diferencia respecto a los otros riesgos. Durante el análisis cualitativo se clasifican los riesgos en categorías. Estas varían de acuerdo al tipo de proyecto y al tipo de especialista que identificó el riesgo: ingeniería, ambiente de desarrollo y restricciones. Esta etapa de análisis cualitativo captura las opiniones de los especialistas y de igual forma la experiencia y apreciación correcta es un factor decisivo en la calidad de la gestión de riesgos. El análisis cuantitativo de los riesgos se refiere en cambio a una estimación numérica de la probabilidad de ocurrencia e impacto de cada riesgo; y

utilizando técnicas, métodos estadísticos y de simulación¹, se calcula la probabilidad de alcanzar los objetivos del proyecto. Este es un elemento importante cuando se desea obtener un valor numérico de la certeza de éxito de un proyecto y una vez más su acercamiento a la realidad depende de cuán cerca a lo real estén los valores de probabilidad de ocurrencia.

Durante la etapa de mitigación, se elaboran planes de contingencias para los riesgos que se decidan afrontar. (PMI, 2004)

Durante todo el ciclo de vida del proyecto se mantiene un control y seguimiento del proyecto mediante un monitoreo continuo de los riesgos ya identificados y los nuevos que puedan aparecer a lo largo del proyecto. (PMI, 2004)

Aunque las primeras teorías sobre el análisis cuantitativo aparecieron hace más de dos décadas, sigue siendo evidente la poca utilización de sus técnicas en los proyectos de desarrollo de software actuales del mundo.

Además, las técnicas propuestas por los diferentes modelos en el mundo son habitualmente descripciones generales, y aunque muchas tienen buen soporte matemático o científico, carecen de explicación objetiva de cómo utilizarlas y cómo sus resultados se integran con el análisis cuantitativo.

En la mayoría de los proyectos de software solo se identifican los riesgos y de manera poco consciente.

En la actualidad con el avance de las tecnologías en el campo de la informática, los procesos de desarrollo de software se hacen cada vez más complicados, y cualquier decisión que se tome trae consigo determinadas consecuencias.

El análisis de riesgo cuantitativo se ocupa específicamente de la revisión cuantitativa de los riesgos que pueden presentarse en los distintos tipos de proyectos de software. Una correcta planificación, apoyada sobre esta metodología, permite evitar los riesgos que pueden ocasionar desde pequeños incidentes frecuentes de bajas consecuencias, hasta incidentes de mayores magnitudes y poco comunes. Este proceso es relativamente nuevo y ha probado ser muy valioso como instrumento en la gestión de proyectos de diversas ramas de la economía con distintos niveles de impacto de índole social.

Frecuentemente en los proyectos de software no son bien entendidos los requerimientos, las planificaciones y presupuestos son irrealistas, o se subestima la complejidad del software; surgen

¹ Una simulación de proyecto usa un modelo que traduce las incertidumbres especificadas a un nivel detallado en su impacto potencial de los objetivos, que están expresados al nivel de todo proyecto. Las simulaciones de los proyectos son generalmente realizadas usando la técnica de Monte Carlo.

situaciones que afectan al proyecto y todos estos factores constituyen riesgos que están presentes, y conllevan a catástrofes ignorarlos o no tomar medidas para ello.

La Universidad de las Ciencias Informáticas (UCI) es una institución con solo seis años de experiencia en el desarrollo de software, y aunque ha avanzado considerablemente y ha obtenido muchos logros, todavía le falta madurez en el proceso de desarrollo de software, específicamente en la gestión de riesgos.

Entre los problemas que afectan la calidad de los productos que son liberados en la UCI se encuentra la inadecuada gestión de riesgos, la no utilización de herramientas para su aplicación, y que tienen un marco informático (Figueredo, 2007).

Existen deficiencias en la adaptación de los modelos para el análisis cuantitativo de los riesgos en los proyectos de software en la UCI, y es por ello, entre otros factores, que los proyectos se atrasan en las entregas, y los productos no tienen la calidad requerida o satisfacen las necesidades del cliente. La gestión de riesgos requiere un cierto esfuerzo y aunque su completa aplicación no garantiza el éxito, sí es una práctica primordial. Por estas razones se hace necesario, la creación de un modelo que solucione esta realidad.

Por tanto la investigación se orienta para resolver el siguiente **problema científico**:

- Las insuficiencias de los métodos de análisis cuantitativo de los riesgos para los proyectos de software, está afectando la gestión de riesgos en los proyectos productivos de la UCI y dificultando el análisis del impacto y de la probabilidad de ocurrencia de los riesgos.

Para ello se plantea como **objetivo general**:

- Desarrollar un modelo para el análisis cuantitativo de los riesgos en los proyectos de desarrollo de software, basado en técnicas de softcomputing.

Objeto de estudio: La gestión de riesgos de proyectos de software.

Campo de acción: El análisis cuantitativo de los riesgos de proyectos de software.

Idea a Defender: Si se desarrolla un modelo para el análisis cuantitativo de los riesgos, basado en técnicas de softcomputing, entonces se logrará mejorar el análisis del impacto y la estimación de la probabilidad de ocurrencia de los riesgos en los proyectos de desarrollo de software en la UCI.

Objetivos Específicos:

- 1- Realizar un estudio del estado del arte del análisis cuantitativo de los riesgos en proyectos de software.
- 2- Confeccionar un modelo basado en técnicas de softcomputing que permita analizar la probabilidad de ocurrencia y el impacto de los riesgos en proyectos de software.
- 3- Validar el modelo desarrollado.

Variables:

- Variables Independientes: un modelo para el análisis de los riesgos, y la implementación de los algoritmos para la estimación del impacto y la probabilidad de ocurrencia de los riesgos.
- Variables Dependientes: mejora del proceso de análisis cuantitativo de riesgos.

Tabla 1 Método de evaluación de las variables independientes

VARIABLES INDEPENDIENTES	DIMENSIÓN	INDICADORES	UNIDADES
Modelo para el análisis cuantitativo de los riesgos.	Representación	Representación gráfica	__ Sí __ No
		Grado de completitud y claridad de la documentación	__ Alto __ Medio __ Bajo
	Aplicabilidad del modelo.	Aplicable a proyectos de gestión.	__ Sí __ No
		Estimar el impacto de forma cuantitativa exacta	__ Sí __ No
		Estimar la probabilidad de ocurrencia	__ Sí __ No
Implementación de los Algoritmos	Complejidad	Complejidad Espacial	Memoria(KB)
		Complejidad Temporal	O()
	Sencillez	__ Sí __ No	

Para dar respuesta al problema anteriormente planteado, se proponen las siguientes *tareas de la investigación*:

1. Analizar el enfoque de los centros de pensamiento existentes en materia de análisis cuantitativo en la gestión de riesgos:
 - a. Society for Risk Analysis (SRA)
 - b. International Project Management Association (IPMA)
 - c. Project Management Institute (PMI)
 - d. Software Engineering Institute (SEI)
 - e. Microsoft Corporation
 - f. Institute of Electrical and Electronics Engineers (IEEE)
 - g. International Standard Organization (ISO)

- h. Risk World
2. Analizar el enfoque de los modelos existentes, y de ellos clasificar los más representativos, en cuanto al análisis cuantitativo en la gestión de riesgos:
 - a. Modelo de Boehm
 - b. Modelo de Pressman
 - c. Modelo de Hall
 - d. Software Risk Evaluation (SRE)
 - e. Continuous Risk Management (CRM)
 - f. Microsoft Solution Framework (MSF)
 - g. Project Management Body of Knowledge (PMBOK)
 - h. Capability Maturity Model Integration (CMMI)
 - i. Software Process Improvement and Capability Determination (SPICE)
 - j. Information Technology / Software Life Cycle Processes (Tecnología de Información/Procesos de ciclo de vida de software)
 - k. Rational Unified Process (RUP)
 - l. Acquisition Risk Management (ARM)
 - m. Project Management o Dirección Integrada de Proyectos (DIP)
 - n. Risk-Driven Software Process Improvement (DriveSPI)
 - o. Modelo de Procesos para la Industria de Software (MoProSoft)
 - p. Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información (MAGERIT)
 - q. Metodología Eurométodo
 - r. Team Risk Management (TRM)
 3. Identificar las principales tendencias, y herramientas para el análisis cuantitativo de la gestión de riesgos.
 4. Implementar algoritmos basados en técnicas de softcomputing que permitan calcular el impacto y la probabilidad de ocurrencia de los riesgos en proyectos de software.
 5. Conformar el modelo.
 6. Evaluar el modelo propuesto y analizar los resultados.

Para realizar las tareas antes propuestas se utilizan los métodos científicos de la investigación: teórico y empírico.

Métodos teóricos:

Histórico lógico: Se realizó un estudio del arte, desde el inicio de la investigación, sobre la evolución que ha tenido en el mundo el tema análisis cuantitativo de los riesgos de proyectos de software, sus principales tendencias a lo largo de los años y desarrollo. Se evaluó a profundidad la bibliografía y se reconocieron las ventajas y desventajas de cada una de las tendencias, logrando así identificar los rasgos y características esenciales del análisis cuantitativo de proyectos de software y estableciendo una conexión entre su concepción histórica y la actualidad.

Hipotético Deductivo: Se realizó un análisis hipotético-deductivo, pues a partir de la problemática existente en la UCI se planteó el objetivo general de la investigación y los objetivos específicos, y se propuso además una idea a defender que en el transcurso de la investigación se corrobora, siguiendo para ello métodos científicos fundamentados llegando a introducir nuevos conocimientos que posteriormente fueron evaluados.

Sistémico: Se planteó el problema y su solución como un todo, realizando un estudio de cada uno de los procesos y criterios del análisis cuantitativo de riesgos en proyectos de software, estableciendo dependencias y conexiones entre cada uno de los procesos para poder lograr un resultado integral, estableciéndose así un sistema sostenible y completo. Facilitó determinar los pasos para el análisis cuantitativo, el algoritmo de solución y definir las relaciones entre estos y el modelo propuesto para favorecer el logro de una correcta y exhaustiva gestión de riesgos, y para obtener una implementación de algoritmos que ayude en el análisis cuantitativo de los riesgos.

Métodos empíricos:

Experimental: proporcionó un estudio del objeto en el cual se crearon las condiciones y se adaptaron las existentes para verificar la hipótesis y el modelo formulado. Suministró además la revelación de la realidad para actuar sobre ella y transformarla, y verificar el estado del fenómeno.

Aporte Teórico Práctico:

La novedad científica del presente trabajo es el desarrollo de un modelo e implementación de algoritmos para el análisis cuantitativo de los riesgos, basada en técnicas de softcomputing, permitiendo así la mejora del proceso de análisis y la estimación del impacto y la probabilidad de ocurrencia de los riesgos en los proyectos de desarrollo de software en la UCI.

Estructura del trabajo:

La tesis está compuesta por tres capítulos. En el primer capítulo se realiza un análisis de los principales centros de pensamiento y la evolución de la gestión de riesgos en el mundo del software,

las tendencias de los principales modelos y herramientas, así como su estado en el mundo, en Cuba y en la UCI.

En el Capítulo 2 se describe el modelo propuesto para el análisis cuantitativo de los riesgos basado en técnicas de softcomputing, explicando el por qué de cada paso del modelo, y se describe la propuesta de la implementación de algoritmos para tal propósito, así como las actividades, entradas y salidas que describen el proceso.

El Capítulo 3 describe la evaluación, valoración y validación de la solución propuesta.

Capítulo 1: Estudio del Estado del Arte sobre Análisis Cuantitativo de Riesgos

1.1. Introducción.

La primera técnica cuantitativa de la administración moderna de proyecto en el área del análisis de riesgos relacionado con los tiempos o plazos del proyecto fue el diagrama de Gantt, desarrollada por Henry Gantt en 1917.

El diagrama de Gantt proporciona un resumen gráfico del progreso de un listado de actividades que son mostradas verticalmente, representando el inicio y la duración de cada actividad por una línea horizontal a lo largo de una escala de tiempo. (Galway, 2004)

Ante las limitantes del diagrama de Gantt, y en la búsqueda de nuevas herramientas, a mediados de los años 50, la Oficina de Proyectos Especiales Polaris desarrolló la técnica PERT (Program Evaluation Review Technique).

La base del PERT fue un detallado diagrama de todas las tareas anticipadas en un proyecto, organizadas en una red, la cual representa la dependencia de cada tarea con relación a aquellas tareas que las preceden. (Galway, 2004)

En los mismos años 50 se desarrolló una técnica de planificación y administración que fue desarrollada por Du Pont: la técnica Critical Path Method (CPM). Esta técnica también utiliza la representación de una red, pero inicialmente no utilizaba distribuciones de probabilidades para determinar la duración o el plazo de las tareas. Con el avance de las capacidades de los computadores, la técnica CPM fue mejorada utilizando el método de simulación de Monte Carlo². La estimación de los tiempos o plazo de cada tarea aplicando la técnica de Monte Carlo dio lugar a la técnica a CPM estocástico, la cual es acualmente la metodología preferida para evaluar el riesgo en la estimación del tiempo en la administración de proyectos.

La gestión de riesgos en los proyectos de software se comienza a estudiar a principios de los 80, aunque se inició a formalizarse a finales de esa década y principios de los 90, con trabajos del “Institute of Electrical and Electronics Engineers” (IEEE) y el “Software Engineering Institute” (SEI). Estos autores sientan las bases de los procesos y principios involucrados en la gestión de riesgos.

² Bajo el nombre de “Método de Monte Carlo” o “Simulación Monte Carlo” se agrupan una serie de procedimientos que analizan distribuciones de variables aleatorias usando simulación de números aleatorios.

Surgen varios métodos, herramientas y frameworks complementarios a estos modelos o a sus variantes.

El estudio de la gestión de riesgos en los años 80 es llamado Primera Generación o Casuística. Está basada en listas 'casuísticas' de riesgos especiales para proyectos, esto consiste en identificar casos de riesgo y extrapolarlos a otros proyectos. En esta generación se definen los riesgos tecnológicos y las listas de comprobación de riesgos, y se limitan las tareas a la identificación de riesgos en los proyectos con técnicas basadas en cuestionarios, listas de incidencias y de las medidas para contrarrestarlas.

Está precedida por ciertos elementos iniciales, como son:

- Años 40: aparece la teoría de la fiabilidad, arranque de la Teoría del Riesgo en Sistemas Complejos con el Teorema de Lusser: "la probabilidad de éxito (no fallo) de una cadena de componentes es el producto de las probabilidades de éxito de sus elementos". (Zulueta, y otros, 2007)
- Años 60: se introduce el análisis de riesgos cuantitativo y cualitativo. Se obtiene una definición del riesgo "como una entidad con dos dimensiones: probabilidad y consecuencia(s)" o sea, vulnerabilidad e impacto, la cual, en cierto modo sigue vigente en la actualidad.
- Años 70: surge el método general de Rasmussen, que divide en seis etapas lo que debería ser un plan de gestión de riesgos, una de ellas es el análisis. (García, 2007)

A partir de los años 90 surge la Segunda Generación o Taxonómica, donde se empieza a madurar el análisis de riesgos como parte de la gestión de los riesgos, con el desarrollo de modelos como: Modelo de Boehm, Modelo de Hall y su relación con la madurez SEI-CMMI, Modelo de Riesgos del SEI, y el Modelo SPR de mejora de capacidad en la gestión de riesgos. Está basada en modelos de procesos y eventos. Del Modelo de Boehm es que surge el análisis de riesgos, una de las ideas fundamentales que las metodologías posteriores han adoptado.

La Tercera Generación o Causal, es la generación actualmente emergente (García, 2007), con modelos como: Modelo MAGERIT, Modelo McFarlan, Modelo RiskMan y RiskDriver, Modelo DriveSPI, Modelo Eurométodo, Modelo ISPL, Modelo Prisa, entre otros. En esta generación los modelos se refieren en particular a proyectos informáticos. Articula también una causalidad más explicativa y por lo tanto más predictiva entre los elementos del modelo, sobre todo entre los factores de riesgo y sus medidas reductoras o salvaguardas.

En este capítulo se examinan los diferentes modelos y herramientas desarrollados en el mundo, sus principales tendencias, los más destacados autores, sociedades y centros de pensamiento, teniendo

en cuenta las técnicas y métodos que proponen para llevar a cabo el proceso de análisis cuantitativo de riesgos.

1.2. Conceptos asociados al dominio del problema

1.2.1. Definición de riesgo:

El SEI plantea que un riesgo es la posibilidad de sufrir una pérdida.

El “Project Management Body of Knowledge” (PMBOK) plantea que un riesgo de un proyecto es un evento o condición inciertos 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; etc.). Un riesgo puede tener una o más causas y, si se produce, uno o más impactos.

Es cualquier suceso que pueda afectar negativamente a la marcha del proyecto en el futuro, es asociado de manera inexorable a cualquier actividad que se lleve a cabo y que imponga la decisión entre varias alternativas, por tanto, acompaña todo cambio y está presente en cada decisión. Implica elección e incertidumbre. (Fuente, y otros, 2006)

Un riesgo es cualquier suceso que pueda afectar negativamente a la marcha del proyecto en el futuro. El riesgo se haya asociado de manera inexorable a cualquier actividad que se lleve a cabo y que imponga la decisión entre varias alternativas (prácticamente todo es riesgo). El riesgo, por tanto, acompaña a todo cambio y está presente en cada decisión. El riesgo implica elección e incertidumbre. (García, 2007)

Es un evento incierto y tiene un impacto negativo. (Galway, 2004)

Pressman, en *Ingeniería de Software, En enfoque práctico*, define el riesgo como la posibilidad de que un evento adverso, desgracia o contratiempo pueda manifestarse produciendo una pérdida.

El riesgo se define con dos dimensiones: la de incertidumbre (evaluada como probabilidad de ocurrencia), y el efecto dimensión (evaluado como impacto en los objetivos). El cálculo correcto de riesgos requiere la evaluación apropiada, tanto de la probabilidad como del impacto. (Hillson, y otros, 2004)

1.2.2. Definición de gestión de riesgos:

La gestión de riesgos en el ámbito del software procura formalizar conocimiento orientado a la minimización o evitación de los mismos en proyectos de desarrollo de software, mediante la generación de principios y buenas prácticas de aplicación realista. (ROPPONEN, y otros, 2000)

El PMBOK precisa que la gestión de los riesgos del proyecto incluye los procesos relacionados con la planificación de la gestión de riesgos, la identificación y el análisis de riesgos, las respuestas a los riesgos, y el seguimiento y control de riesgos de un proyecto. Los objetivos de la gestión de los riesgos del proyecto son aumentar la probabilidad y el impacto de los eventos positivos, y disminuir la probabilidad y el impacto de los eventos adversos para el proyecto.

Es el arte y la ciencia de identificar, analizar, y responder a los riesgos a lo largo de la vida de un proyecto, con el propósito de lograr los objetivos del proyecto. (Galway, 2004)

1.2.3. Definición de análisis cuantitativo de riesgos:

El PMBOK define que es el proceso de analizar numéricamente el efecto de los riesgos identificados en los objetivos generales del proyecto.

Es el proceso de valorar cuantitativamente los riesgos. Involucra una estimación de incertidumbre del riesgo y su impacto. (Galway, 2004)

Es la determinación cuantitativa del riesgo de una instalación, en base a la determinación de las frecuencias de ocurrencia (probabilidad) e impacto del riesgo.

1.2.4. Definición de softcomputing:

Softcomputing es una rama de la Inteligencia Artificial, centrada en el diseño de sistemas inteligentes capaces de manejar adecuadamente la información imprecisa, incierta y/o incompleta.

El término softcomputing lo introduce, en 1994, Lofti A. Zadeh para denotar una aproximación al razonamiento humano que deliberadamente hace uso de la tolerancia humana a imprecisiones y vaguedades para obtener soluciones razonables que son fáciles de manipular.

Los sistemas borrosos, las redes neuronales, la computación evolutiva, el razonamiento probabilístico y las combinaciones de dichas técnicas son considerados como softcomputing. (Piñero, 2005)

Otro concepto plantea que softcomputing es un término para describir el uso combinado de diferentes aproximaciones computacionales surgidas en los últimos treinta años, y entre las que se destacan: los sistemas difusos, las redes neuronales y los algoritmos evolutivos.

Lofti A. Zadeh propuso la definición de softcomputing, estableciéndola en los siguientes términos:

"Básicamente, softcomputing no es un cuerpo homogéneo de conceptos y técnicas. Más bien es una mezcla de distintos métodos que de una forma u otra cooperan desde sus fundamentos. En este sentido, el principal objetivo de la softcomputing es aprovechar la tolerancia que conllevan la imprecisión y la incertidumbre, para conseguir manejabilidad, robustez y soluciones de bajo costo. Los principales ingredientes de la softcomputing son la lógica fuzzy, la neuro computación y el razonamiento probabilístico, incluyendo este último a los algoritmos genéticos, las redes de creencia, los sistemas caóticos y algunas partes de la teoría de aprendizaje. En esa asociación de lógica fuzzy, neuro computación y razonamiento probabilístico, la lógica fuzzy se ocupa principalmente de la imprecisión y el razonamiento aproximado; la neuro computación del aprendizaje, y el razonamiento probabilístico de la incertidumbre y la propagación de las creencias".

1.2.5. Definición de impacto del riesgo:

El impacto del riesgo calcula la gravedad de los efectos adversos, la magnitud de una pérdida o el costo potencial de la oportunidad si el riesgo llega a producirse dentro del proyecto. (Kuna, 2008)

Puede computarse en unidades de tiempo (retraso), unidades monetarias (incremento de costes), alcance (tamaño del producto en líneas de código, puntos función entre otros). (PMI, 2004)

Es el alcance de lo que sucedería si el riesgo se materializa, el efecto.

1.2.6. Definición de probabilidad de ocurrencia del riesgo:

Es la posibilidad con que un evento o condición puede ocurrir (la dimensión de incertidumbre). (Hillson, y otros, 2004)

Posee un valor entre 0 y 1.

Probabilidad se entiende como la posibilidad de que el riesgo se manifieste en cualquier momento.

1.3. Centros de Pensamiento

Los centros de pensamiento más importantes, en materia de análisis cuantitativo de riesgos, identificados son la Sociedad para el Análisis del Riesgo o Society for Risk Analysis (SRA) por sus siglas en inglés, la Asociación Internacional de Dirección de Proyectos (Internacional Project Management Association (IPMA)), Instituto de Dirección de Proyectos (Project Management Institute (PMI)), Instituto de Ingeniería de Software (Software Engineering Institute (SEI)), Corporación Microsoft (Microsoft Corporation), Instituto de Ingenieros Eléctricos y Electrónicos (Institute of Electrical and

Electronics Engineers (IEEE)), Organización Internacional de Estándares (International Standard Organization(ISO)), y la Risk World.

1.3.1. Society for Risk Analysis (SRA)

La Sociedad para el Análisis de Riesgos, fundada en 1981, ofrece un área de colaboración para profesionales de varias disciplinas vinculados al tema, es la organización líder en la investigación académica en el campo de los riesgos.

Es una sociedad multidisciplinaria, erudita, internacional que mantiene un foro abierto a todos aquellos que estén interesados en el análisis de riesgo.

Entre sus objetivos se encuentra el intercambio de información, ideas y metodologías para el análisis de riesgos. Su publicación más importante es: "Risk analysis, an International Journal".

1.3.2. International Project Management Association (IPMA)

Es uno de los principales repositorios y fuente bibliográfica sobre análisis de riesgos. Creado en 1965, la Asociación Internacional de Dirección de Proyectos, fue en su momento la primera asociación de dirección de proyecto del mundo.

Desde entonces IPMA ha crecido a nivel mundial y se ha vuelto una red internacional para las sociedades de dirección de proyectos. Representa a varias asociaciones de dirección de proyecto nacionales en el nivel internacional. Certifican a gerentes del proyecto, premian los equipos de proyectos exitosos e investigadores, y proporcionan varias publicaciones de dirección de proyectos.

IPMA es una de las principales referencias de la gestión formal de proyectos, y uno de sus segmentos principales es la gestión de riesgos. (Aguilar, 2005)

1.3.3. Project Management Institute (PMI)

Este Instituto fue fundado en 1969. Durante este tiempo el PMI, a través del comité de estándares y colaboradores (entre ellos empresas, universidades, asociaciones de profesionales, especialistas y consultores en proyectos) ha realizado el estudio, evaluación y revisión de los estándares generalmente aceptados a nivel internacional, dando como resultado los estándares que representan el cuerpo de conocimientos de la Dirección de Proyectos, cuyo título original es "Project Management Body of Knowledge" (PMBOK). En 1987 se publicó su primera edición. Una de sus áreas es la descripción de los procesos relacionados con el desarrollo de la gestión de riesgos de un proyecto, y dentro de esta se encuentra como uno de los procesos principales el análisis cualitativo y cuantitativo de los riesgos.

1.3.4. Software Engineering Institute (SEI).

El SEI, fundado en 1984, es una unidad de la Universidad Carnegie Mellon, dedicado al desarrollo de la ingeniería de software. Encargado de la investigación y creación de modelos para la mejora del desarrollo de software, comienza a principios de los años 90 a desplegar trabajos en la gestión de riesgos. Uno de los primeros es el Continuous Risk Management (CRM), un método de gestión de riesgos que incluye los principios básicos que concuerdan en gran medida con el modelo de Boehm. Otros trabajos son: Software Risk Evaluation (SRE), que define un proceso para la identificación, análisis y desarrollo de estrategias de mitigación; Taxonomy-based Risk Identification, un método para facilitar el proceso de identificación de los riesgos; y Team Risk Management (TRM), que define la estructura y operaciones a realizar para gestionar los riesgos en una organización por un equipo. (Williams, 2003)

1.3.5. Microsoft Corporation

Microsoft Corporation, fundada en 1975, corporación multinacional productora de software líder en el mundo. En el 2002 promueve Microsoft Solutions Framework, un conjunto de guías, prácticas, procedimientos y herramientas para el desarrollo de aplicaciones, que incluye una disciplina para la Gestión de Riesgos: Microsoft Solutions Framework – Risk Management Discipline.

1.3.6. Institute of Electrical and Electronics Engineers (IEEE)

La IEEE es una asociación técnico-profesional mundial dedicada a la estandarización, entre otras tareas. Está formada por profesionales de las nuevas tecnologías, como ingenieros eléctricos, ingenieros en electrónica, científicos de la computación, ingenieros en informática e ingenieros en telecomunicación. Aunque su creación se remonta a 1884, en 1963 fue que adoptó su nombre. Esta organización ha formalizado la gestión de riesgos con el modelo propuesto por Boehm y continúan desarrollando investigaciones y publicaciones en este campo. La IEEE definió además un estándar para la gestión de riesgos: 1540-2001, que puede combinarse con el estándar 12207, también llamado: de procesos de ciclo de vida de software, o puede ser usado independientemente.

1.3.7. International Standard Organization (ISO)

Es la Organización Internacional para la Estandarización. Creado en 1947, es el organismo encargado de promover el desarrollo de normas internacionales de fabricación, comercio y comunicación para

todas las ramas industriales a excepción de la eléctrica y la electrónica. Su función principal es la de buscar la estandarización de normas de productos y seguridad para las empresas u organizaciones a nivel internacional.

ISO desarrolló una norma para la gestión de riesgos: ISO/IEC 16085:2006, que define un proceso para la administración de riesgos y puede ser aplicada en el ciclo de vida de un proyecto de software.

1.3.8. Risk World

Es uno de los primordiales repositorios de materiales de riesgos. Reúne una gran variedad de referencias a noticias, publicaciones internacionales, libros, bases de datos de riesgos de diferentes sectores, herramientas para ayudar en la gestión de riesgos; asociaciones e institutos en el mundo enfocados a la gestión de riesgos, grupos de discusión, artículos y otros materiales. Durante algunos años fue el publicador del “Risk Analysis, an International Journal” de la Society for Risk Analysis (SRA).

1.4. Modelos

1.4.1. Continuous Risk Management (CRM)

Desarrollado por The Software Assurance Technology Center” (TSATC) de la NASA en conjunto con el SEI, a principios de los 90. Es un método que define seis principios o funciones para la gestión de riesgos: identificación, análisis, plan, seguimiento, control, todos ellos basados en la comunicación. Como se muestra en la Figura 1.



Figura 1 CRM

En este método estos procesos ocurren continuamente e iterativamente, pues “consideran los riesgos antes de que se vuelvan problemas”. Definen en el análisis, con el objetivo de proveer datos para posteriormente tomar decisiones, el proceso de examinar cada riesgo en detalle, cómo se relacionan unos con otros y cuáles son los más importantes.

Durante la evaluación se analiza cada riesgo y se deciden valores para el impacto (pérdidas o efecto negativo en los objetivos del proyecto en caso de que ocurra el riesgo), probabilidad (probabilidad de ocurrencia del riesgo) (Rosemberg, y otros, 1999). En este proceso cada riesgo debe ser entendido lo suficientemente como para permitir la dirección de la toma de decisiones.

El análisis de los riesgos debe comenzar a la vez que el desarrollo del proyecto de software. CRM representa un enfoque válido y positivo para abordar de forma sistemática la gestión de riesgos en proyectos de desarrollo de software, aunque insuficiente en su descripción oficial para proyectos complejos de desarrollo de software de gestión. (Estéves)

Este modelo cuenta con una estructura sólida, bien definida a través de sus seis funciones, una de ellas el análisis. A su vez tiene tres subprocesos: evaluar, clasificar y priorizar. Pero en su fase de análisis a pesar de que trabaja con los valores de probabilidades de ocurrencia del riesgo y de impacto, sólo se establecen valores cualitativos para cada atributo y los criterios para decidirse por un valor u otro; o sea, no se trabaja con los valores cuantitativos.

1.4.2. Capability Maturity Model Integration (CMMI)

Es un modelo integrado de madurez para la mejora de procesos en proyectos de software. Es un modelo de calidad del software que clasifica las empresas en niveles de madurez (Gracia, 2005). La gestión de riesgos es un área de procesos contemplada en su nivel 3 que forma parte de la gestión de proyectos, y tiene como objetivo identificar los problemas antes de que ocurran. Una de las tareas genéricas de este nivel es la identificación y análisis de riesgos.

Este modelo cuenta con estructura bien definida, pero no establece cómo se debe realizar el análisis cuantitativo de los riesgos paso a paso.

1.4.3. Software Process Improvement and Capability Determination (SPICE)

ISO/IEC 15504, también conocido como SPICE desde 1995, es un estándar para evaluar la capacidad del proceso, donde se valora la organización del desarrollo software en la dimensión del proceso contra los atributos del proceso en la dimensión de capacidad.

Fue aprobado por la comisión ISO (International Organization for Standardization)/IEC (International Electrotechnical Commission). JTC1 (Joint Technical Committee on Information Technology).

Es un modelo para la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software. Está alineado con el estándar ISO/IEC 12207 que define los procesos del ciclo de vida del desarrollo, mantenimiento y operación de los sistemas de software.

Uno de los procesos de la Dimensión de Procesos de Gestión (MAN) de este estándar, es la Gestión de Riesgos. Este estándar posee poca documentación y la gestión de riesgos se materializa de forma insuficiente, pues no explica cómo implementar las estrategias que propone para el análisis de los riesgos, o sea, no detalla este proceso.

1.4.4. Information Technology / Software Life Cycle Processes (Tecnología de Información/Procesos de ciclo de vida de software)

Publicado en 1995 como estándar internacional por la ISO/IEC JTC1, ISO/IEC 12207, también conocido como Information Technology / Software Life Cycle Processes, es el estándar para los procesos de ciclo de vida del software de la organización ISO.

Es una norma técnica que establece un marco de referencia común para los procesos del ciclo de vida del software.

Establece un proceso de ciclo de vida para el software que incluye procesos y actividades que se aplican desde la definición de requisitos, pasando por la adquisición y configuración de los servicios del sistema, hasta la finalización de su uso.

La estructura del estándar ha sido concebida de manera flexible y modular de manera que pueda ser adaptada a las necesidades de cualquiera que lo use. Para conseguirlo, el estándar se basa en dos principios fundamentales: modularidad y responsabilidad. Con la modularidad se pretende conseguir procesos con un mínimo acoplamiento y una máxima cohesión. En cuanto a la responsabilidad, se busca establecer un responsable para cada proceso, facilitando la aplicación del estándar en proyectos en los que pueden existir distintas personas u organizaciones involucradas. (Singh, 1999)

En todas las fases del ciclo de vida propone el análisis de los criterios apropiados que incluya los riesgos, costos y beneficios de cada opción, y procedimientos para gestionar los riesgos. No obstante la gestión de riesgos la agrupa formalmente dentro de los procesos organizativos del ciclo de vida, esta incluye un análisis de riesgos a lo largo de la implementación en espiral del modelo.

En este proceso se analizan los riesgos en términos de probabilidades y consecuencias, y se determina la prioridad en el tratamiento de estos riesgos.

1.4.5. Rational Unified Process (RUP)

También conocido como Proceso Unificado de Desarrollo, es una metodología de desarrollo de software.

Es un proceso que de manera ordenada que define las tareas y quién de los miembros del equipo de desarrollo las hará. Es además una guía para usar UML (Unified Modeling Language) (Booch, y otros, 2000). Es la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP plantea que la aproximación iterativa está dirigida por los riesgos. Los autores de la metodología refieren como una mala experiencia, el no tener una planificación de riesgos. En tres de sus cuatro fases RUP le da gran importancia a identificar los riesgos críticos y a mitigarlos. Este proceso solo se lograría haciendo un análisis previo de los mismos, pero RUP no explica de manera puntual cómo sería el análisis de los riesgos. No obstante define que la gestión de riesgos es una actividad de la disciplina de gestión de proyectos, y en ella se debe elaborar el artefacto: plan de gestión de riesgos, donde una de sus tareas es evaluar y analizar los riesgos, una vez que hayan sido identificados. Identifica qué acciones se tomarán para reducir la probabilidad de ocurrencia del riesgo, o para reducir su impacto en el proyecto, pero no expone cómo se calcularían esta probabilidad e impacto del riesgo.

1.4.6. Acquisition Risk Management (ARM)

Es un proceso propuesto para la gestión de riesgos. Esta guía proporciona un programa para el mejoramiento de adquisición. Los riesgos en el proceso de ARM son gestionados a lo largo ciclo de vida del software. En uno de los principales objetivos de esta guía se encuentra el análisis de riesgos, previsto por el equipo del proyecto para todas las funciones del mismo.

El proceso de ARM hace un análisis de los riesgos identificados en la adquisición del software enfocándose en los riesgos de alta prioridad los cuales son incorporados al plan de adquisición del software. Este es un proceso continuo que integra todas las actividades del ciclo de vida del software. (Gallagher)

Aunque ARM propone que se debe hacer un análisis de los riesgos, se enfoca más en la adquisición que en el análisis.

1.4.7. Project Management o Dirección Integrada de Proyectos (DIP)

Desarrollado en los últimos quince años. Tiene como finalidad optimizar el uso de los recursos previstos para un proyecto, a través de un proceso de dirección adecuado, y conseguir el cumplimiento

de los objetivos de alcance, coste, plazo y calidad. Este posee un proceso de análisis de riesgos, que permite conocer los posibles efectos negativos.

1.4.8. Risk-Driven Software Process Improvement (DriveSPI)

Surge como resultado de aplicar metodologías de gestión de riesgos a varios proyectos de desarrollo de software. Fue llevado a cabo en Europa entre los años 1995 y 1997 con el objetivo de validar en aplicaciones de prueba un marco de trabajo donde mejorar la madurez de los procesos de software con un fuerte énfasis en la gestión de riesgos, culminando en la definición de un conjunto de líneas guía (una serie de prácticas para aplicar gestión de riesgo a proyectos de desarrollo y de mejoramiento de software). (García, 2007)

Estas líneas guía ayudan a poner en práctica la gestión de riesgos en las empresas. Esta metodología tiene como uno de sus procesos la definición de métricas de riesgo y la clasificación de riesgos, a través de un análisis previo.

1.4.9. Modelo de Procesos para la Industria de Software (MoProSoft)

Es un modelo de procesos y evaluación apropiada para la industria de software fue desarrollado en México. En el proceso gestión de procesos, aparecen el establecimiento o actualización del Plan de Manejo de Riesgos y la definición del Plan de Manejo de Riesgos respectivamente. (Gil, 2009)

Este plan se centra en objetivos que ayudan a gestionar los riesgos. Precisamente uno de los procesos del plan es el análisis cuantitativo de los riesgos.

1.4.10. Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información (MAGERIT)

Fue creado inicialmente para gestionar los posibles riesgos de seguridad derivados de la utilización de medios electrónicos, informáticos y telemáticos, el cual ha sido adaptado a proyectos posteriormente. Propone guías de actividades que contemplan un eficiente tratamiento de riesgos, que puede ser adaptable por su amplitud y los fundamentos que trata, a los proyectos de software.

Se divide en tres sub modelos: de procesos, de entidades y de eventos, dirigidos a las actividades de gestión de riesgos. En el primero de ellos, una de sus tareas es el análisis de los riesgos. El análisis de los riesgos implica la evaluación del impacto, señala los riesgos existentes identificando las amenazas y determinando la vulnerabilidad del sistema de prevención de dichas amenazas.

Este método fue elaborado por un equipo del Comité Técnico de Seguridad de los Sistemas de Información y Tratamiento Automatizado de Datos Personales, del Consejo Superior de Informática de España. Es una metodología de carácter público, perteneciente al Ministerio de Administraciones Públicas (MAP). Su utilización no requiere autorización previa del MAP.

Algunos autores proponen el empleo de MAGERIT con la herramienta PILAR (Ver epígrafe 1.9.1), bajo la política de que en análisis constituye el núcleo central de toda actuación organizada y sistemática en materia de seguridad para mejorar la gestión global de riesgos. Debido a que el empleo de MAGERIT es complejo y prácticamente imposible de realizar a mano cuando el número de activos es relativamente grande, se justifica el uso de una ayuda en forma de herramienta. Además, debido al inevitable mantenimiento futuro, necesario para la propia organización, impone el la adopción de una herramienta que automatice este proceso de análisis.

1.4.11. Metodología Eurométodo

El Eurométodo es un proyecto de la Comisión Europea, cuyos primeros planteamientos datan de 1989, y que culminó con el Eurométodo v.1.1 en 1996.

Este marco metodológico ayuda a planificar y desarrollar contratos de proyectos y servicios referentes a sistemas de información. Desde 1996 trabaja con orientación plena al riesgo de proyectos y desarrolla, en uno de sus capítulos el de análisis y gestión de dichos riesgos. O sea, una de las fases de la gestión de riesgos del Eurométodo es el análisis. También define un conjunto de medidas de salvaguarda.

En su análisis de riesgo incluye: documentar el estado inicial y el estado final, valorar los factores de situación, evaluar la incertidumbre y complejidad generales, analizar la probabilidad de riesgos, e identificar los riesgos críticos. Aunque este análisis es más bien desde el punto de vista cualitativo.

1.4.12. Team Risk Management (TRM)

Es un nuevo paradigma para la administración de programas o proyectos por desarrolladores, usando los principios y herramientas de gestión de riesgos para administrar cooperativamente los riesgos y las oportunidades (Higuera, 2005). Establece como uno de sus procesos el análisis de los riesgos, que es ejecutado para determinar qué es importante para el proyecto, para cambiar prioridades y para asignar recursos. Para un grupo de riesgos se cuantifica su impacto, probabilidad y time frame (período en el que se deben tomar medidas al respecto).

Esta metodología, que hereda el paradigma de gestión de riesgos del SEI, y creada por este instituto, establece una construcción de ambiente en cambios de procesos, métodos, y herramientas que

permiten al cliente y al equipo de desarrollo trabajar cooperativamente, continuamente gestionando riesgos a través del ciclo de vida de desarrollo de software, pero no explica detalladamente cómo debería ser el proceso de análisis cuantitativo de estos riesgos.

1.5. Modelos Basados en la Experiencia

1.5.1. El Modelo de Boehm

La IEEE fue una de las primeras instituciones en formalizar la gestión de riesgos con el modelo propuesto por Boehm en 1989. (García, 2007)

El modelo de Boehm propone procesos y técnicas divididas en dos partes principales: medición de los riesgos y control de los riesgos. Dentro de la medición incluye procesos como: identificación, análisis y priorización. Para el proceso de análisis propone modelos de rendimiento, modelos de costo, análisis de redes, análisis de decisiones³ y análisis del factor de calidad, como puede verse en la Figura 2.

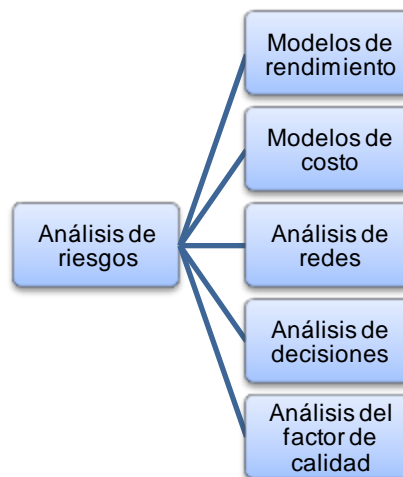


Figura 2 Análisis de Riesgos. Modelo de Boehm

Creado en 1988, el modelo de Boehm se basa en experiencia, pues plantea como una de las estrategias del análisis de los riesgos el análisis de decisiones y este no es más que calcular la probabilidad de experiencias anteriores. Otra de sus técnicas para el análisis que lo inclina hacia ser un modelo basado en la experiencia es el análisis de redes. A pesar de ello no es factible usarlo en el

³ Un análisis de decisiones es usualmente estructurado como un árbol de decisiones. Este diagrama describe una decisión bajo consideración y las implicadas de elegir una u otra de las alternativas disponibles. Incorpora probabilidades de riesgos y los costos o beneficios de cada camino lógico de sucesos y decisiones futuras. La resolución del árbol de decisión permite ver cuál decisión produce el mayor valor esperado al tomar la decisión, cuando todas las implicadas son ciertas, costos, beneficios y decisiones inciertas subsecuentes son cuantificadas.

análisis cuantitativo porque en este modelo no se explica con exactitud los criterios con los que se trabaja en el análisis cuantitativo de los riesgos y los criterios para obtener los riesgos priorizados y no son precisos o cuantitativos.

1.5.2. Modelo de Hall

Creado en 1997, el modelo de Hall, se ajusta a los procesos definidos por Boehm y el de la IEEE en general (García, 2007).

Incluye los procesos involucrados, la infraestructura necesaria para la gestión de riesgos, su implementación, y casos de estudios. Hace mucho énfasis en las características de las personas, los procesos la infraestructura y la implementación de la gestión de riesgos. Este modelo tiene muy poca documentación pública.

1.6. Modelos Matemáticos

1.6.1. Microsoft Solution Framework (MSF)

MSF es un compendio de las mejores prácticas en cuanto a administración de proyectos se refiere. Más que una metodología rígida de administración de proyectos, es una serie de modelos que pueden adaptarse a cualquier proyecto de tecnología de información.

MSF es un marco de trabajo propuesto por Microsoft, que cubre el ciclo de vida del proyecto y propone métodos y herramientas para el desarrollo del mismo. Dentro de sus disciplinas proponen Risk Management Discipline, como un mecanismo para enfrentar la incertidumbre en los proyectos, y proponen un enfoque proactivo donde continuamente se evalúen los riesgos.

Cuenta con varias fases que estructuran su desarrollo y una de ellas es el análisis de los riesgos. Este es el segundo paso dentro de Risk Management Discipline.

En la fase de análisis y priorización se trabaja con valores probabilísticos. Durante este proceso se convierten los datos del riesgo en un formulario utilizable en la toma de decisiones, y se priorizan los riesgos obteniendo un ranking para centrarse posteriormente en los más altos riesgos.

Durante esta etapa se analiza la probabilidad del riesgo. Para ello MSF define una escala con valores probabilísticos, verbales y numéricos para utilizar: En la Tabla 2 se muestra la probabilidad de ocurrencia de los riesgos, para MSF.

Tabla 2 Probabilidad de ocurrencia de los riesgos

Rango de Probabilidad	Valor de Probabilidad usado para cálculos	Expresión en Lenguaje Natural	Valor Numérico
-----------------------	---	-------------------------------	----------------

1% al 33%	17%	Bajo	1
34% al 67%	50%	Medio	2
68% al 99%	84%	Alto	3

La asignación de valores numéricos es un aspecto importante, pues calculan la exposición del riesgo a partir de estos valores. Respecto a otros modelos, la asignación de valores numéricos es un aspecto positivo en MSF.

También en este proceso se determina el impacto del riesgo, es decir la magnitud de la pérdida o la ganancia para determinados objetivos. MSF también propone una escala de valores para identificar la magnitud en cuanto a costo y tiempo (Ver Tabla 3).

Tabla 3 Impacto esperado en el costo

Valor Numérico	Pérdida Monetaria
1	Menos de \$100
2	\$100-\$1000
3	\$1000-10000
4	\$10000-100000
5	\$100000-\$1000000
6	\$1000000-\$10 millones
7	\$10 millones-\$100 millones
8	\$100 millones-\$1 billón
9	\$1 billón-\$10 billones
10	Más de \$10 billones

A cada valor de la escala se le asignan valores numéricos lineales, no utilizan ninguna otra “función”, lo que indicaría el grado de aceptación de los valores, y puede resultar interesante en el análisis de los riesgos.

MSF incluye un elemento interesante: la exposición del riesgo. La exposición de un riesgo es una medida de su amenaza, y se define una función para calcular la exposición dependiendo del grado de aceptación de la entidad. MSF propone una forma de calcularla y destaca que la utilización de una matriz (probabilidad x impacto) ayuda a los que toman decisiones, pues se puede hacer una escala de colores para ilustrar las exposiciones altas o bajas, como puede apreciarse en la Tabla 4.

Tabla 4 Matriz de probabilidad e impacto

Probabilidad-Impacto	Bajo=1	Medio=2	Alto=3
Alto=3	3	6	9
Medio=2	2	4	6
Bajo=1	1	2	3

Exposición Baja= 1 o 2, Exposición Media=3 o 4, Exposición Alta=6 o 9

Los riesgos se deben ordenar por la exposición, y adicionalmente plantean una fórmula para que determinadas propiedades de los riesgos influyan en su posición del Rank en mayor o menor medida, tal como se muestra en la Tabla 5.

Tabla 5 Fórmula para calcular la exposición

Valor de Ranking=0.5(probabilidad x impacto)-0.2(cuando necesitó)+0.3(costo de control x control de probabilidad con que trabajará))
--

Como resultado de este proceso se obtiene la “master risk list” que contienen una lista de los riesgos con los atributos primarios identificados y evaluados.

Adicionalmente definen otros métodos y otra representación de los riesgos en esta etapa, como se muestra en la Tabla 6.

Tabla 6 Representación del riesgo

Términos	Propósito	Estado
Declaración del riesgo	Articular claramente un riesgo	Requerido
Probabilidad	Cuantificar la probabilidad de ocurrencia	Requerido
Impacto	Cuantificar la severidad de pérdida o magnitud de coste de oportunidad	Requerido
Criterio de Jerarquización	Medida de importancia	Requerido
Prioridad (rango)	Dar prioridad a las acciones	Requerido
Propietario	Garantizar la continuidad a través del plan de acción de riesgo	Requerido
Plan de Mitigación	Describir medidas preventivas	Requerido
Plan de Contingencia	Describir medidas correctivas	Requerido
Causa Raíz	Guiar la planeación de intervención efectiva	Opcional
Efecto de Caída	Garantizar una estimación de impacto apropiada	Opcional
Contexto	Documentar la información antecedente para capturar el intento del equipo en riesgo de superficie	Opcional
Tiempo de implementación	Capturar la importancia de los controles de riesgo implementados sin un cierto plazo	Opcional

Métodos matemáticos adicionales

Este marco de trabajo, creado por Microsoft, propone una disciplina para el análisis de riesgos, basada principalmente en modelos matemáticos: Risk Management Discipline; aunque posee como técnicas adicionales de análisis de riesgos el análisis de árbol de decisiones, una de las características de los modelos matemáticos. Estas herramientas deben utilizarse si el equipo cree que pueden aportar algo positivo a la asignación de prioridades o al proceso de planeamiento.

Otras de las técnicas o métodos matemáticos adicionales son: el análisis causal⁴ y el análisis de Pareto⁵. Como se menciona anteriormente en este trabajo, la decisión de usar estas herramientas depende de las decisiones y propósitos del equipo.

⁴ El análisis causal es un análisis causa-efecto. Mediante el análisis se determina el alcance y el tamaño de un problema.

Este método toma la gestión de riesgos como una disciplina diseñada para ayudar al equipo a identificar las prioridades, a tomar las decisiones estratégicas correctas, y a controlar las emergencias que puedan surgir. Proporciona un entorno estructurado para la toma de decisiones y acciones, valorando los riesgos que se puedan provocar, aunque también se enfoca en asignar valores cualitativos.

1.6.2. Project Management Body of Knowledge (PMBOK)

El PMBOK, es un cuerpo de conocimiento para la gestión de proyectos y estándar nacional para los Estados Unidos desde 2001, que propone una guía genérica básica para la gestión de los riesgos.

El análisis forma parte de su proceso sistemático de la gestión de riesgos, dividido en análisis cualitativo y análisis cuantitativo. Incluye maximizar la probabilidad y consecuencias positivas y minimizar las negativas.

El análisis cuantitativo se realiza a los riesgos priorizados en el proceso anterior, o sea, el cualitativo, analiza el efecto de estos riesgos y les asigna una calificación numérica. Proponen técnicas para cuantificar los posibles resultados del proyecto y sus probabilidades (por concepto de los riesgos identificados), evaluar la probabilidad de cumplir un objetivo, cuantificar el riesgo general del proyecto e identificar objetivos viables por concepto de riesgos.

En la Figura 3 se detallan las entradas, herramientas y técnicas, y salidas del análisis cuantitativo.

Se utiliza la técnica de entrevista para cuantificar la probabilidad y el impacto, distribución de probabilidades, árbol de decisiones y modelado, entre otras. Estas técnicas definen las características de PMBOK, que hacen clasificarlo como un modelo basado en la experiencia. En general estas técnicas pueden ser muy efectivas, el PMBOK las menciona y aunque en muchos casos se ejemplifican aplicadas al análisis cuantitativo de la gestión de riesgos, no queda claro cómo utilizarlas durante todo el proceso.

⁵ El análisis de Pareto es una comparación cuantitativa y ordenada de elementos o factores relativos a un problema, según su contribución a determinado efecto. Esta técnica es valiosa en la asignación de prioridades a los problemas de calidad, en el diagnóstico de causas y en la solución de las mismas.

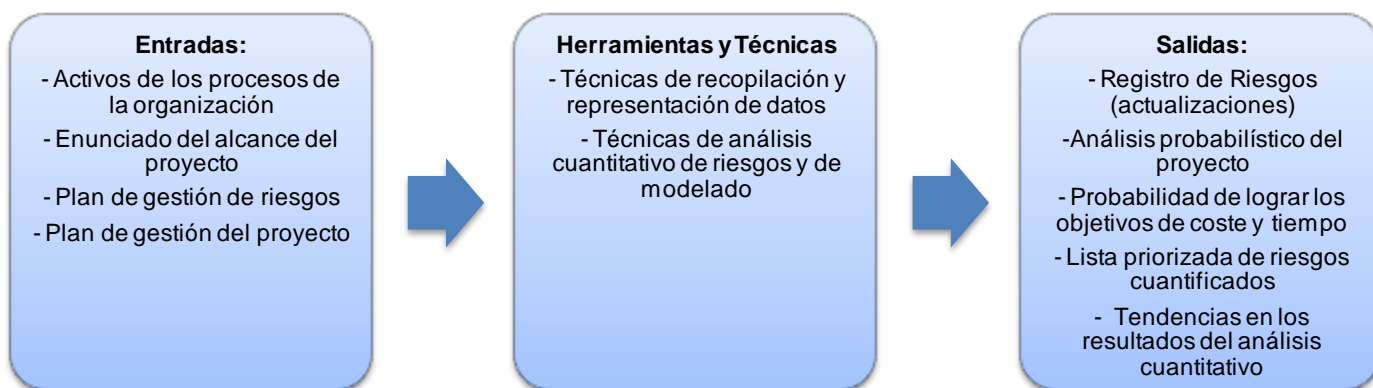


Figura 3 Entradas, Herramientas y Salidas del Análisis Cuantitativo

Este modelo también usa otras técnicas basadas en modelos matemáticos, tal es el caso del análisis del valor monetario esperado y del análisis del árbol de decisiones.

A pesar de ser este modelo uno de los más completos y abarcadores, posee la desventaja de carecer de una explicación exacta de cómo se usarían estas técnicas.

Usa la técnica de análisis de árbol de decisiones para:

- Cuantificar los posibles resultados del proyecto y sus probabilidades.
- Evaluar la probabilidad de lograr los objetivos específicos del proyecto.
- Identificar los riesgos que requieren una mayor atención mediante la cuantificación de su contribución relativa al riesgo general del proyecto.
- Identificar objetivos de coste, cronograma o alcance realistas y viables, dados los riesgos del proyecto.
- Determinar la mejor decisión de dirección de proyectos cuando algunas condiciones o resultados son inciertos.

Este es uno de los modelos más completos, cuenta con un análisis cuantitativo definido, aunque no del todo, ya que no explica exactamente cómo sería, ni cómo se usan cada una de las técnicas o herramientas.

1.6.3. Modelo de Pressman

Es un procedimiento de gestión de riesgos, donde el segundo de sus procesos es llamado Proyección o Evaluación del Riesgo. (Pressman)

Dentro de este proceso, se plantea que el jefe de proyecto, junto a otros gestores y personal técnico, realiza cuatro actividades de proyección del riesgo:

- Establecer una escala que refleje la probabilidad percibida del riesgo.

- Definir las consecuencias del riesgo.
- Estimar el impacto del riesgo en el proyecto y en el producto.
- Apuntar la exactitud general de la proyección del riesgo de manera que no existan confusiones.

Estos se pueden auxiliar de una tabla de riesgos, que podría implementarse como un modelo de hoja de cálculo para lograr un fácil manejo y organización de las entradas. En la primera columna listar los riesgos, en la siguiente la probabilidad de aparición de cada riesgo, que puede estimarse por cada miembro individual del proyecto. Se sondea a los miembros del equipo individualmente de un modo rotativo (Round-Robin⁶) hasta que comience a converger su evaluación sobre la probabilidad del riesgo.

A continuación se valora el impacto de cada riesgo y se determina una categoría de impacto. Las categorías para cada uno de los cuatro componentes de riesgo -rendimiento, soporte, coste y planificación temporal- son promediadas para determinar un valor general de impacto. Una vez que se han completado las cuatro primeras columnas de la tabla de riesgo, la tabla es ordenada por probabilidad y por impacto. Los riesgos de alta probabilidad y de alto impacto pasan a lo alto de la tabla. Esto consigue una priorización del riesgo de primer orden.

El jefe del proyecto estudia la tabla ordenada resultante y define una línea de corte. La línea de corte (dibujada horizontalmente desde un punto en la tabla) implica que solo a los riesgos que quedan por encima de la línea se les prestará atención en lo adelante. Los riesgos que caen por debajo de la línea son reevaluados para conseguir una priorización de segundo orden.

Como plantea Pressman, el impacto del riesgo y la probabilidad tienen diferente influencia en la gestión. Un factor de riesgo que tenga un gran impacto pero muy poca probabilidad de que ocurra no debería absorber una cantidad significativa de tiempo de gestión. Sin embargo, los riesgos de gran impacto con una probabilidad moderada a alta y los riesgos de poco impacto, pero de gran probabilidad, deberían tenerse en cuenta en los procedimientos de análisis de riesgos posteriores. (Pressman)

La probabilidad de riesgo puede determinarse haciendo estimaciones individuales y desarrollando después un único valor de consenso. Aunque este enfoque es factible, se han desarrollado técnicas más sofisticadas para determinar la probabilidad de riesgo. Pressman propone que los controladores de riesgo pueden valorarse en una escala de probabilidad cualitativa que puede tener los siguientes valores: imposible, improbable, probable y frecuente. Después puede asociarse una probabilidad

⁶ Round Robin es un método para seleccionar todos los elementos en un grupo de manera equitativa y en un orden racional, normalmente comenzando por el primer elemento de la lista hasta llegar al último y empezando de nuevo desde el primer elemento. (Mora, y otros, 2008)

matemática con cada valor cualitativo (por ejemplo: una probabilidad del 0.7 al 1.0 implica un riesgo muy probable). Luego el autor propone una evaluación del impacto del riesgo, teniendo en cuenta: su naturaleza, su alcance y cuándo ocurre, y calculando la exposición al riesgo. La naturaleza del riesgo indica los problemas probables que aparecerán si ocurre. Por ejemplo, una interfaz externa mal definida para el hardware del cliente (un riesgo técnico) impedirá un diseño y pruebas tempranas y probablemente lleve a problemas de integración más adelante en el proyecto. El alcance de un riesgo combina la severidad (¿cuán serio es el problema?) con su distribución general (¿qué proporción del proyecto se verá afectado y cuántos clientes se verán perjudicados?). Finalmente, la temporización de un riesgo considera cuándo y por cuánto tiempo se dejará sentir el impacto.

Fórmula para calcular la exposición al riesgo en general (ER):

$$ER = P * C$$

Donde P es la probabilidad de que ocurra un riesgo, y C es el coste del proyecto si el riesgo ocurriera. Por ejemplo: Identificación del riesgo: Solo el 70% de los componentes del software planificados para reutilizarlos pueden, de hecho, integrarse en la aplicación. La funcionalidad restante tendrá que ser desarrollada de un modo personalizado.

Probabilidad del riesgo: 80% (probable).

Impacto del riesgo: 60 componentes de software reutilizables fueron planificados. Si solo el 70% pueden usarse, 18 componentes tendrán que desarrollarse improvisadamente (además de otro software personalizado que ha sido planificado para su desarrollo). Puesto que la media por componentes es 100 líneas de código (LDC) y los datos locales indican que el coste de la ingeniería del software para cada LDC es de \$ 14.000; el coste global (impacto) para el desarrollo de componentes sería $18 * 100 * 14 = \$25.200$.

Exposición al riesgo: $ER = 0.80 * 25.200 \sim \20.200 .

Plantea además que: "La experiencia dice que el 80% del riesgo total de un proyecto (por ejemplo: el 80% de la probabilidad de fracaso del proyecto) se debe solamente al 20% de los riesgos identificados."

La exposición al riesgo se puede calcular para cada riesgo en la tabla de riesgos, una vez que se ha hecho una estimación del coste del riesgo. La exposición al riesgo total para todos los riesgos (sobre la línea de corte en la tabla de riesgos) puede proporcionar un significado para ajustar el coste final estimado para un proyecto. También puede ser usado para predecir el incremento probable de recursos de plantilla necesarios para varios puntos durante la planificación del proyecto.

El equipo del proyecto deberá volver a la tabla de riesgos a intervalos regulares, volver a evaluar cada riesgo para determinar qué nuevas circunstancias hayan podido cambiar su impacto o probabilidad.

Como consecuencia de esta actividad, puede ser necesario añadir nuevos riesgos a la tabla, quitar algunos que ya no sean relevantes y cambiar la posición relativa de otros.

Por último, en este proceso de análisis del riesgo, Pressman propone una evaluación del riesgo, estableciendo un conjunto de ternas de la forma $[r_i, l_i, x_i]$, donde r es el riesgo, l es la probabilidad del riesgo, y x es el impacto del riesgo. En esta evaluación del riesgo se sigue examinando la exactitud de las estimaciones que fueron hechas durante la estimación o proyección del riesgo, se intenta dar prioridades a los riesgos que no se habían cubierto, y se empieza a pensar las maneras de controlar y/o impedir los riesgos que sean más probables de aparecer. Para que sea útil la evaluación, se debe definir un nivel de referencia de riesgo. Hay un nivel para la degradación del rendimiento, exceso de coste, dificultades de soporte o retrasos de la planificación temporal (o cualquier combinación de los cuatro) que provoquen que se termine el proyecto. Un nivel de referencia de riesgo tiene un solo punto, denominado punto de referencia o punto de ruptura, en el que la decisión de seguir con el proyecto o dejarlo, por ser los problemas muy graves, son de igual modo aceptables, pues una vez que la exposición al riesgo supera el nivel de referencia, el proyecto debe ser terminado.

Este modelo es abarcador y es uno de los que mejor esclarece su uso en un proyecto, específicamente en un proyecto de software.

1.6.4. Software Risk Evaluation (SRE)

Es un método para identificar, analizar, evaluar y desarrollar estrategias de mitigación para los riesgos; es una herramienta de diagnóstico para crear una línea base de riesgos para un proyecto. El SEI lo propone como método a utilizar en conjunto con CRM y Team Risk Management. Es más efectivo utilizarlo como el iniciador de CRM. (Williams, 2003)

Una de sus fases es el análisis de los riesgos, donde estos son analizados, categorizados en áreas, y priorizados según lo define el método. Luego son entregados al gerente del proyecto. En esta fase, además de identificarse la condición y consecuencia de un riesgo, se evalúa el riesgo en cuanto a su exposición, determinando previamente un promedio entre los criterios de impacto y probabilidad de ocurrencia del riesgo. Propone un análisis continuo mediante sesiones de análisis en grupos.

Method Description, V2.0 es un Software de Evaluación de Riesgos (SRE), propuesto por Ray C. Williams, George Pandelios y Sandra G. Behren. Es un software propietario que permite identificar, analizar y desarrollar estrategias de mitigación de riesgos en un sistema de software que se encuentre en desarrollo.

Este modelo posee poca precisión ya que se basa en criterios de los participantes y no en técnicas más acreditadas y probadas.

1.7. Modelos Basados en la Simulación

1.7.1. PMBOK

Este modelo también usa técnicas basadas en simulación, tal es el caso del análisis de sensibilidad.

Usa técnicas tales como la simulación de Monte Carlo para:

- Cuantificar los posibles resultados del proyecto y sus probabilidades
- Evaluar la probabilidad de lograr los objetivos específicos del proyecto
- Identificar los riesgos que requieren una mayor atención mediante la cuantificación de su contribución relativa al riesgo general del proyecto
- Identificar objetivos de coste, cronograma o alcance realistas y viables, dados los riesgos del proyecto
- Determinar la mejor decisión de dirección de proyectos cuando algunas condiciones o resultados son inciertos.

De igual modo que las técnicas mencionadas anteriormente de PMBOK, estas técnicas de simulación carecen de una explicación concreta sobre su uso.

1.7.2. Microsoft Solution Framework (MSF)

Este marco de trabajo, expuesto anteriormente, posee como otras técnicas adicionales de análisis de riesgos la simulación y el análisis de sensibilidad⁷, dos de las características de los modelos basados en la simulación. Como se había mencionado anteriormente, la decisión de usar estas herramientas debería ser basada en el valor que el equipo desee obtener para el proceso de priorización y planificación del costo de los recursos.

1.7.3. Principales Tendencias

En la Figura 4 puede observarse un resumen de la clasificación de los principales modelos según sus tendencias o técnicas propuestas. Los Modelos de Boehm y de Hall son basados en la experiencia ya que proponen como una de sus técnicas el análisis de redes. MSF, PMBOK, SRE y el Modelo de Pressman son modelos matemáticos porque exponen las técnicas como el análisis causal, análisis de

⁷ El análisis de sensibilidad ayuda a determinar qué riesgos tienen el mayor potencial de impacto en el proyecto. Este método evalúa el grado en que la incertidumbre de cada elemento del proyecto afecta el objetivo que está siendo examinado, cuando los demás elementos son mantenidos en sus valores básicos. (PMI, 2004)

árbol de decisiones, análisis del valor monetario esperado, análisis de Pareto, entre otros. PMBOK y MSF, también poseen técnicas que los incluyen en la clasificación de modelos basados en la simulación porque exponen técnicas como el análisis de sensibilidad y la simulación de Monte Carlo.

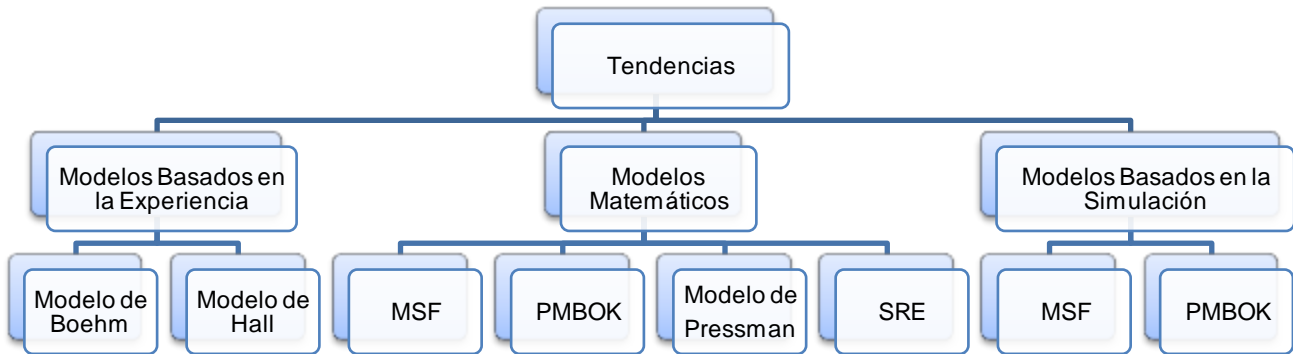


Figura 4 Tendencias de los principales modelos

1.8. Análisis Cuantitativo de Riesgos en la UCI

La gestión de riesgos en la UCI posee documentos oficiales para algunos de sus elementos, uno de ellos es el “Manual de Procedimientos para el Diagnóstico de Riesgos”.

El procedimiento no se ajusta a proyectos productivos de software donde se requiere un mayor nivel de detalle y clasificación en los riesgos, así como técnicas objetivas y precisas para su análisis cualitativo y cuantitativo, para lograr más valor en el proyecto y la gestión del conocimiento adquirido de los riesgos.

La Universidad no cuenta en estos momentos con un sistema que gestione el análisis cuantitativo de los riesgos, que tan útil es, ni existe un antecedente a la propuesta que se quiere establecer.

Para la frecuencia e impacto no se definen técnicas para su identificación o cálculo.

En la UCI, también a lo largo de sus dos graduaciones anteriores, se han desarrollado varias tesis de pregrado que abarcan el tema de la gestión de riesgos, y que han sido validadas mediante la valoración de expertos o su aplicación.

En la tesis de pregrado “Propuesta de procedimiento para el desarrollo y aplicación de la Gestión del Riesgo en proyectos de producción de software” (Susé, 2007), se hace un estudio del estado del arte de la gestión de riesgos en el mundo, se identifican los riesgos según su categoría, se propone una evaluación global del riesgo del proyecto, componentes y controladores del riesgo, proyección del riesgo (evaluación del impacto del riesgo (clasificados en catastrófico, crítico, marginal o despreciable), y evaluación del riesgo), refinamiento de los riesgos; reducción, supervisión y gestión del riesgo;

riesgos y peligros para la seguridad, y por último se propone un procedimiento para la gestión de riesgos. Pero este se hace de manera general, por lo que no abarca el análisis cuantitativo de los riesgos en sí.

No obstante se propone una parte del procedimiento propuesto por Pressman para la estimación y evaluación del riesgo.

También en la tesis de pregrado *“Propuesta de una guía para la planificación de los proyectos productivos en la Facultad 3 de la UCI”* (Benito, 2007), se hace un estudio del estado del arte de tres metodologías para la gestión de riesgos (PMI, SEI-CRM, y la propuesta por Pressman), y de algunas herramientas para la gestión de riesgos, pero no se profundiza en la fase de análisis, ni específicamente en el análisis cuantitativo. Aunque sí se indican algunos métodos de gestión de riesgos u organizaciones: Eurométodo, Safe, SEI-CRM, RiskIt y los métodos para la gestión de riesgos del IEEE y del PMI (Project Management Institute) y se plantea que en cada método se establecen categorías para funciones del riesgo en diferentes fases.

En *“Propuesta para la Gestión de Riesgo en los proyectos productivos de la UCI”* (López, y otros, 2007), se hace un estudio muy universal de la gestión de riesgos en las metodologías RUP (Rational Unified Process), MSF (Microsoft Solution Framework) y XP (Extreme Programming), y de manera general para todo proyecto de software, según diversos autores. Sobre el análisis cuantitativo se menciona, acertadamente, sus posibles métodos, según el Servicio Nacional de Aduanas de Chile, tales como: análisis de probabilidad, de consecuencias, estadístico/numérico, de redes, de decisión, de costo del ciclo de vida, modelado o simulación computacional, árboles de falta y análisis de árbol de eventos, diagramas de influencia, entre otros, pero no se explica en qué consisten ninguno de ellos.

Luego, basándose en lo que plantea RUP, se realiza una propuesta para el análisis de los riesgos donde la entrada es una lista de riesgos y la salida es una caracterización de los riesgos y se mencionan los atributos en el análisis de riesgos propuesto por RUP: impacto, probabilidad de ocurrencia (en por ciento) y exposición al riesgo (impacto*probabilidad de ocurrencia), que se puede expresar en semanas o días. Se propone que este análisis lo debe hacer el equipo de administración de riesgo, que deben analizar los riesgos según la probabilidad, calcular la exposición al riesgo, valorar el impacto y establecer una categoría, y realizar un ordenamiento según la probabilidad y el impacto. Pero de igual modo, se hace más bien del punto de vista cualitativo y no se explica con exactitud cómo debería ser un proceso de análisis cuantitativo de los riesgos.

En la tesis de pregrado titulada *“Análisis de Riesgos en la Planificación de proyectos Informáticos para el Sistema de Salud Cubano en la Facultad 7”* (Ávila, 2007), tampoco se propone explícitamente cómo sería un análisis cuantitativo de los riesgos, solo se mencionan algunos de los pasos a tener en

cuenta, tales como: establecer una escala que refleje la probabilidad percibida del riesgo, definir las consecuencias del riesgo, estimar el impacto del riesgo y en el producto, y apuntar la exactitud general de la proyección del riesgo. Sin embargo se enfoca en el punto de vista cualitativo.

En la tesis de pregrado *“Gestión de Riesgos en el proyecto de Informatización del Conocimiento Geológico en Cuba”* (Cuza, y otros, 2007), se hace una comparación entre los modelos: ISO/IEC 12207, CMMI y MoProSoft, de acuerdo a lo que cada uno plantea sobre la gestión de riesgos. En esta comparación puede verse cómo estos dos últimos modelos son los que plantean que debe haber un análisis cuantitativo en este proceso. Se proponen el uso de algunas métricas para medir la efectividad del proceso de gestión de riesgo, pero no proponen cómo sería el análisis cuantitativo.

En la tesis de pregrado *“Propuesta de modelo para la Gestión de Riesgos en los proyectos de producción de software”* (García, 2008) se propone de manera general cómo debería ser este proceso de gestión, basados en el modelo propuesto por PMBOK, y aunque la propuesta es bastante abarcadora, pues propone entradas, participantes, actividades, algunas técnicas y salidas, en el análisis cuantitativo no se hace una completa explicación de este proceso. Sólo se explican las técnicas: entrevista y juicio de expertos.

En la tesis de pregrado *“Una guía para el tratamiento de riesgos para el software educativo en la Universidad de las Ciencias Informáticas”* (Hidalgo, y otros, 2007) se aborda la gestión de riesgos en la UCI, sus debilidades y fortalezas. Se definen los niveles de atención a los riesgos, quién se encargará de estos riesgos y las etapas que tendrá el proceso y cómo se realizará la identificación de los riesgos. En cuanto al análisis, se definen las entradas técnicas y herramientas del proceso. En cuanto a las técnicas, una de las propuestas es la matriz de probabilidad de impacto, pero de forma literal de acuerdo a una clasificación definida. No se desarrolla el tema del análisis cuantitativo de los riesgos. Sólo se menciona someramente el análisis de los riesgos.

En *“Análisis y Gestión de Riesgo para el desarrollo de las aplicaciones del proyecto Atención Primaria de Salud (APS)”* (Figueredo, 2007), se expone detalladamente la metodología de gestión de riesgos desarrollada por el Instituto de Administración de Proyectos (PMI). Se determinan los riesgos predominantes en el proyecto APS siendo identificados y evaluados. Se proponen las medidas necesarias para que sean mitigados, tomando como base la metodología desarrollada por PMI. En esta tesis no se desarrolla el tema de análisis cuantitativo de riesgos, pues sólo se hace un análisis cualitativo, se menciona que también existe el análisis cuantitativo, y se explican brevemente los conceptos de impacto y probabilidad de ocurrencia del riesgo, así como que los que van a resultar más importantes son aquellos cuya importancia es alta, y cuya probabilidad de ocurrencia también es alta, y posteriormente se pasa a planificar la respuesta al riesgo. Se comparan algunas metodologías y

métodos existentes en cuanto al proceso de gestión de riesgos que abarcan, tales como: Eurométodo, Safe, SEI, IEEE, Riskit y PMI.

También es “*Análisis del Método de Estimación empleado para el desarrollo del proyecto SIGEP*” (Rodríguez, y otros, 2007), se trata el tema de la gestión de riesgos y se centra en realizar un análisis del método de estimación, teniendo en cuenta los elementos que debe contemplar y los resultados obtenidos de la aplicación del método analizado, para identificar sus deficiencias y proponer mejoras para su futura aplicación. En este trabajo se aborda cómo pueden influir los riesgos en un método de estimación y como gestionarlos, explica cómo identificarlos, analizándolos de forma independiente de acuerdo a su clasificación (del negocio, relacionados con el cliente, del proceso, tecnológicos, del entorno de desarrollo, asociados con la experiencia de los desarrolladores). En cuanto al método de estimación, se asignan valores a la categoría y a la probabilidad e impacto de los riesgos, en este último se definen los valores de impacto en catastrófico, crítico, marginal y despreciable, pero no se explica cómo se obtuvo el valor de probabilidad e impacto del riesgo. También se aborda la complejidad de acuerdo a los roles y casos de uso. Esta estimación se hace basada en valores cualitativos y no en técnicas concretas de análisis cuantitativo, por lo que este proceso no se aborda claramente.

En “*Aplicación de una estrategia para la planificación de la gestión de riesgos en un proyecto caso de estudio*” (López, 2008), se hace un estudio de los principales enfoques a nivel internacional acerca de la gestión de riesgos, pero no se abarca cómo sería el análisis de los riesgos en el proceso de planificación de los riesgos.

En *Gestión de Riesgo en el Proyecto Sistema de Gestión Fiscal* (Hechavarría, y otros, 2008), se realiza un análisis cuantitativo de los riesgos, basados en el proceso propuesto por PMBOK, aunque no se explica cómo se calculó para dicho caso la probabilidad ni cómo se computó el impacto. Se usó la técnica de exposición al riesgo, que es el caso más simple de análisis cuantitativo, basados en cálculos matemáticos ($\text{exposición} = \text{probabilidad} * \text{impacto}$), en este caso el impacto está dado en unidades de tiempo (retraso) y en unidades monetarias (incremento de costes), como se plantea en el modelo de Pressman.

En *Modelo de Gestión de Riesgos y análisis de propuesta de aplicación para la ejecución del mismo* (Moreira, 2008), se propone también un modelo para la gestión de riesgos muy similar al propuesto por PMBOK. También se proponen técnicas para el análisis cuantitativo de los riesgos y se explican someramente las definiciones de cada una, tales como: entrevistas, análisis de sensibilidad, análisis

del árbol de decisiones, simulación y análisis de riesgos de costos; pero no se especifica cómo se implementarían estas técnicas.

En la tesis *Herramienta para el análisis y gestión de riesgos de la seguridad informática para BANDEC* (Corzo, 2008), se hace un estudio de la metodología MAGERIT, y se propone la herramienta desarrollada para BANDEC, de acuerdo a sus necesidades. En esta herramienta se calcula el riesgo, pero no se especifica cómo ni mediante cuál técnica u operación matemática.

En *Guía Metodológica para Administrar los Riesgos en los proyectos productivos de Realidad Virtual* (Ramírez, y otros, 2008), se propone como uno de los procesos de la guía metodológica, el análisis de los riesgos. En este se estima la probabilidad de ocurrencia del riesgo y su impacto, pero se lleva a cabo en base a valores cualitativos y los valores cuantitativos se hacen corresponder con un valor cualitativo, por ejemplo, si el riesgo es poco probable, su probabilidad tiene un valor entre 0.1 y 0.4, si el impacto es marginal, tiene valor 1. Luego se calcula el factor de riesgo, donde se plantea que este es igual a la suma de la probabilidad más el impacto. Por ello el análisis cuantitativo no es exacto y preciso.

1.9. Herramientas

Existen una gran cantidad de herramientas software de gestión de riesgos, específicamente para el análisis cuantitativo, disponibles en el mercado, las cuales siguen diferentes metodologías. Las más significativas son las siguientes:

1.9.1. EAR/PILAR

Software que pone en práctica y amplía la Metodología MAGERIT (Ver epígrafe 1.4.10) y es diseñado para apoyar el proceso de gestión de riesgos a lo largo de períodos largos, proporcionando el análisis incremental.

La herramienta EAR, también llamada: entorno de análisis de riesgos, proporciona cálculos rápidos, y genera una gran cantidad de texto y gráfico. (Ver Anexo 2: Herramienta EAR/PILAR).

Plantean que los activos están expuestos a amenazas que, cuando se materializan, degradan el activo, produciendo un impacto. Si se estima la frecuencia con que se materializan las amenazas, se puede deducir el riesgo al que está expuesto el sistema.

La herramienta posee servicios de salvaguardas, que o bien reducen la frecuencia de ocurrencia, o bien reducen o limitan el impacto. Dependiendo del grado de implantación de estas salvaguardas, el

sistema pasa a una nueva estimación de riesgo que se denomina riesgo residual (riesgos en el sistema tras la implantación de las salvaguardas determinadas en el plan de seguridad de la información).

Dispone de una biblioteca estándar de propósito general, y es capaz de realizar calificaciones de seguridad respecto a normas ampliamente conocidas como: ISO/IEC 27002:2005 (Código de buenas prácticas para la Gestión de la Seguridad de la Información); SP800-53:2006 (Recommended Security Controls for Federal Information Systems o Controles de Seguridad Recomendados para Sistemas de Información Federal); y Criterios de Seguridad, Normalización y Conservación del Consejo Superior de Informática, y para el Impulso de la Administración Electrónica, de España).

Los resultados obtenidos por la herramienta aportan una versión simplificada de complejidad del problema, mediante la que es posible comprender mejor el mismo y adoptar conclusiones razonadas.

Proveedor: Centro Criptológico Nacional (España).

Idioma: español, inglés, francés, portugués, italiano y otros.

Sistemas Operativos: Windows, Unix.

Licenciamiento: Comercial.

Última versión: 4.3 (2009).

1.9.2. Technical Risk Identification and Mitigation System (TRIMS)

Herramienta integrada de Gestión de Riesgos que emplea ingeniería de conocimientos y se enfoca en la identificación, medición y seguimiento de riesgos técnicos de proyectos.

Proveedor: Best Manufacturing Practices.

Plataforma: Win32.

Sistemas Operativos: Windows 98, ME, NT, 2000, y XP.

Idioma: inglés.

Licenciamiento: Libre.

Posee gerenciamiento integrado de riesgos y generación de reportes.

Orientada a categorías de riesgos predefinidas para sectores específicos.

1.9.3. @Risk

Herramienta de Inteligencia de Negocios, que muestra todos los resultados posibles de una situación de negocios e indica la probabilidad de que ocurran.

Proveedor: Palisade.

Sistemas Operativos: Windows 98, 2000 y XP.

Idioma: Español, inglés, alemán, francés y japonés.

Licenciamiento: Comercial.

Funciona en Excel (compatible con las versiones de 2000 a 2007). Presenta resultados con buenos gráficos, análisis de escenarios, identifica las situaciones –o combinaciones que producen un resultado determinado, integración con herramientas estándares del mercado (MS Project), y variedad de eficaces funciones analíticas (Búsqueda de Objetivo, Análisis de Tendencia, Análisis Avanzado de Sensibilidad) (Ver Anexo 3: Herramienta @Risk). Ofrece un informe estadístico completo de las simulaciones, así como acceso a todos los datos generados. Realiza análisis de riesgo utilizando simulación de Monte Carlo para mostrar una gran cantidad de escenarios posibles en una hoja Excel. Se puede evaluar qué riesgos tomar y cuáles evitar. Puede contestar preguntas como: ¿Cuál es la probabilidad de que las utilidades sean más de \$10 millones? ¿Cuáles son las posibilidades de perder dinero en cierto proyecto?.

También existen @Risk for Project, @Risk Developer Kit, @Risk Accelerator Network Edition, @Risk Standard, @Risk Professional, @Risk Industrial.

Última versión: 5.0 (2009).

1.9.4. Chinchón

Proveedor: Free.

Plataforma: Java.

Licenciamiento: Libre.

Es una herramienta para analizar cuantitativamente el riesgo de un sistema de información. La herramienta sigue el modelo MAGERIT 1.0. Su versión 1.3 fue elaborada por el Dr. José Antonio Mañas, profesor de la Escuela Técnica Superior de Ingenieros de Telecomunicaciones de la Universidad Politécnica de Madrid, España. La entrada se escribe en XML, y realiza un análisis de la posición de riesgo, sirviendo de apoyo a su gestión. Los derechos de propiedad intelectual pertenecen al autor, quien ha puesto la herramienta en el dominio público.

1.9.5. Acertus, Securac`s Enterprise Risk Management (ERM)

Proveedor: Acertus.

Licenciamiento: Comercial.

Es una herramienta de soluciones de software que permite a las organizaciones identificar, medir, gestionar y mitigar los riesgos, y cumplir en todos los aspectos de la organización. Basada en la web, y con múltiples lenguajes, es una solución de software para la gestión de riesgos diseñada para mejorar la forma en que las organizaciones evalúan los riesgos, la seguridad y el cumplimiento regulatorio.

Provee herramientas en tiempo real para todas las operaciones y una interfaz única.

1.9.6. MB Risk Management (MBRM)

Se especializa en mostrar gráficos, para una mejor comprensión de todas las variables que afectan y modifican los valores de riesgo, pero no lleva por sí misma toda la información al respecto. Posee varias versiones de herramientas con este propósito, como UNIVSWAP, UNIVCONV, entre otras. (Ver Anexo 4: Herramienta MBRM).

1.9.7. SCRAM99

SCRAM99 es un software que permite evaluar el nivel del calendario y los riesgos de costo de un proyecto, identificando las áreas del proyecto que se beneficiarían con una mayor atención de gestión, identificando las tendencias de riesgos con el propósito de prevenir los problemas antes de que se produzcan, establece y mantiene el plan de contingencia, y analiza sus impactos potenciales. Es un software propietario. (Ingersoll, 2000.)

Es un software para ejecutar simulación de Monte Carlo en MS Project con MS Excel. Realiza análisis probabilístico de riesgos usando estimación fuzzy para predecir los resultados de un proyecto. Simula la ejecución de proyectos, estima la duración y costo con los criterios: Optimista, Más probable, y Pesimista. Define los puntos de decisión y uso de lógica de decisión para modelos de proyectos exactos con pruebas, y planes de contingencia.

1.9.8. Valoración sobre las herramientas actuales

Las herramientas actuales para el análisis cuantitativo de riesgos están caracterizadas por un efectivo y rápido análisis cuantitativo de riesgos. Presentan los resultados con gráficos que facilitan la comprensión y generan detallados reportes, los cuales le permiten al cliente saber qué puede o esté marchando mal en su organización.

La mayoría de estas herramientas poseen la desventaja de que no se comercializan libremente y el enfoque está definido más desde el punto de vista cualitativo que cuantitativo.

1.10. Conclusiones Parciales

Luego de haber realizado un estudio del arte de algunas de las herramientas y metodologías para gestionar los riesgos en un proyecto, es posible afirmar que no existen herramientas que brinden un

soporte adecuado y estandarizado que se adecue a las necesidades actuales de la gestión de riesgos.

A partir de este estudio se llegaron a las siguientes conclusiones:

- No están completamente definidos los indicadores necesarios para un adecuado análisis y estimación de riesgos de proyectos de software.
- Todo el enfoque en el análisis de los riesgos está pensado desde un punto de vista más cualitativo que cuantitativo.
- La mayor parte de los trabajos se hacen manualmente y los cálculos, aunque se use un ordenador, se hacen bajo demanda, de manera que los programas utilizados son pasivos.
- Las principales tendencias actuales de los modelos, métodos o técnicas para el análisis cuantitativo de riesgos son: modelos basados en la experiencia, basados en modelos matemáticos, y basados en la simulación.
- El análisis cuantitativo de riesgos debe ser el centro de la gestión de riesgos, un proceso imprescindible, más automatizado, más extensivo a todos los riesgos y, más proactivo, de manera que se reduzca el coste de gestión de cada riesgo pudiendo mitigarse, de este modo, más riesgos.

Las metodologías, estándares y modelos analizados pueden fusionarse y obtener lo mejor de cada una de ellas.

Capítulo 2: Modelo para el Análisis Cuantitativo de Riesgos

2.1. Introducción.

El análisis cuantitativo del riesgo sucede al análisis cualitativo del riesgo. Pretende la estimación del impacto de los riesgos de forma numérica cuantificable, generalmente respecto a las variables: costo, alcance, tiempo y calidad.

El siguiente modelo se basa en trabajos analizados en el Capítulo 1. Se tiene en cuenta la situación actual de la UCI, la problemática existente y las experiencias en el análisis cuantitativo de riesgos.

Se describen los procesos con sus entradas y salidas.

2.2. Características del Modelo

El modelo propuesto posee una serie de actividades que se deben realizar para obtener el valor de la probabilidad e impacto de los riesgos y documentar los resultados obtenidos. Las actividades son las siguientes, tal como se muestra en la Figura 5.

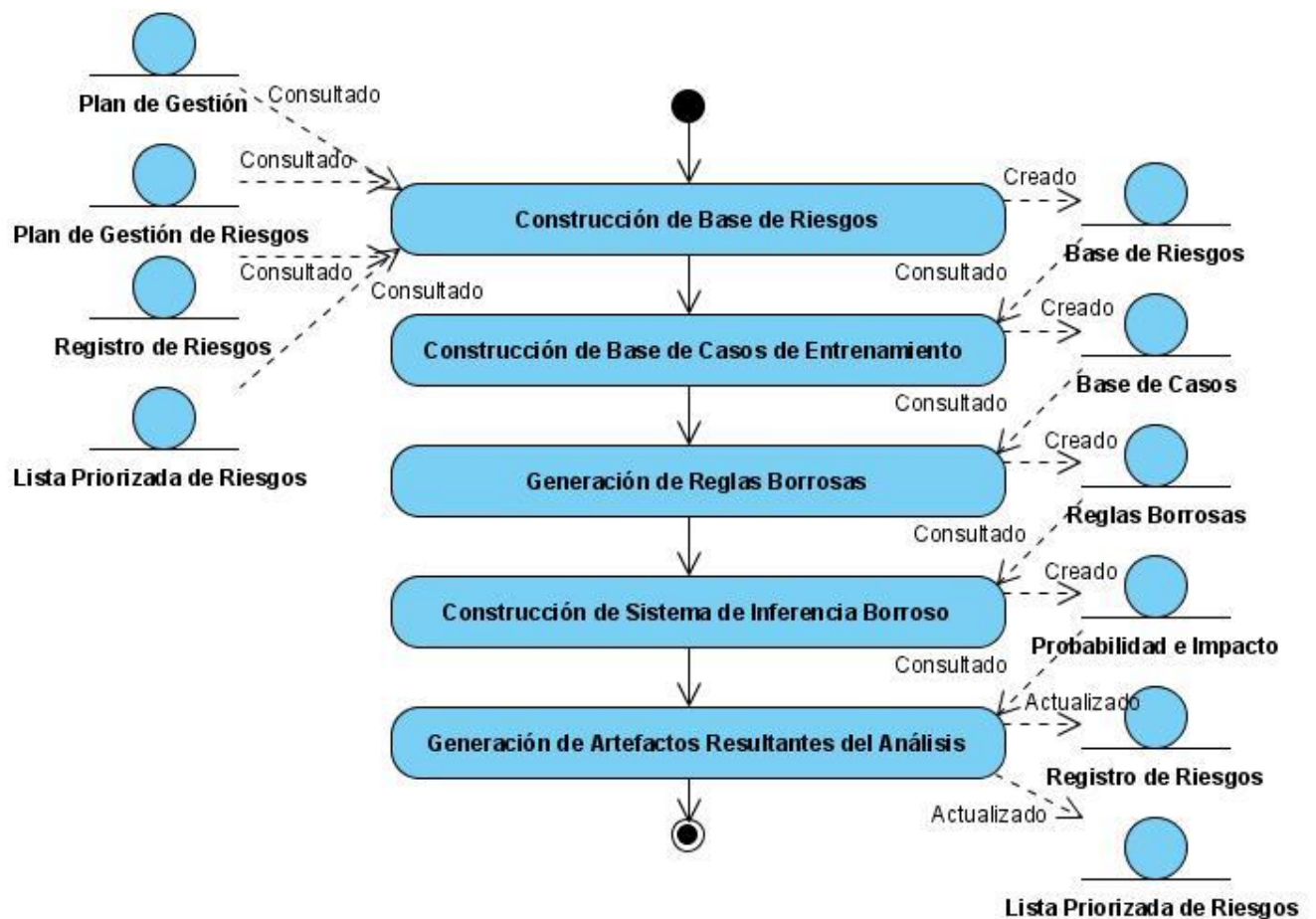


Figura 5 Modelo para Análisis Cuantitativo de Riesgos

A continuación se describen los elementos fundamentales del modelo:

- Construcción de la base de riesgos: para la correcta aplicación del modelo, basado en las técnicas de softcomputing previstas, es necesario contar con una base de riesgos, obtenidos principalmente, resultado de la experiencia de proyectos anteriores, los cuales a su vez contarán con un grupo de parámetros a través de los cuales serán evaluados.
- Construcción de la base de casos de entrenamiento: luego del análisis de los riesgos presentes en la base de casos, se realiza una selección de los más representativos para el proyecto en cuestión, de acuerdo con las características del mismo, así como de la certidumbre de la información de cada uno de los riesgos presentes, para conformar con estos datos una base de casos de entrenamiento.

- Generación de reglas borrosas: con la base de casos de entrenamiento obtenida se crean las reglas de aprendizaje, que permitirá a través de un conjunto de características históricas de riesgos presentes en proyectos anteriores evaluar un riesgo determinado.
- Construcción de la red y del sistema de inferencia borroso: utilizando las reglas de aprendizaje obtenidas, se evalúa un determinado riesgo obteniéndose como salida final la probabilidad y el impacto del riesgo en cuestión. Esto es posible debido a la creación de una red y que es usada por un sistema de inferencia borroso. Se propone la creación de un sistema de inferencia borroso basado en la clasificación, o sea, en el reconocimiento de determinado patrón.
- Generación de artefactos resultantes del análisis: como resultado final se actualizan los siguientes artefactos: Registro de Riesgos y Lista Priorizada de Riesgos.

2.1.1. Alcance

Este modelo es aplicable a proyectos de producción de software que deseen implementar una gestión de riesgos periódica con el objetivo de realizar un análisis cuantitativo de los riesgos ya identificados previamente, para determinar la probabilidad de ocurrencia y el impacto de los mismos, que permitirá disminuir las amenazas y aumentar las oportunidades.

El modelo ofrece descripción y guía objetiva para la realización de los procesos involucrados en el análisis cuantitativo de los riesgos, especificando sus entradas, salidas, participantes, descripción del flujo de trabajo y las actividades que se realizan durante su ejecución. Define métodos a aplicar para dicha realización, así como la implementación de los algoritmos que se propone usar con este propósito.

2.1.2. Principios

Los principios aquí propuestos tributan a un adecuado manejo de la gestión de riesgos en proyectos de software y por ende del análisis cuantitativo de los mismos. Estos son:

- **Personal involucrado:** Para que el proceso sea efectivo es necesario que el personal involucrado en este proceso se sienta comprometido con la validez de los resultados. Se propone que las personas involucradas sean:
 - Líder del proyecto.
 - Analista de Sistemas.
 - Arquitecto.
 - Ingeniero de Pruebas.

- Planificador
- Gestor de Recursos Humanos

En el modelo participan además expertos externos al proceso de desarrollo, que participan activamente en el proceso completo de análisis cuantitativo.

El **líder del proyecto** es el que mejor debe conocer cómo será la gestión de riesgos en el proyecto, pues es el que, anteriormente, debió elaborar el Plan de Gestión de Riesgos.

Debe ser además el responsable de garantizar el cumplimiento de los cronogramas y compromisos productivos; de supervisar y de guiar el proceso de desarrollo de software. Le corresponde organizar y controlar el trabajo de los miembros, y participar en la toma de decisiones estratégicas. Debe mantener el control y seguimiento periódico de los indicadores de su proyecto, y determinar medidas correctivas cuando sea necesario. Además debe ser el que gestione los recursos materiales necesarios para el proyecto y para el equipo de desarrollo. El **analista de sistemas** es el que debe dominar el negocio y fiel velador porque se cumplan los requisitos del cliente, a pesar de los riesgos que aparezcan en el proyecto. Debe participar en la definición del proyecto, y ser el responsable de traducir la comunicación entre clientes y desarrolladores.

El **arquitecto** es el que mejor debe conocer el desarrollo y qué consecuencias en él podría traer determinado riesgo. Debería ser el líder técnico y el que posea la autoridad para tomar decisiones de este tipo. Constantemente comunica el valor de lo que se está haciendo y entiende el proceso de desarrollo. Le corresponde ser el responsable de que las políticas y estándares, definidos en la arquitectura, se cumplan. Ayuda al líder del proyecto a estimar los costos, y ayuda a ubicar al personal según sus posibilidades.

El **ingeniero de pruebas** también debió participar en la creación del Plan de Gestión de Riesgos, y es el que vela por el ajuste al modelo durante su aplicación. Es un eslabón clave en este proceso porque es el encargado de velar por que se cumplan los parámetros y estándares de calidad establecidos, para que el producto sea aceptado finalmente por el cliente.

El **planificador** debe participar en el análisis cuantitativo porque es el responsable de planificar y controlar las tareas que se asignan, los planes de iteraciones, y de actualizar el cronograma, plan de proyecto y fechas de entregas. Es el que debe llevar el control de la ejecución del proyecto, medir la eficiencia del desarrollo, y controlar imprevistos y contratiempos.

El **gestor de recursos humanos** es el que mediante la emisión de su juicio, permitirá conocer datos o propiedades de los riesgos que no podrían conocerse sin haber tenido experiencia en situaciones similares. Es el especialista en información por lo que cumple la función de ingeniero de conocimientos, y es necesaria su presencia para iniciar el análisis cuantitativo de los riesgos.

Deben analizarse las probabilidades de ocurrencia y el impacto de los riesgos, y con la colaboración de todo el equipo deben tomarse las medidas necesarias para minimizar sus efectos negativos para el proyecto.

- **Enfoque de mejora continua:** el principal elemento que contribuye a la mejora es el aprendizaje. El modelo asume un aprendizaje sistemático, pues cada cierto tiempo establecido por el equipo, se realiza una retroalimentación de los posibles riesgos a los que se ve expuesto, la probabilidad de ocurrencia y el impacto de los mismos, actualizando de esta forma un conjunto de reglas de aprendizaje que agilizan el análisis.
- **Centrado en los clientes:** con su aplicación, el modelo se centra en obtener un beneficio visible para los clientes para lograr su mayor involucración en este proceso. A la vez, mantienen un equilibrio con la calidad, tiempo y costo para lograr un efecto similar en todas las partes involucradas en el proyecto.

2.1.3. Premisas

Las siguientes premisas son cruciales para una aplicación efectiva del modelo:

- **Personal capacitado en el análisis cuantitativo de riesgos:** es necesario que se asigne un equipo capacitado en el tema de la gestión de riesgos en sentido general, independientemente de que cubran alguna otra responsabilidad en el proyecto, dicho equipo es el encargado de todo el flujo, son los que identifican, y priorizan los riesgos, entradas fundamentales para el análisis cuantitativo, además de que tendrían además del resultado cuantitativo tendrían una visión general de los mismos.
- **Software requerido:** para la aplicación del modelo es necesario contar con el software indispensable.
- **Plan de gestión de riesgos:** denota la involucración activa de la dirección del proyecto con la gestión de riesgos, así como las tareas generadas durante el análisis cuantitativo en los planes y mecanismos de planificación y control del proyecto. Asegura una vinculación con el resto de los procesos organizativos, de control y de desarrollo del proyecto.

2.1.4. Construcción de la Base de Riesgos

La primera de las actividades del modelo es la construcción de la base de riesgos, para ello se necesitan consultar una serie de artefactos de entrada, tal como se describe en la Figura 5. Los involucrados en el análisis cuantitativo utilizan, para realizar las actividades, los artefactos que se muestran en la Figura 6: el plan de gestión del proyecto, el plan de gestión de riesgos del proyecto, el registro de riesgos y la lista priorizada de riesgos, creados en el proceso de análisis cualitativo anterior. El plan de gestión del proyecto facilita información acerca del plan de gestión del cronograma, plan de gestión de costes, entre otros. El plan de gestión de riesgos permite conocer los roles, responsables, planificación de las actividades a realizar, presupuesto, técnicas a usar, entre otros datos, que permiten conocer y ubicar a los involucrados en el contexto del proyecto, y poder llenar algunos valores de la base de riesgos. El registro de riesgos es el artefacto que mayor información provee para la base de riesgos, pues contiene, de cada riesgo: identificador, etapa, estado, prioridad, categoría, entrevistado, evento, consecuencia, causa, posible probabilidad, posible impacto, marco de tiempo, estrategia de respuesta, responsables, riesgos relacionados, observaciones, entre otros. La lista priorizada de riesgos contendrá los riesgos priorizados cualitativamente, siendo uno de los resultados del análisis cualitativo, y se actualizará con los riesgos priorizados cuantitativamente.

Posteriormente, con el desarrollo de las actividades de gestión de riesgos, algunos de estos valores pueden ser actualizados, tal es el caso del valor de la probabilidad e impacto del riesgo, que se llenan solo después del análisis cuantitativo.

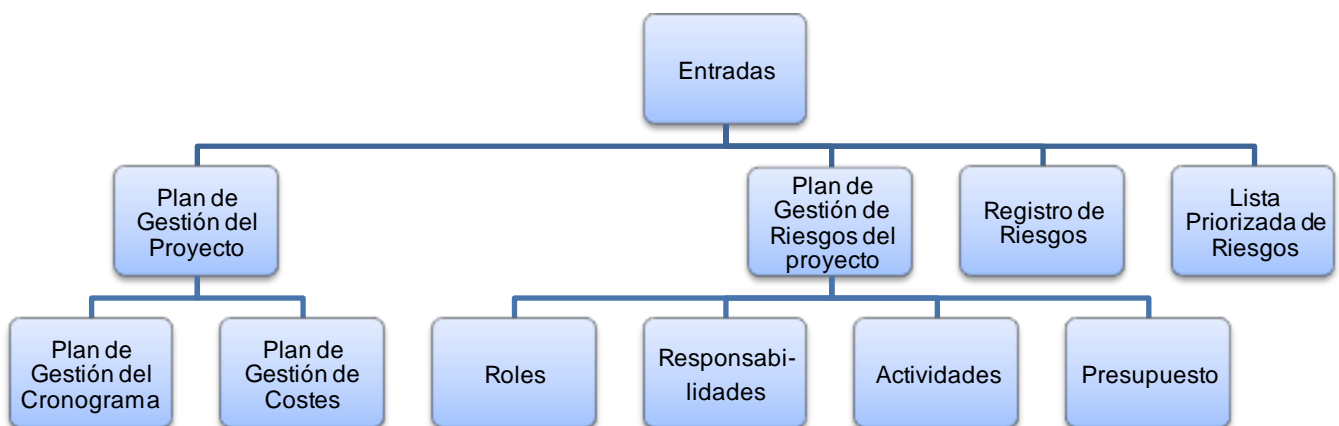


Figura 6 Entradas para la construcción de la base de riesgos

Para la representación de los riesgos se usó una propuesta entendible, usable, práctica y que contribuye a conocer más datos sobre el riesgo (García, 2008). Esta es una representación vectorial

del riesgo que incluye propiedades claves para su uso adecuado por todos los procesos de la gestión de riesgos, tal como se aprecia en la Figura 7. Es fácil de representar computacionalmente, apoya al análisis y a las búsquedas. Sus parámetros pueden sufrir variaciones de acuerdo al proyecto en cuestión y para una mejor gestión de los mismos se agrupan de acuerdo a la categoría a la que pertenecen.

Identificador	Etapas	Estado	Prioridad	Categoría	Entrevistado	Consecuencia	Causa		...
---------------	--------	--------	-----------	-----------	--------------	--------------	-------	--	-----

Figura 7 Representación de la base de riesgos

Identificador: identificador único del riesgo, debe ser único dentro de la universidad, debe ser una cadena de caracteres donde los dos primeros identifiquen la facultad, los tres siguientes el proyecto, y luego enteros consecutivos, por ejemplo: 03COE15. Los riesgos siempre presentes, o riesgos comunes deben tener los primeros indicadores y cuando un riesgo se decida incluir como riesgo siempre presente, debe cambiarse el indicador del proyecto por GLB, para indicar que es un riesgo global.

Etapas: etapa del proyecto donde se manifiesta el riesgo. (Inicio, Elaboración, Construcción, Cierre, Soporte).

Estado: uno de estos valores: activo o no activo.

Prioridad: orden de prioridad para atender este riesgo.

Categoría: identificador de la categoría, o sea, una de las definidas por el modelo. (Ver Anexo 1: Taxonomía del desarrollo de software).

Entrevistado: persona que identificó el riesgo.

Probabilidad: valor indicando la probabilidad de ocurrencia. Este valor es uno de los definidos en el Plan de Gestión de Riesgos.

Impacto: valor de impacto para cada objetivo de proyecto, de acuerdo a lo definido en Plan de Gestión de Riesgos.

Consecuencia: descripción textual de las consecuencias que trae la ocurrencia del evento. Debe expresar claramente el impacto sobre los objetivos del proyecto, no tiene por qué ser sobre todos los objetivos. En otra propiedad del riesgo se cuantificará y valorará este impacto. Representa la secuela o resultado de la materialización de un problema.

Por cada consecuencia encontrada, se describirá para cada riesgo qué tanto influye en la misma, así como el impacto que tiene en el proyecto en cuestión, en base al tiempo, el costo y el grado de insatisfacción del cliente.

Causa: descripción textual de la causa que origina el riesgo. Es importante que sea precisa, descriptiva y objetiva. Es escrito en presente. Una causa es una condición identificable que hace que algo suceda.

Por cada causa encontrada, se describirá para cada riesgo qué tanto influye en la misma, así como la probabilidad de ocurrencia de cada una.

Frecuencia: el factor de aparición del riesgo en el tiempo de proyecto. Expresa un valor porcentual respecto al tiempo total del proyecto indicando con qué frecuencia aparece este riesgo.

Certidumbre de la Información: valor que indica cuán seguro se está de la información obtenida sobre el riesgo.

Marco de tiempo: tiempo mínimo en el que deben ejecutarse los planes de contingencia o explotación para este riesgo. Debe expresarse en la unidad de medida acordada en el plan de gestión de riesgos.

Esta representación conforma la base de riesgos, que es usada por el primer módulo para el aprendizaje e identificación de tendencias y patrones, con riesgos comunes, recopilados de distintos proyectos de gestión de la UCI, así como de la bibliografía consultada. Cada riesgo en esta base posee los valores respectivos de los parámetros previamente definidos para este modelo, de modo que se pueda construir la base de casos, a partir de la información que la misma brinde. Esta también deberá actualizarse con los nuevos riesgos que vayan surgiendo, ya que el desarrollo de software es una rama en constante cambio, ya sea por la competencia o por las condiciones socio-económicas de la empresa donde se va a implantar el software desarrollado o en desarrollo.

Para una mejor comprensión de cómo se desarrolla el análisis, la generación de la reglas y posteriormente la inferencia, se abordarán brevemente los algoritmos en los que se basa el modelo propuesto.

2.1.5. Construcción de Base de Casos de Entrenamiento

En esta actividad se consulta la base de riesgos realizada en la actividad anterior, para dar como resultado una base de casos para entrenamiento. Para esta base de casos no se toman todos los parámetros de la base de riesgos, ya que algunos de estos parámetros iniciales no son relevantes, por ejemplo el entrevistado no es un dato relevante para calcular el impacto o la probabilidad de un riesgo, y por tanto no se toma este indicador para la construcción de la base de casos de entrenamiento. Lo mismo sucede en el caso de la probabilidad y el impacto iniciales en la base de riesgos.

Existen distintas variantes en que puede producirse la entrada al módulo de aprendizaje:

- El experto define el número de intervalos: esto implica el número de términos lingüísticos que va a tener la variable. Para obtener dichos intervalos el sistema implementa dos métodos para discretizar.
- El experto define cuáles son los intervalos: en esta variante no es necesario aplicar los métodos para discretizar, y puede pasarse directamente a la fuzziificación.
- El experto no sabe cuáles ni cuántos intervalos son, de ahí que sea necesario la clusterización. (Liliana, 2002)

La primera variante se corresponde con el fichero de entrada *.data*. El fichero *.data* es un fichero de texto que le permite al usuario especificar las características de los atributos y los casos. La estructura de estos ficheros solo permite que los atributos métricos sean discretizados, y los intervalos no pueden ser definidos por los expertos, sino que son calculados a través de los métodos de discretización que implementa el sistema. La segunda y tercera variante se corresponden con los ficheros *.lrn* y *.names*, que será la estructura que se utilizará en el modelo propuesto, por ende no se realizará el proceso de discretizar y se pasará directo a la fuzziificación.

El proceso de fuzziificación parte de los atributos discretizados, permitiendo obtener de cada atributo una variable lingüística, y de cada intervalo se construye una función de membresía para cada término lingüístico de dicha variable lingüística⁸, especificando el solapamiento deseado.

Un fichero *.lrn* es un fichero que contiene una base de casos y siempre va acompañado de un fichero *.names*. En el fichero *.names* se puede especificar si se quiere clusterizar en el caso de que el experto no sepa cuáles ni cuántos intervalos tiene un atributo, si el experto conoce cuáles o cuántos son los intervalos en que se divide un atributo puede especificarlo y no será necesario la clusterización.

2.1.6. Generación de Reglas Borrosas

En esta actividad se usa como entrada, la base de casos generada en la actividad anterior.

En el aprendizaje, se presenta el proceso de construcción de las reglas borrosas dividido en dos etapas fundamentales. La primera dedicada a la construcción de los conjuntos borrosos y las variables lingüísticas y la segunda a la generación de la base de reglas a partir de las variables lingüísticas construidas. Ambas etapas pueden ser desarrolladas en los algoritmos MLRUL y/o GenRul5, que permiten la generación de reglas borrosas (Piñero, 2005). Antes de poder usar cualquiera de los

⁸ Las variables lingüísticas constituyen el centro de las técnicas de modelación borrosa (Piñero, 2005). Encapsulan las propiedades de aproximación o conceptos imprecisos en una forma sistemática y conveniente computacionalmente. Estas reducen la aparente complejidad de describir un sistema por la correspondencia a una etiqueta semántica para el concepto fundamental. Aún una variable lingüística siempre representa un espacio borroso. En su estructura se encuentran: un conjunto borroso, y clasificadores.

algoritmos, primero se deben realizar los procesos de discretización o clusterización para luego realizar la fuzzyficación, también existe la opción de que el usuario especifique la cantidad de intervalos y sus extremos, y ya no tenga que realizarse la discretización o clusterización antes de la fuzzyficación.

El concepto “aprendizaje” puede ser formulado como un problema de búsqueda completa, de principio a fin, desde un espacio predefinido por un potencial de hipótesis hasta encontrar la hipótesis que más se adapte a los ejemplos de entrenamiento. Existen varias formas de representar el conocimiento, entre ellas las reglas constituyen una de las más usadas, pues su modo de representación es el del razonamiento lógico de los seres humanos. Con el surgimiento del aprendizaje automatizado la generación de reglas se realiza automáticamente mediante algoritmos, facilitando el trabajo de los ingenieros del conocimiento, y de especialistas en el desarrollo de sistemas expertos. (Liliana, 2002)

Esta actividad ofrece dos ficheros de salida. El primero es un fichero *.rul* que será usado para realizar el proceso de inferencia y el segundo es un fichero texto que describe las reglas borrosas, previamente generadas. El fichero de descripción solo puede ser usado para comprender las reglas borrosas generadas, mientras que el fichero *.rul* no es un fichero de explicación y puede ser usado para clasificar los casos desconocidos, como puede apreciarse en la Figura 8.

El algoritmo MLRUL se usa para la generación de reglas, pues construye particiones sucesivas por medio de reordenamientos del conjunto de entrenamiento hasta concluir con una partición⁹ donde queden agrupados convenientemente los casos con propiedades similares (Piñero, 2005).

La variable lingüística con mayor relevancia en la clasificación de una partición determinada, se selecciona como criterio de comparación para la creación de sus particiones durante el reordenamiento de los casos. La relevancia se determina por medio de una expresión matemática que propicia la homogeneidad entre casos de una misma clase y la heterogeneidad entre estratos correspondientes a clases diferentes. Para reordenar cada partición son usados diferentes métodos. Para escoger el atributo idóneo este algoritmo provee cuatro métodos:

- El primero es el método de Mántaras que calcula las distancias entre las particiones como medida de selección del atributo.
- Se propone la fórmula de Cramer como un segundo método, esta es una medida de la interrelación entre las variables.
- Un tercer método permite la selección del atributo mediante el uso de la fórmula de la entropía.

⁹ Se denomina partición a un determinado agrupamiento de casos

- Finalmente, el algoritmo ofrece otro método, MLRelevance, que es una nueva medida de selección del atributo. Este último brinda la heterogeneidad entre elementos que pertenecen a diferentes clases, y la homogeneidad entre los elementos que pertenecen a la misma clase.

El algoritmo termina cuando todas las particiones son particiones cerradas. Una partición es cerrada si y solo si todas las clases dentro de ellas tienen el mismo consecuente. Este algoritmo garantiza por su propia definición, un aprendizaje independiente del orden en que aparecen los casos del conjunto de entrenamiento.

El algoritmo MLRUL posibilita además la manipulación de rasgos irrelevantes, la ausencia de información y manejo de datos con información ruidosa.

A continuación se describen en la Figura 8 cada uno de los pasos del algoritmo MLRUL, basados en los resultados de la tesis doctoral “Un modelo para el aprendizaje y la clasificación automática basado en técnicas de softcomputing”.

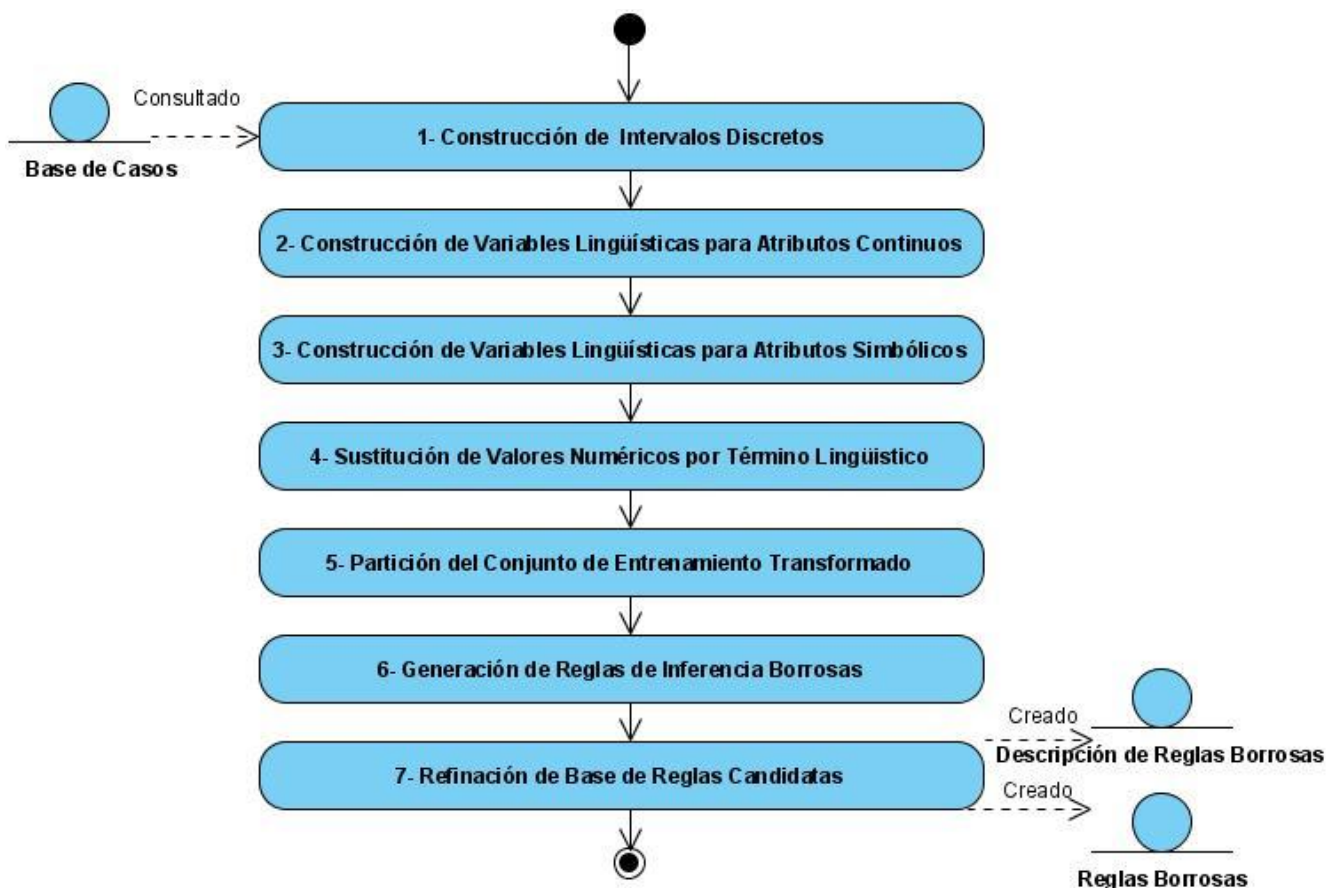


Figura 8 Creación de reglas borrosas. Algoritmo MLRUL

1- Construcción de Intervalos Discretos

Los intervalos discretos se construyen a partir de los atributos continuos.

Consiste en convertir los atributos continuos en discretos, o sea construir los intervalos discretos para cada atributo en los que se puedan agrupar todos los valores de un atributo continuo. Este posee tres variantes:

- Que los intervalos sean propuestos por los especialistas en la temática específica del problema en cuestión.
- Seguir un proceso sencillo de discretización de los atributos, tomando como estrategia: la construcción de intervalos de igual tamaño donde la complejidad de calcular la amplitud de todos los atributos es $O(n.r)$, o de intervalos con igual frecuencia cuya complejidad computacional es un $O(n)$.
- La tercera variante se basa en llevar a cabo un proceso de clusterización a nivel de atributo por medio del cual se construyan los intervalos.

a. Algoritmo secundario: Clúster de atributo continuo

- I. Dado el conjunto original de datos numéricos, ordenarlos.
- II. Calcular todas las distancias entre todos los datos numéricos tomados dos a dos. En este paso se obtiene el conjunto de todas las distancias.
- III. Calcular el valor de umbral que se utilizará como criterio de parada del agrupamiento. Se propone que se calcule como el rango del percentil en el espacio de las distancias.
- IV. Tomando el conjunto original de datos, crear clústeres tal que cada agrupamiento contenga inicialmente solo un valor. Inicialmente la cantidad de clústeres debe ser igual al conjunto original de datos.
- V. Unir los clúster adyacentes teniendo en cuenta que las distancias entre estos clústeres no supere el valor de umbral. Se toma como distancia entre clústeres, la distancia entre los elementos más lejanos.
- VI. Repetir el paso V hasta que ya no sea posible unir dos clústeres adyacentes.

2- Construcción de Variables Lingüísticas para Atributos Continuos.

En este paso se construyen las variables lingüísticas para cada uno de los atributos continuos. Para cada atributo continuo se construirá una variable lingüística formada por tantos conjuntos borrosos como intervalos discretos se hayan calculado para el atributo en el paso anterior. Para ello se

determinan los parámetros de las funciones de pertenencia. Luego se realizan los procesos de discretización o clusterización para después realizar la fuzzyficación.

a. Discretización y Clusterización

El objetivo principal de la **clusterización** es dividir los puntos que representan diferentes observaciones en clústeres, de tal modo que los puntos de un mismo clúster estén más cercanos entre ellos que los puntos que pertenecen a diferentes clústeres.

Cuando se somete un atributo a un proceso de clusterización, lo primero que se hace es estandarizar los valores de dichos atributos. Los atributos se estandarizan mediante las siguientes fórmulas:

$$z_i = \frac{x_i - \bar{x}}{v}$$

$$v = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}$$

Donde:

x_i es el valor a estandarizar.

\bar{x} es la media de los valores.

n es la cantidad de valores del atributo.

Después se realiza el proceso inverso de la estandarización para llevar los intervalos a los valores reales y construir las funciones de membresía.

El método de clusterización por el umbral, ordena los valores y compara la diferencia de los adyacentes con un determinado umbral, si no sobrepasa el umbral, se ubican ambos valores en la misma clase, en caso contrario se ubican en distintas clases.

Para el cálculo del umbral se trabaja con el percentil. Un percentil es un punto que divide a la distribución de frecuencias en dos partes de tal forma que, a su izquierda o por debajo de él, se encuentre un determinado por ciento del total de observaciones. En general un percentil se denotará por P_q , donde q representa qué porcentaje de las observaciones se encuentran por debajo de él. De esta forma por debajo de P_{20} se encuentra el 20% de las observaciones, por debajo de P_{75} se encuentra el 75% de las observaciones, etc.

La fórmula para el cálculo de los percentiles es la siguiente:

$$P_q = L + \frac{\left(\frac{q \cdot n}{100} - N\right)c}{n_p}$$

Donde:

q: por ciento de las observaciones que se quiere a la izquierda de P_q .

n: total de observaciones.

n_p : frecuencia de la clase donde cae el percentil.

N: frecuencia acumulada hasta la clase anterior a la clase donde se encuentra el percentil.

c: amplitud de la clase donde cae el percentil.

L: límite inferior de la clase donde cae el percentil.

Ya calculado el valor del percentil, el proceso de formación de clústeres permite, a partir de un ordenamiento de los valores, escoger entre tres criterios para comparar la diferencia de los adyacentes con el umbral (el valor del percentil), y de allí, si no sobrepasa el umbral se ubican ambos valores en la misma clase, en caso contrario se ubican en distintas clases. Siempre teniendo en cuenta que las distancias calculadas son el cuadrado de la diferencia de los elementos.

Estos criterios son:

- Buscar la menor distancia entre las medias de los elementos de los grupos que se están comparando.
- Buscar la menor distancia entre los elementos más lejanos de los grupos que se están comparando.
- Buscar la menor distancia entre los elementos más cercanos de los grupos que se están comparando.

La **discretización** es el proceso de transformar atributos de dominio continuo a dominios discretos. Dado un dominio definido como un intervalo $[a, b]$, discretizar el atributo significa producir una partición. A partir de allí los valores de los atributos son etiquetas que representan cada elemento de la partición. (Liliana, 2002)

Los métodos para discretizar son:

- Intervalos de igual tamaño.

Este método, como su nombre lo indica, discretiza teniendo en cuenta el tamaño, es decir, la amplitud de las particiones, de forma tal que esta sea la misma para todos los intervalos. Para lograr esto, el tamaño del intervalo responde a un cálculo dado por la fórmula siguiente:

$$\text{Amplitud} = \frac{\text{max} - \text{min}}{K}$$

Donde:

max y min: son los valores máximo y mínimo del intervalo inicial $[a, b]$,

K: es la cantidad de clases que se quieren formar con la discretización.

Culminado el proceso de discretización por este método se tienen K intervalos, todos de igual amplitud, sin importar cuántos elementos tiene cada uno de ellos.

- Igual frecuencia por intervalo.

Al discretizar por este método el objetivo es lograr que todas las particiones tengan la misma cantidad de elementos. Esto responde a la siguiente fórmula:

$$Frecuencia = \frac{N}{K}$$

Donde,

N: es la cantidad de elementos del intervalo inicial [a, b],

K: es la cantidad de clases que se quieren formar con la discretización.

Cuando se haya hecho el proceso de discretización se tendrán K intervalos, todos con la misma cantidad de elementos, es decir, la misma frecuencia, sin importar la amplitud de ellos.

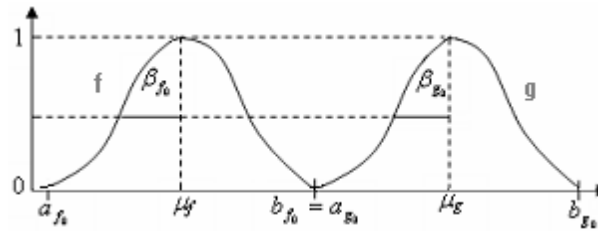
Al discretizar un atributo, se obtienen los intervalos en que éste fue dividido.

b. Fuzzificación

El proceso de fuzzificación parte de los atributos discretizados permitiendo obtener de cada atributo una variable lingüística, y de cada intervalo se construye una función de membresía para cada término lingüístico de dicha variable lingüística especificando el solapamiento deseado. Para ello es necesario obtener los valores de sus parámetros a partir de los atributos discretizados. El tipo de función de membresía a utilizar para los atributos, en este caso, son: funciones de membresía campanas Beta y funciones de membresía triangulares. (Liliana, 2002)

- **Determinación de las Funciones de Membresía Beta (Bell):**

Se parte de los puntos que son extremos de los intervalos del atributo discreto del cual se quiere obtener la variable lingüística. Para poder construir la primera función de pertenencia es necesario tener en cuenta el primer intervalo y el segundo. Para construir el resto de las funciones de pertenencia sólo es necesario considerar la función adyacente que la precede. En la siguiente figura se muestra un ejemplo de función bell, también conocida como campana.



a_{f_0} b_{f_0} : Extremos del primer intervalo del atributo discreto.

a_{g_0} b_{g_0} : Extremos del segundo intervalo del atributo discreto. $a_{g_0} = b_{f_0}$

β_{f_0} β_{g_0} : Valores Beta de las funciones f y g respectivamente.

Esta función se utiliza en los casos de variables numéricas¹⁰.

3- Construcción de Variables Lingüísticas para Atributos Simbólicos

En este paso se construyen las variables lingüísticas para cada uno de los atributos simbólicos. Para ello se determinan los parámetros de las funciones de pertenencia. Luego se realizan los procesos de discretización o clusterización para después realizar la fuzzyficación, procesos explicados anteriormente. Se construyen las funciones de pertenencia correspondientes a cada uno de los atributos simbólicos. De forma similar a como ocurre en el caso numérico cada atributo constituirá una variable lingüística, y cada valor del atributo estará representado por un conjunto borroso y su respectiva función de pertenencia.

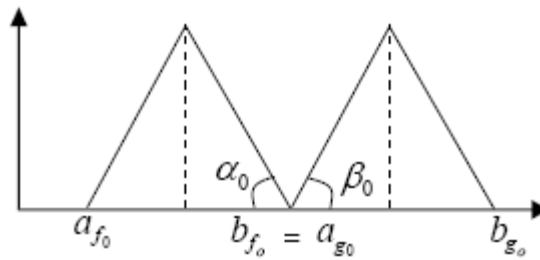
- **Determinación de las Funciones de Membresía Triángulo:**

El proceso constará de dos partes: la construcción de las funciones de pertenencia iniciales y la transformación hasta obtener el efecto deseado en la transformación de las curvas. Durante la construcción de las funciones de pertenencia finales se debe verificar la relación de proporcionalidad

$$\frac{b_{f_0} - \mu_f}{b_{g_0} - \mu_g} = \frac{b_f - \mu_f}{b_g - \mu_g}$$

mediante la ecuación: $\frac{b_{f_0} - \mu_f}{b_{g_0} - \mu_g} = \frac{b_f - \mu_f}{b_g - \mu_g}$, para evitar que se desproporcionen. En la siguiente figura se muestra un ejemplo de función de tipo triángulo.

¹⁰ Se refiere a las que tengan números como valores. Por ejemplo el marco de tiempo, que debe ser un número



Esta función se utiliza en el caso de variables simbólicas¹¹.

4- Sustitución de Valores Numéricos por Término Lingüístico

En este proceso se sustituyen, en todos los casos, los valores numéricos por el término lingüístico del conjunto borroso donde alcanza mayor grado de pertenencia, y se tratan a los casos que tengan valores ausentes.

En este paso cada valor numérico de los casos del conjunto de entrenamiento es sustituido por el término lingüístico más apropiado de acuerdo al principio de máxima pertenencia. En el cual cada una de las alternativas es etiquetada con el término lingüístico en el que presente mayor valor de membresía.

En este paso del algoritmo también se tratan los casos que tengan valores ausentes. Se propone que para cada tupla con valores ausentes se consideren una de las dos estrategias siguientes:

- Ignorar las tuplas con valores ausentes que se sugieren hacer cuando lo que falta es el nombre de la clase o cuando el número de atributos con valores ausentes de la tupla es relativamente grande, por ejemplo: que falte el 50 % de los atributos predictores, y teniendo en cuenta además que la ausencia de la tupla en cuestión no afectará significativamente al conjunto de entrenamiento ni a los resultados finales.
- En el caso de que no se decida eliminar la tupla, se sustituye el valor ausente por un nuevo término lingüístico llamado *none*. Este nuevo término se manipula en lo adelante como un término normal, solo que al final en la formación de las reglas no se considera en la construcción de los antecedente. Esta estrategia evita la posible introducción de errores que se produce por otros enfoques donde se sustituye el término por el valor de la media o algún otro estadígrafo¹².

¹¹ Se refiere a las variables que tienen como valor una cadena de caracteres, por ejemplo: la prioridad, que puede ser alta, media o baja

¹² Es la medida que en Estadística se aplica sobre una muestra. En general se utilizan dos tipos: Los de Tendencia Central y los de Dispersión. Entre los primeros tenemos: a) las medidas denominadas promedios, o sea aquellas que tratan de localizarse hacia el centro de la serie; moda, media y mediana; y b) los cuartiles y deciles, o cuartas y décimas partes de las

5- Partición del Conjunto de Entrenamiento Transformado

Este paso es el más importante del algoritmo MLRUL, como resultado de este paso se obtiene una partición del conjunto de entrenamiento a partir de la aplicación del algoritmo secundario: Construcción de Particiones. Además se propone un criterio alternativo para la selección de rasgos: el criterio MLRelevance, en el algoritmo MLRUL.

a. Algoritmo secundario: Construcción de particiones

- I. Crear la colección de conjuntos abiertos, inicialmente esta tiene un solo elemento, el conjunto de entrenamiento obtenido en el paso 4 del algoritmo MLRUL. Complejidad n .
- II. Crear la colección de conjuntos cerrados inicialmente vacía.
- III. Si la colección de conjuntos abiertos está vacía, terminar, en caso contrario seleccionar un conjunto abierto cualquiera de la colección.
- IV. Seleccionar sobre el conjunto abierto seleccionado, el atributo más relevante; o sea el atributo que discrimina mejor los objetos que pertenecen al atributo más relevante.
- V. Particionar el conjunto abierto seleccionado, tomando como criterio, para la separación de los objetos en subconjuntos, la igualdad respecto al atributo más relevante seleccionado en el paso anterior. En este paso se ordenan los casos del conjunto abierto seleccionado, teniendo en cuenta los valores del atributo más relevante.
- VI. Para cada uno de los nuevos subconjuntos construidos en el paso anterior a partir del conjunto abierto seleccionado, chequear si es abierto o cerrado, y añadirlo a la colección de conjuntos abiertos o cerrados, según corresponda.
- VII. Ir al paso II.

6- Generación de Reglas de Inferencia Borrosas

En este paso se generan las reglas de inferencia borrosas a partir de la partición construida, o sea, a partir de la partición creada del conjunto de entrenamiento formada por la colección de conjuntos cerrados de objetos. De cada conjunto cerrado se conocen los objetos presentes y los atributos que participaron en la conformación de dicho conjunto. En este momento cada objeto es un caso donde los valores de sus atributos corresponden con términos lingüísticos de los conjuntos borrosos construidos previamente, este resultado se obtuvo a partir de la aplicación del principio de máxima membresía en

observaciones; estos sólo se aplican en los datos agrupados. Entre los de Dispersión están: la desviación media, la desviación mediana, la varianza, la desviación típica o estándar, la dispersión absoluta y relativa.

el paso 4. Para cada uno de los conjuntos cerrados de la colección, se toma un caso cualquiera de objetos y se genera una regla borrosa a partir de él. Se parte del algoritmo: Generación de una regla borrosa a partir de un caso y atributos que se explica a continuación.

a. Algoritmo secundario: Generación de una regla borrosa a partir de un caso y atributos.

- I. A partir de cada atributo del caso, se construye un antecedente de la regla que contenga el conjunto borroso cuyo término lingüístico se corresponda con el valor de este atributo. En caso de que el valor del atributo sea el desconocido, ignorar este atributo en la generación de la regla.
- II. Formar una regla por la combinación de los antecedentes construidos, tomando como consecuente de la regla, la clase del caso.

7- Refinación de Base de Reglas Candidatas.

Para refinar la base de reglas candidatas se propone usar el algoritmo secundario: Refinación de base de reglas candidatas. Como entrada a este algoritmo se tiene una base de reglas candidatas y el conjunto de casos de entrenamiento, como salida este algoritmo provee una base de reglas final que es un subconjunto de la base de reglas candidatas. Cada regla candidata tiene asociado un conjunto de números enteros que indican los índices de los casos del conjunto de entrenamiento que son cubiertos por dicha regla. Inicialmente todos los conjuntos asociados a las reglas candidatas son vacíos.

a. Algoritmo secundario: Refinación de base de reglas candidatas

- I. Para cada regla de la base de reglas candidatas y para cada caso del conjunto de entrenamiento: si la regla cubre al caso en un grado mayor que 0,90), adicionar el caso al conjunto de casos cubiertos por la regla, indicando que el caso es cubierto por la regla.
- II. Inicializar los conjuntos de casos cubiertos, la base de reglas final y el conjunto total de los casos.

2.1.7. Construcción de Sistema de Inferencia Borroso

En su forma más simple, el problema de clasificación se presenta como un proceso de reconocimiento de determinado patrón. Consiste en que se conoce que un universo de objetos, se agrupan en un número dado de clases, de las cuales se tiene, de cada una, una muestra (una parte) de entes, que pertenecen a ella, y el dilema consiste en dado un nuevo objeto poder establecer sus relaciones con

cada una de dichas clases. Se propone, entonces, la creación un sistema de inferencia borroso basado en la clasificación.

Existen tres tipos fundamentales de sistemas de inferencia borrosos: el modelo de Mamdani¹³, Sugeno y Tsukamoto¹⁴ (Piñero, 2005). Estos se diferencian en la forma del consecuente de sus reglas borrosas. Para la aplicación en problemas de clasificación, que es el caso que aquí se explica, con frecuencia se utiliza el Sugeno, precisamente por la estructura de las reglas que este tipo de sistema representa. Este modelo también conocido como TSK, por sus autores Takagi, Sugeno y Kang, utiliza reglas de la forma: **Si** “ x es A ” **y** “ y es B ” **entonces** $z=f(x, y)$. Donde A y B son conjuntos borrosos en el antecedente y z usualmente $f(x, y)$ es un polinomio cuyo grado determina el grado del modelo borroso, y cuyo resultado es un valor numérico, lo que lo convierte en el más aplicable al presente caso, donde se trabaja, precisamente, el análisis desde el punto de vista cuantitativo.

El funcionamiento de un sistema de inferencia borroso se basa en primer lugar en la existencia de un fichero de entrada (.rul) que contiene toda la información necesaria para determinar el sistema de reglas borrosas: descripción del dominio de los atributos, sus relevancias, información de las funciones de membresía asociadas a dichos atributos, valores de sus parámetros, descripción de cada una de las reglas del sistema incluyendo además sus niveles de significación, etc. En segundo lugar, se basa en un mecanismo de conexión existente entre los componentes que posibilita la creación de una red formada por capas, cuya topología se corresponde en alto grado con la estructura del sistema borroso especificado en el fichero de entrada. Cada componente que se utiliza constituye un nodo en la red y una capa de la misma está conformada por nodos que realizan la misma función. El flujo de información a través de esta red, da lugar a que se lleven a cabo los distintos procedimientos de cálculo, que tienen lugar en el proceso de inferencia borroso, dando como resultado la evaluación y la clasificación, como puede observarse en la Figura 9. La clasificación es la que devuelve como resultado el impacto y la probabilidad de los riesgos.

¹³ Mamdani: sistema de inferencia borroso en el cual los consecuentes son considerados variables lingüísticas.

¹⁴ Tsukamoto: sistema de inferencia borroso en el cual la función es considerada en el caso creciente o decreciente.

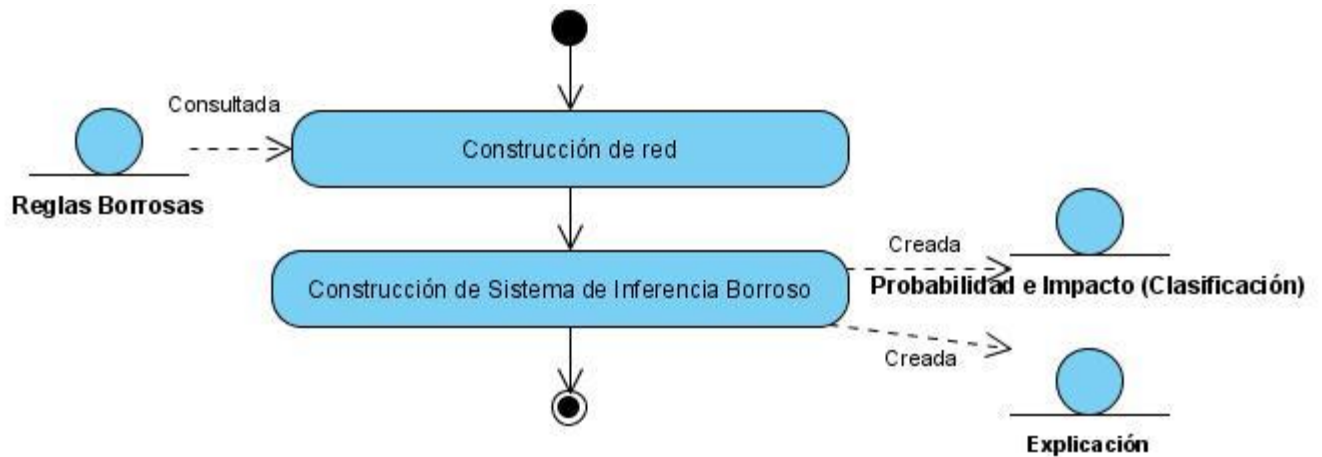


Figura 9 Construcción de Sistema de Inferencia Borroso

Se propone que se construya el sistema de inferencia borroso según la representación orientada a nodos, como se muestra en la Figura 10.

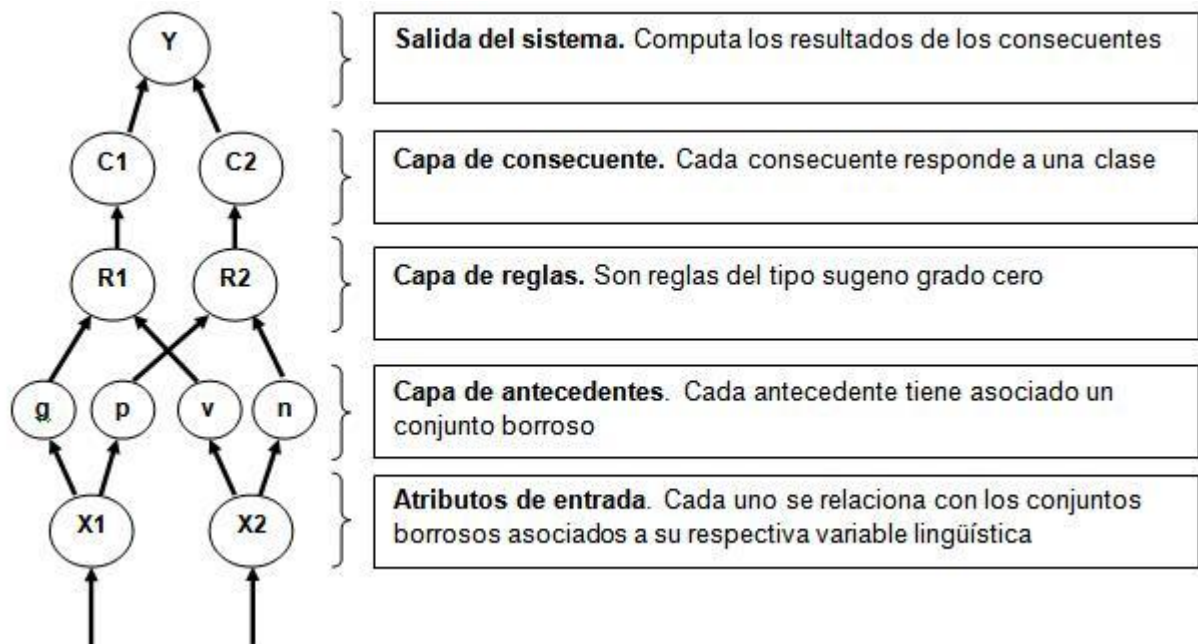


Figura 10 Sistema borroso. Topología de 5 capas, orientada a nodos

La primera capa corresponde a la capa de las variables de entrada o atributos, cada nodo en esta capa se conecta a los conjuntos borrosos que forman a la variable lingüística asociada al atributo.

La segunda capa es la capa de los términos lingüísticos. En este modelo las funciones de pertenencia de esta capa pueden ser funciones triángulo, beta o simbólicas.

La tercera capa se corresponde con la capa de reglas borrosas. El resultado de cada regla se computa a partir de la evaluación de los conjuntos borrosos, contenidos en sus antecedentes.

La cuarta capa son constantes que representan las clases del sistema de inferencia borroso. Los nodos de la capa de consecuentes, computan las respuestas de las reglas relacionadas con ellos, combinando sus respuestas. Dos criterios son propuestos para computar las salidas de las reglas:

- Criterio del máximo: la respuesta del consecuente será la respuesta de la regla borrosa asociada a él que devuelve el mayor valor.
- Calcular la media y la varianza de las respuestas de las reglas asociadas al consecuente.

La última capa del modelo se encarga de conformar la respuesta final del sistema a partir del análisis de los nodos consecuentes presentes en la capa anterior. Se puede señalar esta como la capa donde los consecuentes compiten. El criterio de selección de la respuesta final se hace teniendo en cuenta una de las dos estrategias que se muestran a continuación:

- Criterio del máximo consecuente: el consecuente que indica el mayor grado de pertenencia es devuelto como respuesta del sistema. Este criterio se corresponde con la primera variante (máximo) propuesta como criterio de evaluación del consecuente.
- Criterio del máximo aplicado a los estadígrafos del consecuente: a partir de que de cada consecuente se conoce la media y la varianza de las respuestas de sus reglas, el sistema da como respuesta la clase del consecuente que maximiza la media y minimiza la varianza.

2.1.8. Generación de Artefactos Resultantes del Análisis

Al aplicarse el modelo, deben generarse una serie de artefactos o resultados. Ellos son, en esencia, como se muestra en la Figura 11, una actualización del Registro de Riesgos y la Lista Priorizada de Riesgos.

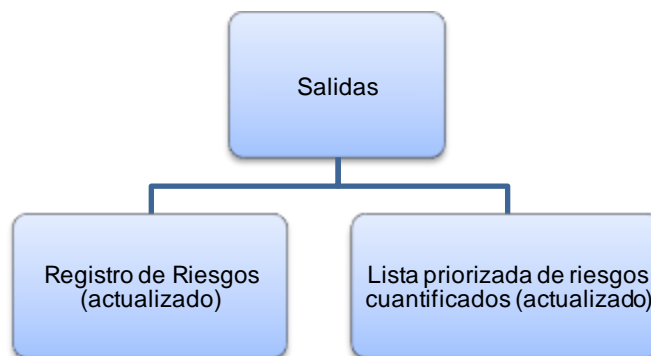


Figura 11 Artefactos de Salida

1. El Registro de Riesgos

Se realizan estimaciones de los posibles resultados del cronograma y los costes del proyecto, listando las fechas de conclusión y costes posibles con sus niveles de confianza asociados. Esta salida, normalmente expresada como una distribución acumulativa, se usa con las tolerancias al riesgo de los interesados para permitir la cuantificación de las reservas para contingencias de coste y tiempo. Un evento incierto que posiblemente puede ocurrir (o sea, con alta probabilidad), pero que tendría poco o ningún efecto sobre los objetivos (un bajo impacto), sería poco significativo. Igualmente un riesgo podría resultar despreciable si su probabilidad de ocurrencia fuese baja, aún cuando su impacto, de cierta importancia, fuese teóricamente posible. **Probabilidad de lograr los objetivos de coste y tiempo:** Con los riesgos que afronta el proyecto, la probabilidad de lograr los objetivos del proyecto, bajo el plan en curso, puede estimarse usando los resultados del análisis cuantitativo de riesgos. Luego de la obtención de la probabilidad y el impacto, se actualiza este artefacto, que debe ser creado en el análisis cualitativo.

2. Lista Priorizada de Riesgos

Durante el transcurso del análisis del riesgo se obtienen los datos correspondientes a cada riesgo de la probabilidad y el impacto, que sirven para ordenar los riesgos por importancia en el proyecto. Es obvio que no todos los riesgos tienen la misma incidencia en el proyecto y es lógico ordenarlos según algún criterio que indique cuáles son los más importantes, ya que puede ocurrir que ante un número alto de riesgos identificados, y debido a la falta de medios, solo se puedan gestionar una parte de ellos. Es por ello que se realiza una priorización de los riesgos reordenándolos en una Matriz de Probabilidad/Impacto, clasificándolos en Priorizados, No Priorizados o Medianamente Priorizados, teniendo en cuenta, en esta ocasión, sus valores numéricos de probabilidad e impacto. (Ver Anexo 7: Lista Priorizada de Riesgos Cuantificados)

Esta lista de riesgos permite conocer además si el riesgo requiere mayor contingencia de costes y/o aquellos que tienen más probabilidad de influir sobre el camino crítico del proyecto.

2.3. Conclusiones Parciales

Con el desarrollo del modelo se lograron establecer los pasos necesarios para realizar el análisis cuantitativo de los riesgos, haciendo uso de técnicas de softcomputing, específicamente de la lógica fuzzy. Particularmente se especificó el alcance del mismo, los principios y premisas para su aplicación y los elementos que garantizan el funcionamiento del modelo, así como una representación del mismo y de sus actividades:

- En la construcción de la base de riesgos, se establecieron los parámetros a medir para un mejor procesamiento de los mismos.
- Durante la construcción de la base de casos de entrenamiento, se estableció cómo deben quedar los parámetros para ser almacenados en los ficheros de entrada al modelo.
- Para la generación de reglas borrosas, se definió el algoritmo a utilizar, y se explicó cada uno de los pasos del mismo, sus entradas y salidas.
- En la construcción de sistema de inferencia borroso, se definió cómo se llevaría a cabo la misma, su estructura, así como sus entradas y salidas.
- Durante la generación de artefactos resultantes del análisis, se valoraron los artefactos existentes, a partir de lo cual se tomaron los que más se adecuan al modelo y se ajustaron al análisis cuantitativo de los riesgos.

Capítulo 3: Validación del Modelo

3.1. Introducción

En este trabajo se propuso utilizar para el análisis cuantitativo la implementación de algoritmos basados en una aleación entre modelos basados en la experiencia y modelos matemáticos; que como se mencionó anteriormente, en este caso, son las técnicas de softcomputing, y de ellas especialmente los sistemas de inferencia borrosos.

3.2. Implantación del modelo

Se realizó una implementación de la primera parte, o primer módulo del modelo propuesto, específicamente de las actividades: construcción de la base de riesgos, la construcción de la base de casos de entrenamiento, y la generación de reglas borrosas. A este módulo se le llamó: Módulo de Aprendizaje o “Learning”. Está prevista la implantación de este modelo en el Centro de Tecnologías de Almacenamiento y Análisis Inteligente de Datos (CENTALAD).

3.2.1. Construcción de la Base de Riesgos

Tal como se explicó en el epígrafe 2.1.4, para construir la base de riesgos, ([Ver Base de Riesgos](#)), fue necesario consultar el Plan de Gestión del Proyecto, el Plan de Gestión de Riesgos y el Registro de Riesgos, este último, generado en el proceso de análisis cualitativo de riesgos. Esto fue preciso, pues de este modo es que se obtuvieron los datos necesarios para llenar los valores de los parámetros definidos para la representación y recopilación de datos de riesgos, explicados en el epígrafe 2.1.4. En el siguiente ejemplo solo se muestran una parte de los valores de estos parámetros, para algunos ejemplos de riesgos. Estos riesgos fueron tomados principalmente de los proyectos ERP- Cuba y ONE (Oficina Nacional de Estadística.) Fueron validados, en el caso del ERP-Cuba por su jefe de producción Ing. Henry Cruz Mulet, y en el caso del proyecto PATDSI por su subdirector de producción Henrik Pestano Pino. (Ver Anexo 9: Declaración de Revisión).

Tabla 7 Ejemplo de Base de Riesgos

Id	Riesgo	Etapas	Estado	Prioridad	Categoría	Entrevistado	Consecuencia
IPERP1	Desconocimiento de las Tecnologías	Elaboración	Activado	Alta	Ingeniería	ERP	Decisiones incorrectas
IPERP2	Tiempo Limitado	Construcción	Activado	Media	Restricción	ERP	Incumplimientos del

					es		Plan
IPERP3	Atraso en el plan de capacitación	Inicio	No Activado	Media	Ingeniería	ERP	Incumplimientos del Plan
IPERP4	Poco conocimiento del negocio	Inicio	No Activado	Alta	Ingeniería	ERP	Decisiones incorrectas
IPERP5	Falta de comprensión de la base de datos	Elaboración	Activado	Alta	Ingeniería	ERP	Incumplimientos del plan
IPERP6	Cambios en la base de datos	Construcción	Activado	Alta	Ambiente de Desarrollo	ERP	Incumplimientos del plan
IPERP7	Falta de servidores para desarrollo y pruebas	Construcción	No Activado	Baja	Restricciones	ERP	Incumplimientos del plan
IPERP8	Inexperiencia organizativa del equipo	Inicio	Activado	Alta	Restricciones	ERP	Decisiones incorrectas
IPERP9	Poco conocimiento sobre arquitectura de sistema desde el punto de vista arquitectónico, no de análisis	Elaboración	Activado	Media	Ingeniería	ERP	Decisiones incorrectas
IPERP10	Dependencia de especialistas funcionales que no concuerdan a la hora de brindar información sobre los procesos de negocio.	Inicio	No Activado	Media	Ingeniería	ERP	Decisiones incorrectas
IPERP11	Falta de personal preparado para consultoría.	Transición	No Activado	Media	Ingeniería	ERP	Decisiones incorrectas
IPERP12	Distintas ubicaciones del personal	Elaboración	Activado	Media	Ambiente de Desarrollo	ERP	Atrasos
IPERP13	Carga extra del personal.	Construcción	Activado	Alta	Restricciones	ERP	Atrasos
IPERP14	Poca madurez del framework	Construcción	No Activado	Media	Restricciones	ERP	Atrasos
IPERP15	Escasa tecnología para realizar las pruebas	Transición	No Activado	Media	Restricciones	ERP	Atrasos

3.2.2. Construcción de Base de Casos de Entrenamiento

Los casos de entrenamiento fueron elaborados de acuerdo a parámetros previamente establecidos. A partir de los valores numéricos de los mismos se construyó la base de reglas que permitió obtener el resultado final. Como se explicó en el epígrafe 2.1.5, para llevarse a cabo fue necesario consultar la base de riesgos creada en el proceso anterior.

Para la construcción de la base de casos se tuvo en cuenta, la tercera de las opciones vistas en el epígrafe dedicado a la construcción de la base de casos en el capítulo anterior: que el experto definió cuáles son los intervalos, por lo que no fue necesario aplicar los métodos para clusterizar, y pudo pasarse directamente a la fuzzyficación, después de la discretización. Ver epígrafes 2-a, y 2-b

Para llenar cada uno de estos valores en la base de riesgos se consultó a especialistas de diferentes proyectos donde estuvieron presentes dichos riesgos.

La estructura en que se guardaron los parámetros en la base de casos es la que se muestra en la Figura 12. Una variable lingüística tiene conjuntos borrosos y estos tienen función de membresía y término lingüístico.

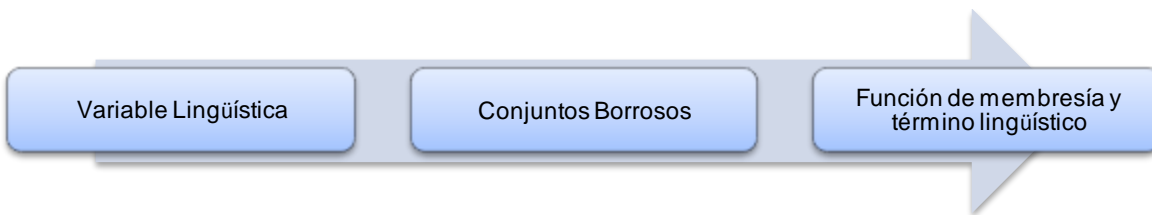


Figura 12 Estructura de la base de casos

Como resultado del proceso de construcción de la base de casos, se obtuvieron dos ficheros que la conforman: el fichero que contiene la estructura de los parámetros de cada una de las variables lingüísticas: (.names), ver Figura 13; y el fichero que posee los valores de esos parámetros (.lfn), ver Figura 14.

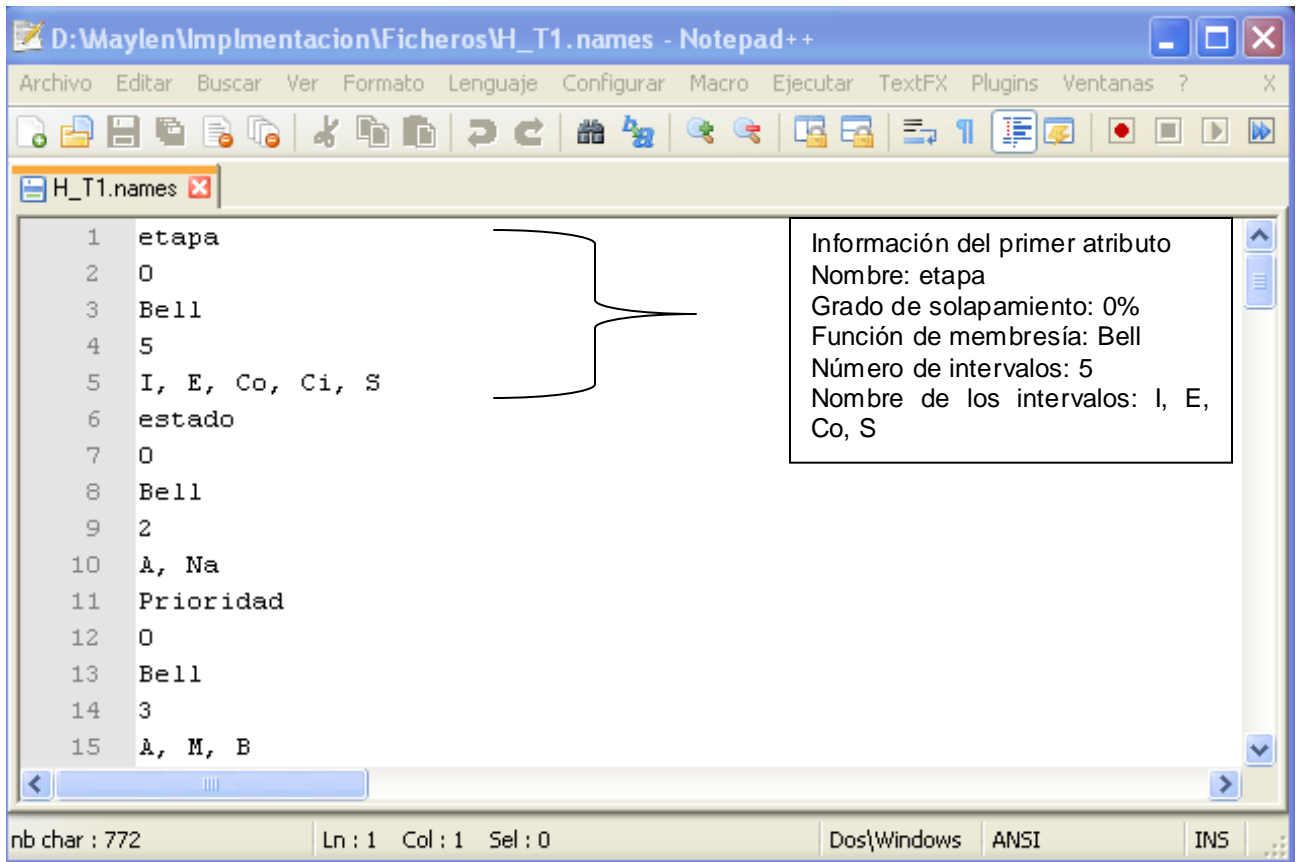


Figura 13 Base de Casos, estructura de los parámetros

Como se había explicado anteriormente el fichero .names siempre va acompañado del fichero .lrn. Este último es el que contiene los valores de cada variable lingüística para cada riesgo, de la base de casos, como puede observarse en la Figura 14. Para una mayor calidad en el resultado final, se hizo una selección de los riesgos de mayor certidumbre en la información de la base de riesgos para conformar la base de casos de entrenamiento.

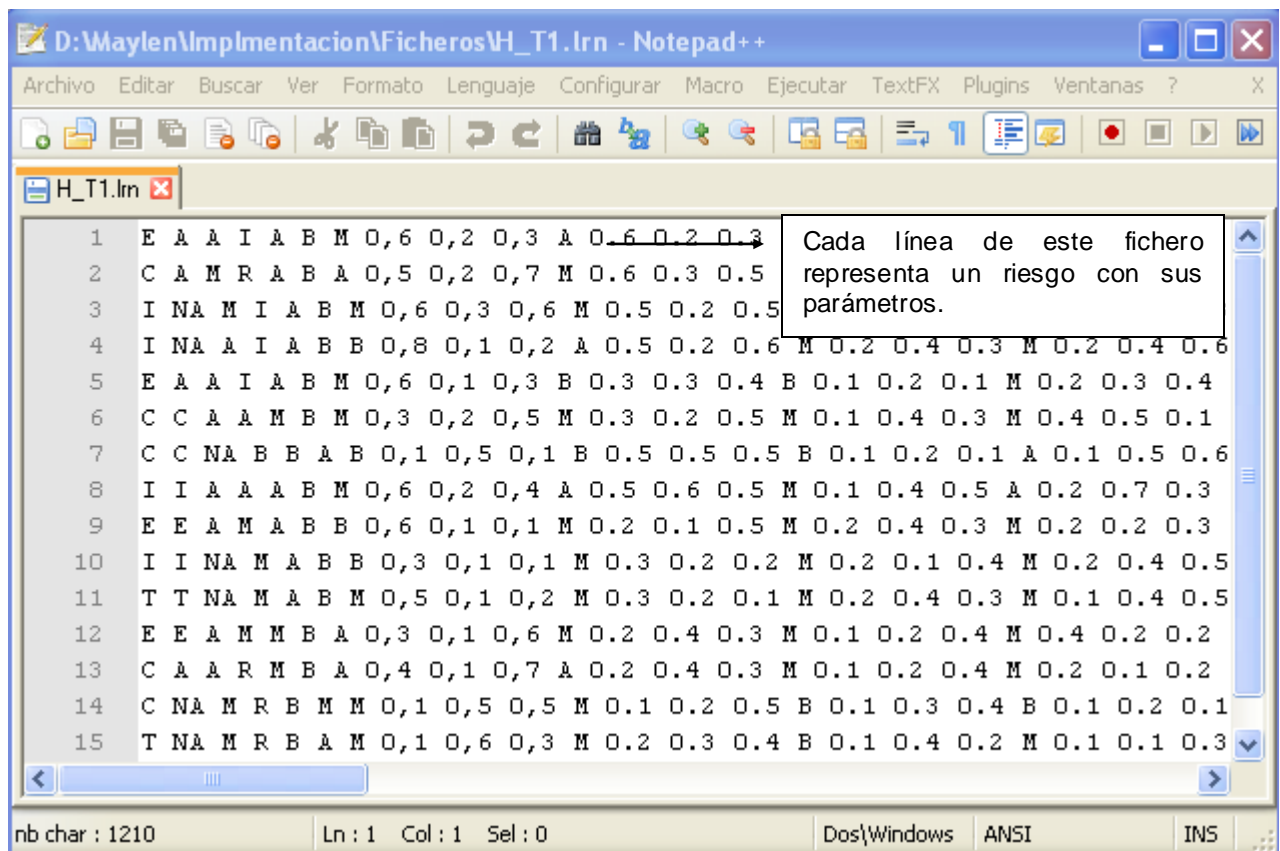


Figura 14 Base de Casos, valores de los parámetros.

A continuación se explican las variables lingüísticas por parámetros del fichero .names.

Las variables obtenidas deben contar con una serie de parámetros tales como:

Nombre: nombre de la variable lingüística.

Grado de solapamiento: va a estar dado por la dilatación de las funciones de pertenecía iniciales de forma proporcional para evitar que se deformen los valores iniciales y que se destruyan las potencialidades de la clusterización.

Función de membresía: una para cada término lingüístico de la variable lingüística.

Número de intervalos: resultados de la clusterización. Van a estar dados por la cantidad de posibles valores que pueda tomar la variable lingüística.

Nombre de los intervalos: resultado de la discretización. Van a estar definidos de acuerdo a los términos lingüísticos que pueda tomar la variable en cuestión.

Parámetros de la base de entrenamiento:

Los valores de los intervalos de las variables: consecuencia y causa, está dada por una muestra de 15 riesgos de la base de casos de entrenamiento. A continuación se muestran los parámetros para cada una de las variables definidas, teniendo en cuenta la muestra.

Etapa: etapa del proyecto donde se manifiesta el riesgo.

Nombre: etapa.

Grado de solapamiento: 0.

Función de membresía: Bell (campana).

Número de intervalos: 5.

Nombre de los intervalos: I, E, Co, Ci, S (Inicio, Elaboración, Construcción, Cierre, Soporte).

Estado: uno de estos valores: Activado o No Activado.

Nombre: estado.

Grado de solapamiento: 0.

Función de membresía: Bell (campana).

Número de intervalos: 2.

Nombre de los intervalos: A, NA (Activado, No Activado).

Prioridad: orden de prioridad para atender este riesgo.

Nombre: Prioridad.

Grado de solapamiento: 0.

Función de membresía: Bell (campana).

Número de intervalos: 3.

Nombre de los intervalos: A, M, B (Alta, Media, Baja).

Categoría: identificador de la categoría, o sea, una de las definidas por el modelo.

Nombre: categoría.

Grado de solapamiento: 0.

Función de membresía: Bell (campana).

Número de intervalos: 0.

Nombre de los intervalos: I, A, R (Ingeniería, Ambiente de Desarrollo, Restricciones) (Ver Anexo 1: Taxonomía del desarrollo de software)

Causa: descripción textual de la causa que origina el riesgo. Es importante que sea precisa, descriptiva y objetiva. Es escrito en presente. Después del análisis de las posibles causas más comunes presentes en la base de riesgos se decidió que se procesarían para estar presentes en la base de casos las siguientes: Falta de Personal Preparado (FPP), Falta de Recursos (FR), y Mala Planeación (MP). La probabilidad de un riesgo va a estar determinada por la sumatoria de la probabilidad de ocurrencia de las causas. Luego este resultado es dividido entre la cantidad de riesgos.

Falta de Personal Preparado (FPP): causa que va a estar dada por el por ciento de personal preparado necesario que no se encuentra disponible.

Nombre: FPP.

Grado de solapamiento: 0.

Función de membresía: Bell (campana).

Número de intervalos: 3.

Nombre de los intervalos: A, M, B.

Falta de Recursos (FRM): causa que va a estar dada por el por ciento de recursos necesarios con los que no cuenta el proyecto. Análisis que se realizará en dinero.

Nombre: FRM (dinero).

Grado de solapamiento: 0.

Función de membresía: Bell (campana).

Número de intervalos: 3.

Nombre de los intervalos: A, M, B.

Mala Planeación (MP): causa que va a estar dada por el por ciento de atrasos que provoque una mala planeación.

Nombre: MP.

Grado de solapamiento: 0.

Función de membresía: Bell (campana).

Número de intervalos: 3.

Nombre de los intervalos: A, M, B.

Ocurrencia de Falta de Personal Preparado (OFPP): probabilidad de ocurrencia de la causa, va a estar dado por un valor numérico entre 0 y 1.

Nombre: OFPP.

Grado de solapamiento: 0.

Función de membresía: Symbolic.

Número de intervalos: 0.

Nombre de los intervalos: none.

Ocurrencia de Falta de Recursos (OFRM): probabilidad de ocurrencia de la causa, va a estar dado por un valor numérico entre 0 y 1.

Nombre: OFRM.

Grado de solapamiento: 0.

Función de membresía: Symbolic.

Número de intervalos: 0.

Nombre de los intervalos: none.

Ocurrencia de Mala Planeación (OMP): probabilidad de ocurrencia de la causa, va a estar dado por un valor numérico entre 0 y 1.

Nombre: OMP.

Grado de solapamiento: 0.

Función de membresía: Symbolic.

Número de intervalos: 0.

Nombre de los intervalos: none.

Consecuencia: descripción textual de las consecuencias que trae la ocurrencia del evento. Debe expresar claramente el impacto sobre los objetivos del proyecto. No tiene por qué ser sobre todos los objetivos, en otra propiedad del riesgo se cuantificará y valorará este impacto. Después del análisis de las posibles consecuencias más comunes presentes en la base de riesgos se decidió que se procesarían, para estar presentes en la base de casos, las siguientes: Decisiones incorrectas (DI), Incumplimiento del Plan (IP), Atrasos (A). El impacto de los riesgos va a estar determinado por la sumatoria de las consecuencias, en base a tres indicadores: costo, tiempo y grado de insatisfacción del cliente. Luego este resultado es dividido entre la cantidad de riesgos.

En el caso del análisis de las consecuencias respecto al grado de insatisfacción del cliente, se analizó a partir de los parámetros alcance y calidad, para una mejor cuantificación. El valor del grado de insatisfacción del cliente fue dado, para cada riesgo, teniendo en cuenta el por ciento de incumplimiento de requerimientos del cliente. Este por ciento fue el resultado de la división de la cantidad de requerimientos afectados, entre el total de requisitos.

Decisiones incorrectas (DI): consecuencia que va a estar dada por el por ciento de decisiones incorrectas que la provocó.

Nombre: DI.

Grado de solapamiento: 0.

Función de membresía: Bell (campana).

Número de intervalos: 3.

Nombre de los intervalos: A, M, B.

Decisiones incorrectas en el tiempo: tiempo perdido por causa de decisiones incorrectas tomadas.

Nombre: DITiempo.

Grado de solapamiento: 0.

Función de membresía: Symbolic.

Número de intervalos: 0.

Nombre de los intervalos: none.

Decisiones incorrectas en cuanto a costo: por ciento que representa el dinero perdido por causa de decisiones incorrectas tomadas.

Nombre: DICosto.

Grado de solapamiento: 0.

Función de membresía: Symbolic.

Número de intervalos: 0.

Nombre de los intervalos: none.

Decisiones incorrectas en cuanto al grado de insatisfacción del cliente: por ciento que representa el grado de insatisfacción del cliente por causa de decisiones incorrectas tomadas.

Nombre: DIGSCliente.

Grado de solapamiento: 0.

Función de membresía: Symbolic.

Número de intervalos: 0.

Nombre de los intervalos: none.

Incumplimiento del Plan (IP): consecuencia que va a estar dada por el por ciento de incumplimiento del plan que la provocó.

Nombre: IP.

Grado de solapamiento: 0.

Función de membresía: Bell (campana).

Número de intervalos: 3.

Nombre de los intervalos: A, M, B.

Incumplimiento del Plan (IP) en el tiempo: tiempo perdido por causa de incumplimientos en el plan.

Nombre: IPTiempo.

Grado de solapamiento: 0.

Función de membresía: Symbolic.

Número de intervalos: 0.

Nombre de los intervalos: none.

Incumplimiento del Plan (IP) en cuanto a costo: dinero perdido por causa de incumplimientos en el plan.

Nombre: IPCosto.

Grado de solapamiento: 0.

Función de membresía: Symbolic.

Número de intervalos: 0.

Nombre de los intervalos: none.

Incumplimiento del Plan (IP) en cuanto al grado de insatisfacción del cliente: grado de insatisfacción del cliente por causa de incumplimientos en el plan.

Nombre: IPGSCliente.

Grado de solapamiento: 0.

Función de membresía: Symbolic.

Número de intervalos: 0.

Nombre de los intervalos: none.

Atrasos (A): consecuencia que va estar dada por el por ciento de atrasos que la provocó.

Nombre: A.

Grado de solapamiento: 0.

Función de membresía: Bell (campana).

Número de intervalos: 3.

Nombre de los intervalos: A, M, B.

Atrasos en el tiempo: tiempo perdido por causa de atrasos en el desarrollo.

Nombre: A tiempo.

Grado de solapamiento: 0.

Función de membresía: Symbolic.

Número de intervalos: 0.

Nombre de los intervalos: none.

Atrasos en cuento a costo: dinero perdido por causa de atrasos en el desarrollo.

Nombre: ACosto.

Grado de solapamiento: 0.

Función de membresía: Symbolic.

Número de intervalos: 0.

Nombre de los intervalos: none.

Atrasos en cuanto al grado de insatisfacción del cliente: grado de insatisfacción del cliente por causa de atrasos en el desarrollo.

Nombre: AGSCliente.

Grado de solapamiento: 0.

Función de membresía: Symbolic.

Número de intervalos: 0.

Nombre de los intervalos: none.

Frecuencia: el factor de aparición del riesgo en el tiempo de proyecto. Expresa un valor porcentual respecto al tiempo total del proyecto, indicando con qué frecuencia aparece este riesgo.

Nombre: frecuencia.

Grado de solapamiento: 0.

Función de membresía: Symbolic.

Número de intervalos: 0.

Nombre de los intervalos: none.

Certidumbre: valor que indica cuán seguro se está de la información obtenida sobre el riesgo. Valor entre 0 y 1.

Nombre: certidumbre.

Grado de solapamiento: 0.

Función de membresía: Symbolic.

Número de intervalos: 0.

Nombre de los intervalos: none.

Marco de tiempo en el que se mitigó: tiempo mínimo en que se demoró en resolver el riesgo. Debe expresarse en la unidad de medida acordada en el Plan de Gestión de Riesgos, en este caso en semanas.

Nombre: tiempo.

Grado de solapamiento: 0.

Función de membresía: Symbolic.

Número de intervalos: 0.

Nombre de los intervalos: none.

3.2.3. Generación de Reglas Borrosas

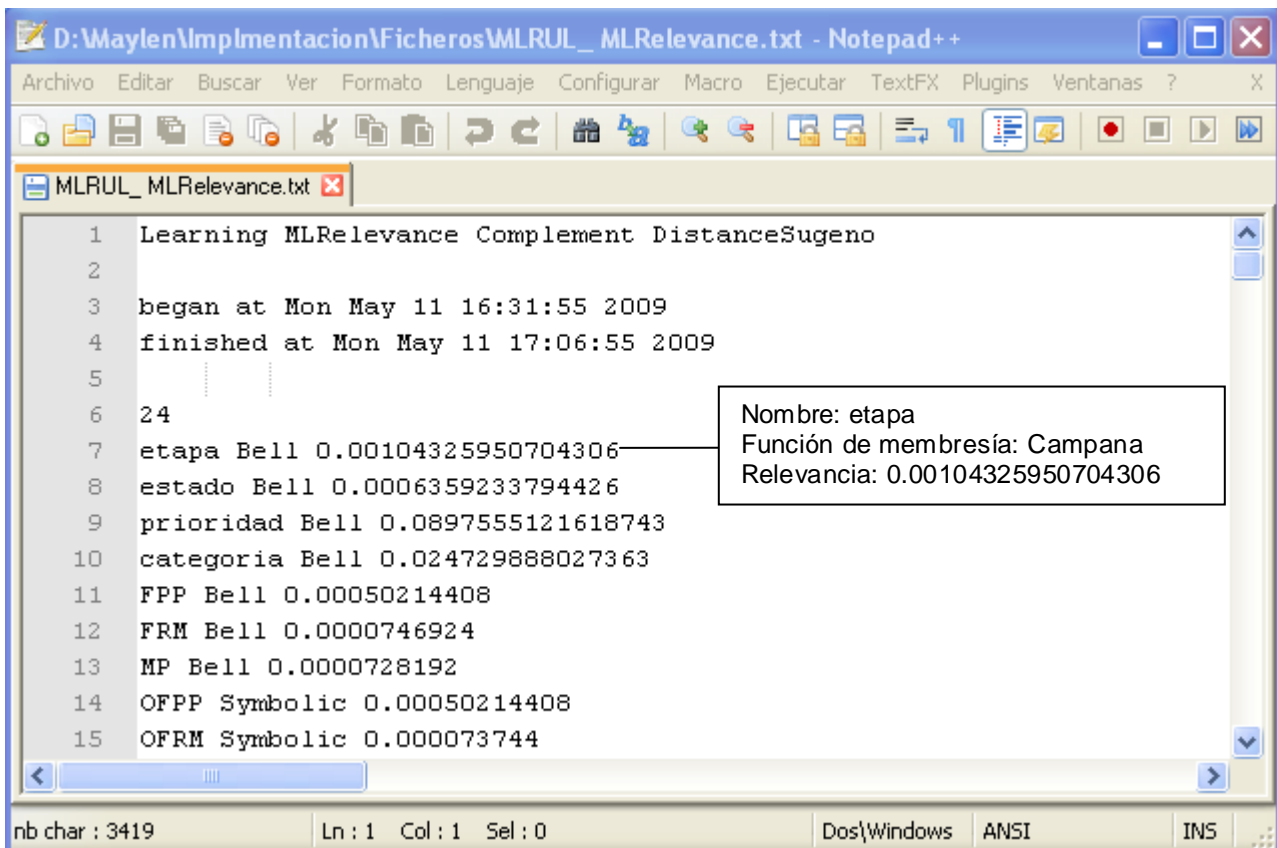
En este Módulo de Aprendizaje, o “Learning”, como se expuso en el 2.1.6, el proceso de construcción de las reglas borrosas estuvo dividido en dos etapas: la de construcción de los conjuntos borrosos y las variables lingüísticas, y la de generación de la base de reglas a partir de las variables lingüísticas construidas. En estas etapas se usó el algoritmo MLRUL para la generación de reglas borrosas.

Los algoritmos que aquí se implementaron, permitieron el aprendizaje automatizado. De esta forma, se pudo crear una base de reglas de aprendizaje, que facilitará la gestión cuantitativa de los riesgos.

Los procesos de Discretización, Clusterización y Fuzzificación de las variables lingüísticas a tratar, el módulo los realiza internamente y es transparente al cliente.

Cada regla permitió clasificar a los riesgos de acuerdo a tres clases: Priorizado (P), No priorizado (NP) y Medianamente Priorizado (MP).

A continuación se muestra, en la Figura 15 y Figura 16, cómo quedaría el fichero que contiene las reglas borrosas.



```
1 Learning MLRelevance Complement DistanceSugeno
2
3 began at Mon May 11 16:31:55 2009
4 finished at Mon May 11 17:06:55 2009
5
6 24
7 etapa Bell 0.00104325950704306
8 estado Bell 0.0006359233794426
9 prioridad Bell 0.0897555121618743
10 categoria Bell 0.024729888027363
11 FPP Bell 0.00050214408
12 FRM Bell 0.0000746924
13 MP Bell 0.0000728192
14 OFPP Symbolic 0.00050214408
15 OFRM Symbolic 0.000073744
```

Nombre: etapa
Función de membresía: Campana
Relevancia: 0.00104325950704306

Figura 15 Fichero de Reglas Borrosas

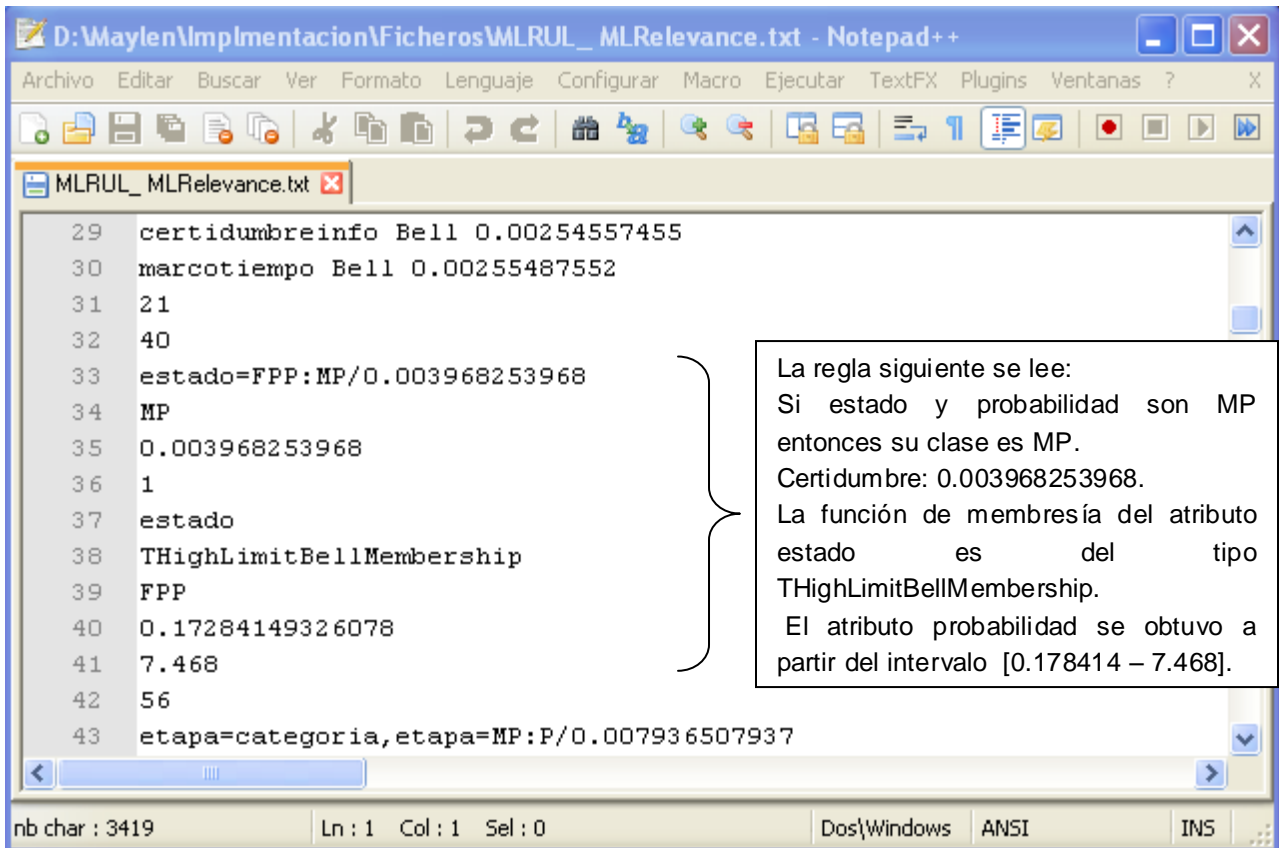


Figura 16 Fichero de Reglas Borrosas

3.2.4. Características de la Implementación

Tal como se había mencionado, la propuesta está basada en técnicas de softcomputing, y de ellas, específicamente, en sistemas de inferencia borrosos.

El objetivo es la obtención de la probabilidad e impacto del riesgo a través del establecimiento de un conjunto de reglas de aprendizaje, sobre la base de un conjunto de riesgos, y la descripción de cada uno de los parámetros para su evaluación previamente establecidos.

3.2.4.1. Arquitectura en la que se enmarca

Como parte del CENTALAD, cuya misión es proveer soluciones integrales y consultorías relacionadas con tecnologías de bases de datos y análisis de información, se encuentra en estos momentos en desarrollo el proyecto Paquete de Análisis para la Toma de Decisiones y Análisis Inteligente de Datos (PATDSI). Esta es una Plataforma Integral para el Análisis de Datos. Incluye: Generador de Reportes, Módulo para la Minería de Datos, Módulo para el Procesamiento Estadístico (SIGE), y posibilita la toma de decisiones y montaje de cuadro de mando integral.

La implementación de los algoritmos para el análisis cuantitativo que se proponen, se insertan en el Módulo para la Minería de Datos, de la forma en que se muestra a continuación, en la Figura 17:

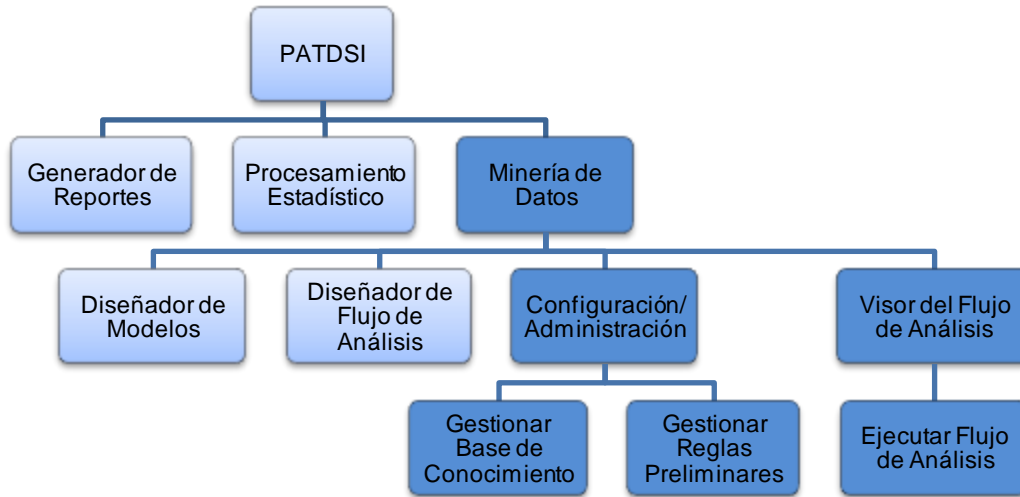


Figura 17 PATDSI

3.2.4.2. Ambiente de desarrollo

En el desarrollo se propone el uso de la arquitectura de PATDSI. (Casares, y otros, 2009)

- **Entorno de Desarrollo Integrado (IDE):**

Eclipse

Es principalmente una plataforma de programación de código abierto y multiplataforma, usada para crear entornos integrados de desarrollo.

Fue desarrollado originalmente por la compañía IBM. Es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto.

Eclipse emplea módulos (en inglés plug-in) para proporcionar toda su funcionalidad, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Esta característica le permite extenderse usando otros lenguajes de programación como son: C/C++ y Python¹⁵.

La versión actual de Eclipse dispone de las siguientes características:

- Editor de texto.
- Resaltado de sintaxis.

¹⁵ Python es un lenguaje de programación interpretado, creado por Guido Van Rossum en el año 1990. La última versión estable del lenguaje es la 3.

- Compilación en tiempo real.
- Pruebas unitarias con JUnit y PyUnit.
- Control de versiones con CVS.
- Integración con Ant¹⁶.
- Asistentes (wizards): para creación de proyectos, clases, tests, etc.
- Refactorización.

Eclipse fue liberado originalmente bajo la Common Public License, pero después fue re-licenciado bajo la Eclipse Public License. La Free Software Foundation ha dicho que ambas licencias son licencias de software libre.

Pydev

Pydev es un módulo que permite utilizar Eclipse para el desarrollo con Python y Jython, convirtiendo a Eclipse en un IDE para Python de primera clase.

Las principales características con que cuenta son:

- resaltado de sintaxis.
- explorador de clases.
- resaltado de errores.
- completamiento de código.

- **Lenguaje de Programación:**

Python

Para la implementación de los algoritmos, que serían parte de una futura herramienta para análisis cuantitativo de riesgos, se utilizó Python. Este es un lenguaje de programación creado por Guido Van Rossum, a principios de los años 90, cuyo nombre está inspirado en el grupo de cómicos ingleses "Monty Python". Es un lenguaje similar a Perl, pero con una sintaxis muy limpia y que favorece un código legible.

Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones de escritorio, hasta servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo.

¹⁶ Apache Ant: herramienta de programación informática para la realización de tareas mecánicas y repetitivas, normalmente durante la fase de compilación y construcción

Se decidió su utilización ya que su sintaxis simple, clara y sencilla; el tipado dinámico, tipos de datos y funciones incorporadas que ayudan a realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero; el gestor de memoria; la gran cantidad de librerías disponibles; por ser un lenguaje multiplataforma (aunque originalmente se desarrolló para Unix, cualquier sistema es compatible con el lenguaje, siempre y cuando exista un intérprete programado para él, ejemplo: Windows, Mac, etc.); y la potencia del lenguaje, entre otros, hacen que desarrollar una aplicación en Python sea sencillo y muy rápido, pues un programa en Python puede tener de 3 a 5 líneas de código menos que su equivalente en Java o C. También porque Python es gratuito, incluso para propósitos empresariales.

Algunos casos de éxito en el uso de Python son Google, Yahoo, la NASA, Industrias Light & Magic, y todas las distribuciones Linux, en las que Python cada vez representa un tanto por ciento mayor de los programas disponibles.

- **Herramienta de modelado:**

- Visual Paradigm**

Como herramienta de modelado se seleccionó Visual Paradigm por las facilidades de uso que brinda, pues es una herramienta CASE que utiliza UML como lenguaje de modelado. Es muy completa y fácil de usar, con soporte multiplataforma, y que proporciona excelentes facilidades de interoperabilidad con otras aplicaciones.

Es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de clases. Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML, al instante, creando de manera simple toda la documentación. Estas funcionalidades podrán usarse para versiones futuras de la implementación propuesta.

Además soporta una serie de lenguajes para la ingeniería inversa, uno de ellos es el Python. Está registrada bajo la licencia de software libre.

- **Herramienta para pruebas de unidad:**

- PyUnit**

Es la herramienta o framework oficial para hacer pruebas unitarias en Python. Se incluye en la librería estándar (en el módulo unittest) y ha sido creado por los desarrolladores de JUnit para facilitar la creación y gestión de tests de prueba en módulos Python.

Este framework es soportado perfectamente por el IDE Eclipse con la extensión PyDev.

Permite separar el código de pruebas del código del propio proyecto para mejorar la comprensión, refactorización y la facilidad, y para hacer cambios en el programa. Por otra parte, se pueden realizar los tests por líneas de comandos.

Las pruebas que es capaz de realizar son pruebas de igualdad, de excepciones, de fallo, pruebas de cordura (comprobar que no se producen situaciones imposibles) y pruebas de mayúsculas (relacionadas con las cadenas de caracteres).

Debido a todo esto, y a que se trata de la herramienta estándar de Python, tiene mucha difusión y existe mucha documentación sobre esta librería en la red, es la que se ha elegido para realizar las pruebas unitarias de la implementación propuesta.

3.2.4.3. Diseño de Clases

A continuación se muestra la estructura de los módulos y paquetes de Python, el diagrama de las clases del diseño más representativas, (ver Figura 18) y los métodos y responsabilidades de algunas de ellas. (Ver Figura 19, Figura 20 y Figura 21).

Acorde con la estructura del lenguaje utilizado: Python, las clases con funcionalidades similares son agrupadas en módulos, los cuales a su vez forman los paquetes, en nuestro caso el “Learning” y el Clasificador.

- **Módulos:**

uUtils: módulo auxiliar para el ordenamiento y el trabajo con determinados valores.

uPartition Queue: estructura del TDA cola, con sus funcionalidades clásicas.

uDataSet: simula la estructura de un Data Set, con sus funcionalidades características.

uLoadProc: módulo utilizado para el manejo de ficheros.

uSpecialiceTrainSet: módulo que carga el conjunto de entrenamiento.

uAttribute: clase auxiliar para el trabajo con algunos parámetros.

uRule: módulo que contiene las características de una regla.

uSet: módulo auxiliar.

uLinguistic variable: conjunto de clases para el trabajo con las variables lingüísticas.

uMemberShipFunction: conjunto de clases para el manejo de las funciones de membresía de las variables lingüísticas y numéricas.

uCluster: módulo para el trabajo de clusterización.

uDiscretiser: módulo para el trabajo de discretización.

uGeneralFuzzyRule: módulo general que realiza las funcionalidades principales.

uMLRuIFuzzyRule: hereda de uGeneralFuzzyRule y realiza la generación de reglas a través del método MLRUL.

- **Diagrama de clases**

A continuación se muestran algunas de las clases más representativas implementadas, y sus principales funcionalidades.

- **Clases más representativas:**

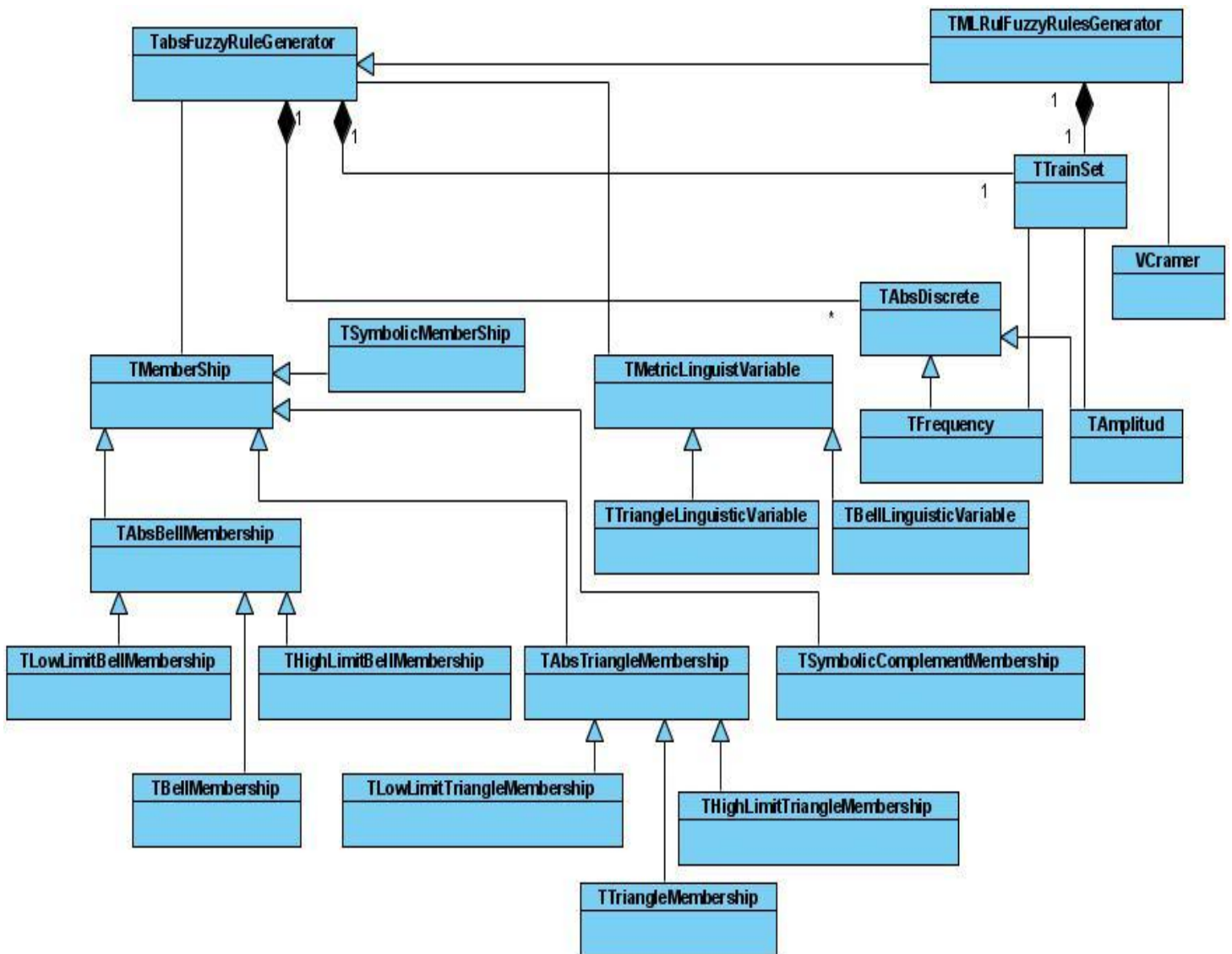


Figura 18 Diagrama de clase s más representativas

TabsFuzzyRuleGenerator: es la clase más general, es una clase abstracta que utiliza otros módulos tales como: uSpecializeTrainset, uLinguisticVariable, uRule, uDiscretizer, uUtils, uAttribute, uMembershipFunctions, uCluster, presentes en el paquete “Learning”, y otros propios del intérprete, tales como: os y sys. Es la encargada de la generación de las reglas borrosas. De ella hereda la clase TMLRuIFuzzyRulesGenerator, implementando algunas de sus funcionalidades.

TMLRuIFuzzyRulesGenerator: es la clase con la funcionalidad principal, contiene al método DoFuzzyRule que es el método fundamental que orienta los algoritmos a través de la llamada a métodos auxiliares. Utiliza otros módulos tales como: uGeneralFuzzyRulesGenerator, VCramer, uSpecializeTrainset, uLinguisticVariable, uRule, uDiscretizer, uUtils, uAttribute, uMembershipFunctions, uParttionQueue, presentes en el paquete “Learning”, y otros propios del intérprete tales como: os¹⁷, sys¹⁸ y math¹⁹.

- **Clases auxiliares:**

TTrainSet: clase encargada del manejo de los casos de entrenamiento.

VCramer: clase encargada del método Cramer.

- **Conjunto de clases encargadas del manejo de los discretizadores:**

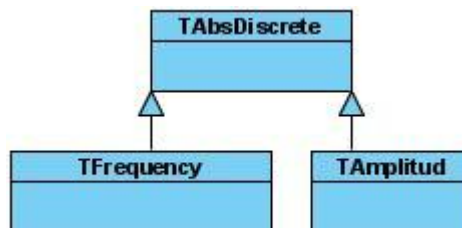


Figura 19 Diagrama de clases: Discretizadores

TabsDiscrete: es la clase más general para el manejo de los discretizadores, de ella heredan TFrequency y TAmplitud.

TFrequency: es la clase responsable de la distribución de los intervalos de las variables lingüísticas, en este caso los intervalos pueden tener diferente tamaño.

¹⁷ El módulo os contiene muchas funciones y datos, presente en Python 2.6.

¹⁸ El módulo sys contiene información sobre el sistema, tal como la versión de Python que se ejecuta, o el nivel de recursión máximo permitido, además es un diccionario que contiene todos los módulos que se han importado desde que arrancara Python, entre otras funciones.

¹⁹ El módulo math permite el acceso a las funciones de la biblioteca C subyacente para las matemáticas de coma flotante.

TAmplitud: clase encargada de la distribución de los intervalos de las variables lingüísticas, en este caso los intervalos pueden tener igual tamaño.

- **Conjunto de clases encargadas del manejo de las variables lingüísticas:**

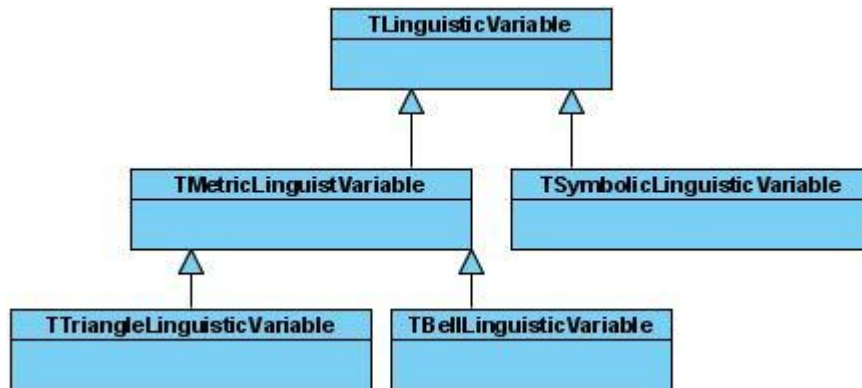


Figura 20 Diagrama de clases: Variables Lingüísticas

TLinguisticVariable: es la clase más general, la responsable de crear las variables lingüísticas.

TMetricLinguisticVariable: hereda de **TLinguisticVariable**, y es la encargada de las variables cuyas funciones de membresía son de tipo campana y triángulo, tales como: **TSymbolicLinguisticVariable** y **TTriangleLinguisticVariable**, que heredan de ella.

TTriangleLinguisticVariable: es la clase autorizada al manejo de las variables numéricas para el caso de funciones de membresía de tipo triángulo.

TBellLinguisticVariable: es la clase responsable del manejo de las variables numéricas para el caso de funciones de membresía de tipo campana.

TSymbolicLinguisticVariable: es la clase encargada de las funciones de las variables cuya función de membresía es de tipo simbólica.

- **Conjunto de clases encargadas del manejo de las funciones de membresía:**

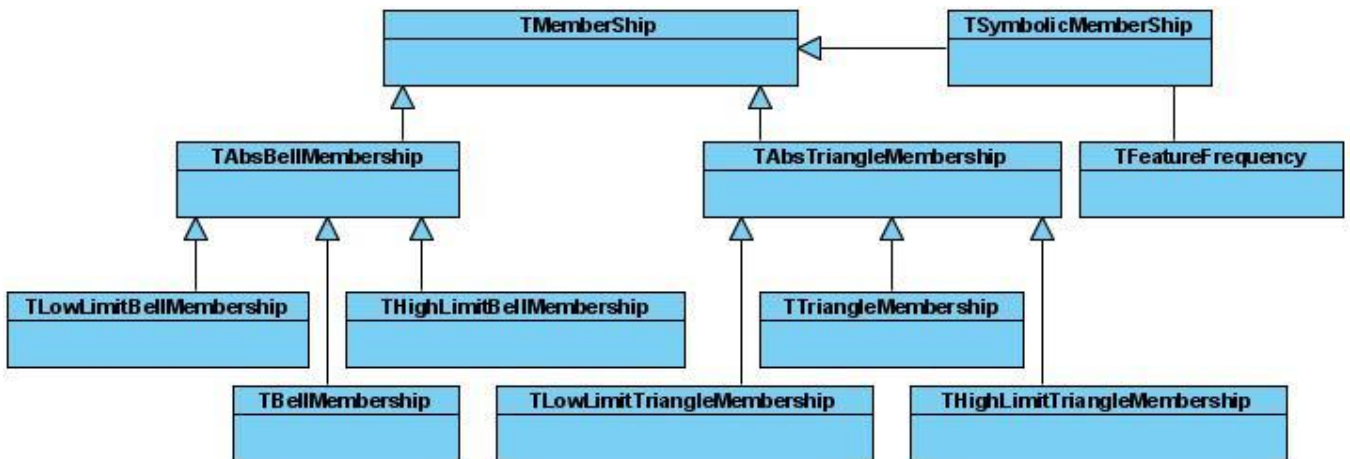


Figura 21 Diagrama de clases: Funciones de Membresía

TMemberShip: es la clase más general, encargada de crear las funciones de membresía.

TSymbolicMemberShip: es la clase encargada de la creación de las funciones de membresía de tipo simbólica.

TAbsBellMembership: es la responsable de de la creación de las funciones de membresía de tipo campana.

TLowLimitBellMembership: es la clase facultada de encontrar el límite inferior de la función de membresía de tipo campana.

THighLimitBellMembership: es la clase autorizada de encontrar el límite superior de la función de membresía de tipo campana.

TBellMembership: hereda de la clase TMemberShip y trabaja con las funcionalidades de la función de membresía de tipo campana.

TAbsTriangleMembership: es la clase encargada de la creación de la función de membresía de tipo triángulo.

TLowLimitTriangleMembership: es la clase responsable de encontrar el límite inferior de la función de membresía de tipo triángulo.

THighLimitTriangleMembership: se encarga de encontrar el límite superior de la función de membresía de tipo triángulo.

TTriangleLinguisticVariable: hereda de la clase TMemberShip y es la responsable de la creación de las funciones de membresía de tipo triángulo.

3.2.4.4. Patrones de Diseño

Los patrones de diseño son una solución estándar para un problema común de diseño dentro de un contexto dado y constituyen una manera más práctica de describir ciertos aspectos de la organización de un programa. Estos permitieron dar flexibilidad y extensibilidad al diseño en cuestión.

- *Experto*: se usó para que cada objeto realizara la funcionalidad de acuerdo a la información que domina, por lo que aseguró que se le asigne determinada responsabilidad a la clase que mayor información posee para cumplir dicha tarea.
- *Creador*: se usó para guiar la asignación de responsabilidades con la creación de objetos, con propósito fundamental de encontrar un creador que se deba conectar con el objeto producido en cualquier evento.
- *Bajo Acoplamiento*: se usó para asignar una responsabilidad de modo que su colocación no incremente el acoplamiento, con el objetivo de diseñar clases más independientes, que reduzcan el impacto de los cambios y sean más reutilizables.
- *Alta Cohesión*: se usó cuando el objeto tenía delimitadas sus responsabilidades, es decir, para no asignar responsabilidades que no estuvieran limitadas dentro de sus funciones.
- *Singleton*: Los patrones GoF²⁰ y los de Python pueden beneficiarse mutuamente. Mientras que a Python le faltan algunas características que el GoF asume, no es imposible construir implementaciones de los patrones que funcionen como sus contrapartes del GoF. En el caso de la implementación del Singleton sustituyó los constructores privados con un mecanismo de excepción y un método de clase con una función regular pero el patrón subyacente es claramente el Singleton. La naturaleza flexible y dinámica del lenguaje provee una buena base para una variedad de distintas y elegantes soluciones. Este patrón forma parte de los patrones GoF, específicamente al grupo de patrones creacionales.

En la propuesta, se diseñó asignando responsabilidades de modo que la cohesión sea alta, los métodos tienen bien delimitadas sus responsabilidades, no recargando en ningún caso sus funcionalidades, permitiendo mayor eficiencia y que el tiempo de respuesta y ejecución no exceda lo estimado, generando por ende bajo acoplamiento.

3.2.4.5. Métricas de Diseño

²⁰ Patrones GoF: Llamados así por ser creados por el grupo Gang of Four (**GoF**), compuesto por Erich Gamma, Richard Helm, Ralph Johnson y John Visides. Son considerados patrones GoF una serie de patrones de los grupos: patrones creacionales, estructurales, de comportamiento y de sistema.

Las métricas de diseño permiten obtener una mejor visión interna del sistema, lo que ayuda a que el diseño evolucione a niveles superiores de calidad.

En su libro sobre métricas orientadas a objetos, Lorenz y Kidd separan las métricas basadas en clases en cuatro amplias categorías: tamaño, herencia, valores internos y valores externos.

Las métricas orientadas al tamaño para las clases orientadas a objetos se centran en el recuento de atributos y operaciones para cada clase individual, y los valores promedio para el sistema orientado a objetos como un todo. Las métricas basadas en la herencia se centran en la forma en que las operaciones se reutilizan en la jerarquía de clases. Las métricas para valores internos de clase examinan la cohesión y los aspectos orientados al código; las métricas orientadas a valores externos, examinan el acoplamiento y la reutilización.

Para comprobar el grado de calidad y fiabilidad del diseño se aplicaron algunas métricas de diseño basadas en clases para medir categorías tales como tamaño, herencia, nivel de profundidad, pues están orientadas a la cohesión, acoplamiento y reutilización.

- **Métrica Tamaño Operacional de Clase (TOC):**

Conocida también con el nombre de Métrica Tamaño de Clase (TC), mide el tamaño de la clase basándose en el número de métodos que posee.

Como puede observarse a continuación, la mayoría de las clases tienen responsabilidad baja (Tabla 8), o sea, pocos procedimientos (Figura 22), una baja complejidad (Tabla 9) y una alta reutilización (Tabla 10).

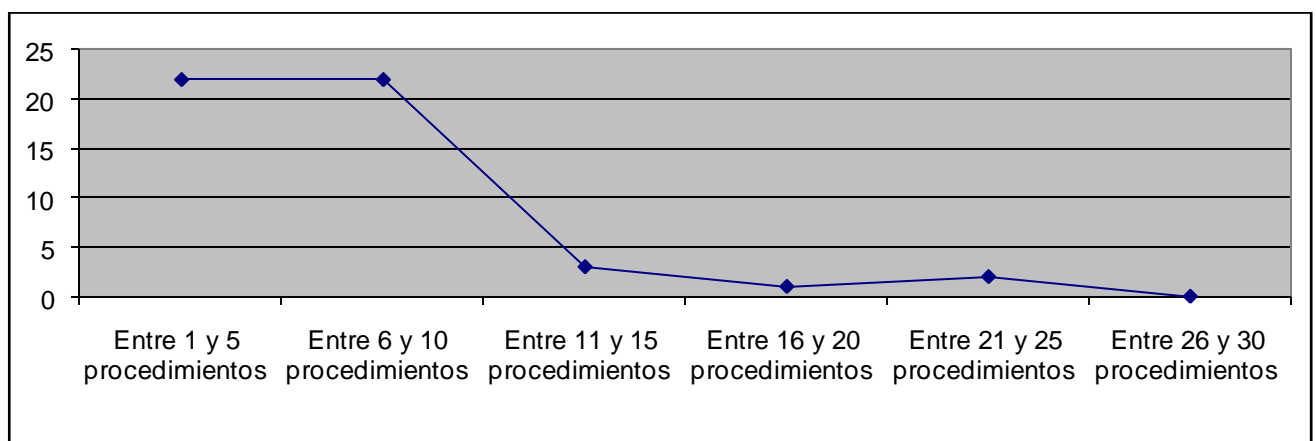


Figura 22 Procedimientos de las clases

Tabla 8 Responsabilidad de las clases

Responsabilidad	Cantidad de clases
Baja	38
Media	9
Alta	5

Tabla 9 Complejidad de las clases

Complejidad	Cantidad de clases
Baja	38
Media	9
Alta	5

Tabla 10 Reutilización de las clases

Reutilización	Cantidad de clases
Alta	38
Media	9
Baja	5

- **Métrica Profundidad de Herencia (PH):**

Conocida además como Métrica Árbol de Profundidad de Herencia (APH).

Como resultado de la aplicación de esta métrica para medir la profundidad de herencia de las clases, se obtuvo, que existe poca profundidad de herencia, como se muestra a continuación en la Figura 23.

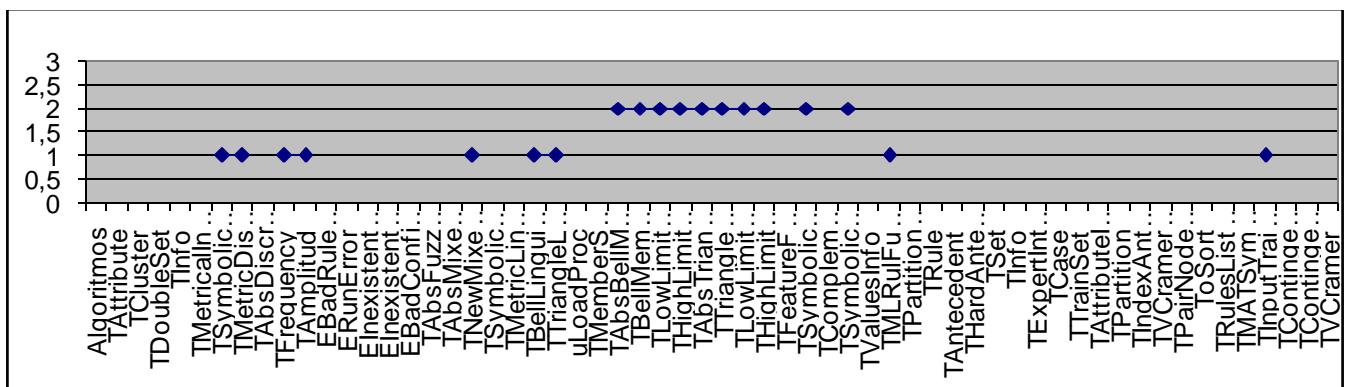


Figura 23 Profundidad de herencia de las clases

3.2.4.6. Diagrama de componentes

A continuación se muestra cómo resultaría la implementación de la futura herramienta. Como se explicó anteriormente, el Módulo de Aprendizaje o “Learning” genera un fichero de reglas de

aprendizaje, que el módulo clasificador debe consultar para la obtención del impacto y la probabilidad de los riesgos, en el fichero de salida final o de clasificación. Es válido destacar nuevamente que en la presente investigación se implementó el módulo de aprendizaje o “Learning”, para la validación del modelo propuesto.

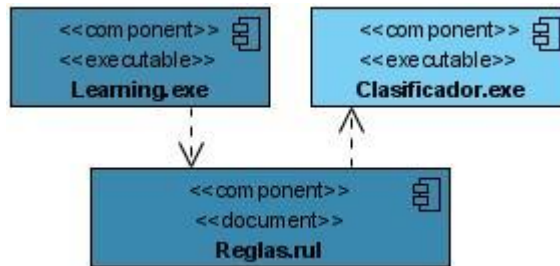


Figura 24 Diagrama de Componentes

3.2.4.7. Pruebas de Unidad

La prueba de unidad o unitaria, es una técnica que permite determinar si un programa cumple con los requisitos solicitados. Es una forma de probar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado. Aunque no descubrirán todos los errores del código, ya que sólo prueban las unidades por sí solas, y por lo tanto, no descubrirán errores de integración, problemas de rendimiento y otros problemas que afectan a todo el sistema en su conjunto; las pruebas de unidad, por definición, cubren la funcionalidad propia del módulo tanto con una perspectiva de caja blanca como de caja negra; pero prestando poca o ninguna atención a la integración con otros módulos. Como se mencionó anteriormente, se usó el Pydev como herramienta oficial para realizar estas pruebas en el lenguaje Python, específicamente con el IDE Eclipse. Para la realización de las pruebas se utilizó el PyUnit, framework estándar para desarrollar los casos de prueba de unidad, para programas desarrollados en el lenguaje utilizado. Para poder hacer un caso de prueba propio en PyUnit es necesario hacer una subclase de "TestCase", que es un objeto que puede correr un único método de prueba. Para realizar una prueba, en Python, se usa el comando "assert", si este comando falla cuando se ejecuta el caso de prueba PyUnit, dará el caso por fallido, de lo contrario el resultado es correcto.

Se realizaron pruebas al módulo y el resultado fue satisfactorio como se muestra en la siguiente figura:

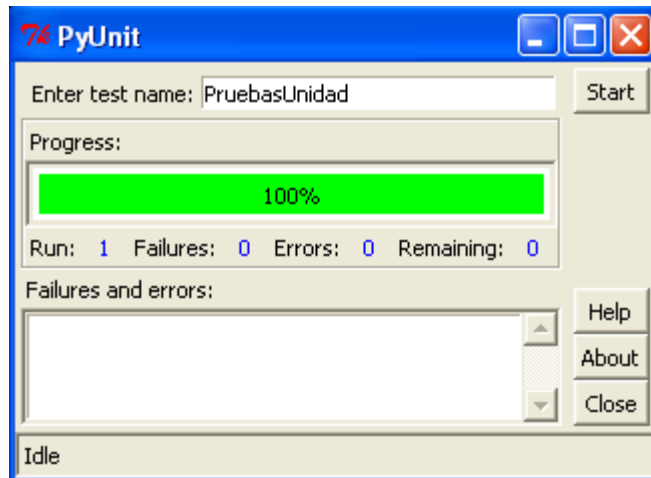


Figura 25 Resultado de las pruebas de unidad

3.2.4.8. Estándar de codificación

Un aspecto muy importante para programar es definir el “estilo” de programación que este utiliza, porque los estilos de código permiten ver con orden y legibilidad el código escrito. Se propone que la implementación desarrollada en este trabajo sea usada para mejorarla, para crear una herramienta completa y robusta para el análisis cuantitativo de los riesgos, y para que el personal que trabaje con estos resultados pueda entender fácilmente el código implementado.

Estándar de Nomenclatura:

- Nomenclatura de las clases:

Los nombres de las clases comienzan con la letra T, a continuación el nombre de la clase en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación *PascalCasing*²¹.

Ejemplo: TAttribute.

- Nomenclatura de las funciones:

El nombre a emplear para las funciones se escribe con la primera palabra en mayúscula, en caso de que sea un nombre compuesto se empleará notación *CamelCasing*²², en inglés y con sólo leerlo se reconoce el propósito de la misma.

²¹ Pascal-Casing es como la notación húngara pero sin prefijos. En este caso, los identificadores y nombres de variables, métodos y funciones están compuestos por múltiples palabras juntas, iniciando cada palabra con letra mayúscula. Ejemplo: *DoSomething*: Este nombre de método está compuesto por dos palabras, ambas iniciando con letra mayúscula. La notación húngara se basa en definir prefijos para cada tipo de datos y según el ámbito de las variables. Ejemplo: *intEdad*, es de tipo Integer y representa la edad de alguna persona

Ejemplo: CombineNearValue.

- Nomenclatura de las constantes:

El nombre a emplear para las constantes se escribe con todas las letras en mayúscula.

Ejemplo: CLUSTERTYPE.

Normas de comentariado:

- Se comentaría alguna función que no esté claro su funcionamiento. Se escribe de la siguiente forma:

Aquí va el comentario.

Estilo del Código:

- Declaraciones dentro del cuerpo de la clase:

Las clases se comienzan a declarar pegado al margen izquierdo.

Ejemplo:

```
class TAttribute :  
    #fName          : string  
    #fValue         : string  
    #fDescription   : string  
  
    def __init__(self, fName = ''):  
        self.fName = fName
```

- Declaraciones dentro del método / el cuerpo de la función:

Las declaraciones dentro del método / el cuerpo de la función se realizan como se muestra en el ejemplo.

Ejemplo:

```
class TAttribute :  
    #fName          : string  
    #fValue         : string  
    #fDescription   : string
```

²² Camel-Casing es común en Java. Es parecido al Pascal-Casing con la excepción que la letra inicial del identificador no debe estar en mayúscula. Ejemplo: *doSomething*: Este nombre de método está compuesto por dos palabras, la primera con todo en minúsculas y la segunda iniciando con letra mayúscula.

```
def __init__(self, fName = ''):
    self.fName = fName
def GetName(self) : #string
    return self.fName
```

- Sangría o Indexado:

La declaración de las funciones se separa un con tab y dos espacios del margen izquierdo, el código dentro de la función se separa de la declaración de la misma con un tab, así como los niveles de indentación dentro de las sentencias de control.

3.2.5. Generación de Artefactos Resultantes del Análisis

Una vez calculada la probabilidad y el impacto por el módulo clasificador, se deben llenar los artefactos de salida propuestos en el epígrafe 2.1.8, el Registro de Riesgos y la Lista Priorizada de Riesgos. Al registro se le incluirá el impacto y la probabilidad de cada riesgo (Ver Anexo 6: Registro de Riesgos), y la Lista Priorizada se actualizará priorizando nuevamente los riesgos, usando una matriz de Probabilidad/Impacto para reagruparlos, teniendo en cuenta que ya se conocen su probabilidad de ocurrencia y su impacto, clasificándolos en Priorizados, No Priorizados o Medianamente Priorizados. (Ver Anexo 7: Lista Priorizada de Riesgos Cuantificados). A continuación se muestra un ejemplo de Registro de Riesgo y de Lista Priorizada de Riesgos.

- **Ejemplo de Registro de Riesgo (actualizado)**

ID: 1	Etapas: Elaboración	Estado: Activado	Prioridad: Alta
Categoría: Ingeniería		Entrevistado: ERP	
Definición del riesgo: Desconocimiento de las tecnologías (evento): Desconocimiento de las tecnologías (consecuencia): Decisiones incorrectas			
Causas: Falta de preparación			
Posible Probabilidad (opcional): Alta		Posible Impacto (opcional): Alto	

Probabilidad: 0,40	Impacto: 0.50
Frecuencia: Alta	Certidumbre: 1

Marco de tiempo: 8 semanas	Estrategia de respuesta: cursos de capacitación
Responsables: Pepe Rodríguez, arquitecto	
Riesgos Relacionados: 2,3,6	
Observaciones:	

• **Ejemplo de Lista Priorizada de Riesgos (actualizada):**

ID	Riesgo	Prioridad	Mayor Contingencia de Costes	Mayor probabilidad de influir sobre el camino crítico
IPERP3	Atraso en el plan de capacitación	Priorizado		X
IPERP1	Desconocimiento de las Tecnologías	Priorizado	X	X
IPERP4	Poco conocimiento del negocio	Priorizado		X
IPERP8	Inexperiencia organizativa del equipo	M. Priorizado		
IPERP13	Carga extra del personal.	M. Priorizado		X

3.3. Análisis del modelo

En el presente trabajo se muestra el modelo desarrollado en su totalidad, y la implementación de los algoritmos que se propone para su aplicación en un 60%. Se diseñó e implementó el módulo de aprendizaje y se abordaron los algoritmos del módulo clasificador, el cual se continuará desarrollando como parte del proyecto en el que se inserta la implementación desarrollada como parte de la presente investigación.

En el mundo existen otros modelos como los abordados en el Capítulo 1, pero como se demostró, la mayoría no aborda de forma completa el análisis cuantitativo de la gestión de riesgos, o es abordada de forma breve e inexacta en la mayoría de los mismos. En el presente trabajo se propone un modelo que afronta el tema con los puntos establecidos para la definición de un modelo, tales como: personal involucrado, entradas, salidas, así como la implementación de algoritmos que ayudarían a crear una herramienta para análisis cuantitativo de riesgos, y esta a su vez permitiría una perfecta aplicación del modelo propuesto.

Se propone que en futuras versiones se finalice su implementación, concluyendo el módulo clasificador que permitirá la obtención de la probabilidad y del impacto del riesgo analizado; así como el desarrollo

de la base de casos en una base de datos, con el gestor PostgreSQL²³, permitiendo así un mejor control de los cambios.

3.3.1. Evaluación de las variables independientes

A continuación se muestran, en la Tabla 11, los resultados de la evaluación de las variables independientes de la investigación, o sea, el modelo para el análisis cuantitativo de los riesgos, y la implementación de los algoritmos.

Tabla 11 Evaluación de variables

VARIABLES INDEPENDIENTES	DIMENSIÓN	INDICADORES	UNIDADES
Modelo para el análisis cuantitativo de los riesgos.	Representación	Representación gráfica	_X_Sí __No
		Grado de completitud y claridad de la documentación	_X_Alto __Medio __Bajo
	Aplicabilidad del modelo.	Aplicable a proyectos de gestión.	_X_Sí __No
		Estimar el impacto de forma cuantitativa exacta	_X_Sí __No
		Estimar la probabilidad de ocurrencia	_X_Sí __No
Implementación de los Algoritmos	Complejidad	Complejidad Espacial	Memoria(1994 KB)
		Complejidad Temporal	$O(x^2 * r)$
	Sencillez	_X_Sí __No	

3.3.2. Análisis de Complejidad

Algoritmo MLRUL

A continuación se muestra la complejidad de algunos de los pasos de los algoritmos secundarios del MLRUL (Tabla 12, Tabla 13, Tabla 14 y Tabla 15), y del MLRUL en general.

- **Creación de clústeres de un atributo continuo:**

Tabla 12 Creación de clústeres

Acción	Complejidad
--------	-------------

²³ PostgreSQL 8.3: Es un poderoso sistema de gestión de base de datos relacional orientada a objetos de software libre. Tiene una arquitectura probada por lo que ha ganado una sólida reputación para la fiabilidad, integridad y exactitud de los datos. Funciona en todos los principales sistemas operativos, incluyendo Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows.

I.	Ordenar el conjunto original de datos	utilizando el algoritmo QuickSort ²⁴ , es $O(r \cdot n \cdot \ln(n))$
II.	Calcular las distancias entre los datos numéricos tomados dos a dos	$O(n^2 r)$
III.	Calcular el valor de umbral	$O(n \cdot r q)$
IV.	Crear clústeres	Despreciable
V.	Unir los clúster adyacentes	$O(r \cdot n^2)$

- **Construcción de particiones**

Tabla 13 Creación de particiones

Acción	Complejidad
I. Crear la colección de conjuntos abiertos.	$O(n)$
II. Crear la colección de conjuntos cerrados	Despreciable
III. Seleccionar conjunto abierto	Despreciable
IV. Seleccionar el atributo más relevante	$O(r^2 n)$
V. Particionar el conjunto abierto	$O(r \cdot n \cdot \ln(n))$
VI. Chequear si los subconjuntos son abiertos o cerrados y añadirlos a la colección de conjuntos que corresponda	$O(n)$

- **Generación de una regla borrosa a partir de un caso y sus tributos:**

Tabla 14 Generación de reglas borrosas

Acción	Complejidad
I. A partir de cada atributo del caso, construir un antecedente de la regla	Despreciable
II. Formar una regla por la combinación de los antecedentes construidos	n.r.

- **Refinación de la base de reglas candidatas.**

Tabla 15 Refinación de la base de reglas candidatas

Acción	Complejidad
I. Para cada regla de la base de reglas candidatas y para cada caso del conjunto de entrenamiento: adicionar el caso al conjunto de casos cubiertos por la regla.	$O(r \cdot n^2)$.
II. Inicializar los conjuntos de casos cubiertos, la base de reglas final y el conjunto total de los casos	$O(n)$
III. A partir de cada atributo del caso, construir un antecedente de la regla	$O(n^2)$

²⁴ El ordenamiento rápido (QuickSort) es un algoritmo basado en la técnica de divide y vencerás, que permite, en promedio, ordenar n elementos en un tiempo proporcional a $n \log n$.

IV. Formar una regla por la combinación de los antecedentes construidos	n.r
---	-----

Finalmente la complejidad del algoritmo MLRUL a partir del análisis de la complejidad de cada uno de sus pasos es un $O(r \cdot n^2)$.

3.4. Conclusiones Parciales

Para la validación del modelo propuesto se desarrolló un ejemplo práctico de cómo llevar a cabo cada uno de los pasos del modelo propuesto, se precisaron características de la implementación y se realizó además un análisis del modelo propuesto, evaluando las variables independientes de la investigación.

Específicamente en cada uno de los pasos se validó de la siguiente forma:

- En la construcción de la base de riesgos, se completaron los parámetros de cada uno de los riesgos analizados.
- Durante la construcción de la base de casos de entrenamiento, se elaboraron las variables lingüísticas y de cada una de ellas los conjuntos borrosos asociados, definiéndose sus valores y elaborándose los ficheros de entrada.
- Para la generación de reglas borrosas, se desarrolló el módulo de aprendizaje, basado en el algoritmo MLRUL.

Durante la generación de artefactos resultantes del análisis, se llenaron las planillas con posibles resultados previstos de la implementación del modelo.

Conclusiones

A partir de los resultados de la investigación, del desarrollo y de la validación del modelo propuesto para el análisis cuantitativo de riesgos, se arriba a las siguientes conclusiones:

- Como resultado del estudio del estado del arte del análisis cuantitativo de los riesgos en proyectos de software, de los principales centros de pensamiento, metodologías, herramientas y modelos existentes, se determinó que no existen herramientas que brinden un soporte adecuado y estandarizado para las necesidades actuales de la gestión de riesgos. Además se estableció que el modelo desarrollado sería un híbrido entre modelos matemáticos y basados en la experiencia.
- El desarrollo del modelo basado en técnicas de softcomputing permitirá analizar la probabilidad de ocurrencia y el impacto de los riesgos en proyectos de software, pues se especificó el alcance del mismo, los principios y premisas para su aplicación y los elementos que garantizan el funcionamiento del modelo, así como una representación del mismo con la definición de las actividades a realizar durante este proceso con sus entradas y salidas.
- Como parte de la validación del modelo: se construyó una base de riesgos para un mejor procesamiento de los mismos, se implementó el Módulo de Aprendizaje que permitió generar las reglas borrosas, y que se usarán en futuras versiones por el Módulo Clasificador para calcular la probabilidad y el impacto de los riesgos, se concretaron los artefactos que se deben generar luego del proceso de análisis cuantitativo, aplicando el modelo propuesto, para documentar el proceso y contribuir a la mitigación de los riesgos.

Recomendaciones

- Aplicar el modelo propuesto en el resto de los procesos productivos de la UCI.
- Profundizar en la asignatura de Ingeniería de Software sobre la gestión de riesgos y dentro de esta, el análisis cuantitativo de riesgos.
- Desarrollar futuras versiones de la implementación propuesta, integrando el Módulo Clasificador que se está desarrollando en el proyecto PATDSI, e integrar la herramienta en la plataforma de teletrabajo y gestión de proyectos del centro de datos CENTALAD, para lograr una herramienta completa para análisis cuantitativo de riesgos, con las características de ser multiplataforma, con interfaz visual, usando como Sistema Gestor de Base de Datos: PostgreSQL.
- Aplicar técnicas de limpieza de datos²⁵ en el proceso de Construcción de Base de Casos de Entrenamiento. (Ver Anexo 8: Ejemplo de Limpieza de Datos)
- Continuar la investigación y posterior perfeccionamiento de este modelo con otras tecnologías existentes.

²⁵ La limpieza de datos es un proceso que consiste en aplicar una serie de tratamientos a datos con el objetivo de obtener datos más fiables, que aporten información más consistente, y que faciliten su utilización en procesos posteriores. Es el proceso de corregir o remover información incorrecta, con formato inapropiado, o duplicada. Los datos se pueden examinar para encontrar fallas mediante la utilización de reglas, algoritmos y tablas de búsqueda. Involucra descomposición y re ensamblaje de datos.

Bibliografía

2006. *Gestión de riesgos en ingeniería del software*. [En línea] Universidad de Murcia, 2006. [Citado el: 24 de Octubre de 2008.] <http://www.um.es/docencia/barzana/IAGP/lagp5.html>.

Métricas para Sistemas Orientados a Objetos. [En línea] [Citado el: 10 de Diciembre de 2008.] http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/gonzalez_d_h/capitulo6.pdf.

Gestión de Riesgos en Proyectos de Software. Guía de Aprendizaje. [En línea] [Citado el: 10 de Diciembre de 2008.] <http://alarcos.inf-cr.uclm.es/doc/pgsi/doc/teo/7/pgsi-t7.pdf>.

2009. *Entorno de Análisis de Riesgos. EAR/PILAR*. [En línea] 2009. [Citado el: 10 de Abril de 2009.] <http://www.ar-tools.com/>.

Aguilar, Catherine. 2005. *Aplicación de Conceptos de Gestión de proyectos y gestión de riesgo en el desarrollo de productos nuevos en el campo de tecnología de información*. [En línea] 2005. [Citado el: 12 de Marzo de 2009.] <http://grad.uprm.edu/tesis/aguilarramos.pdf>.

Alquezar, R. y Otros. *Investigación actual del grupo SOCO: Metodologías híbridas de Softcomputing*. [En línea] [Citado el: 29 de Noviembre de 2008.] <http://www.lsi.us.es/redmidas/Capitulos/LMD19.pdf>.

Ávila, Henry. 2007. *Análisis de Riesgos en la Planificación de proyectos Informáticos para el Sistema de Salud Cubano en la Facultad 7*. Universidad de las Ciencias Informáticas. La Habana : s.n., 2007.

Benito, Sully. 2007. *Propuesta de una guía para la planificación de los proyectos productivos en la Facultad 3 de la UCI*. Universidad de las Ciencias Informáticas. La Habana : s.n., 2007.

Booch, Grady, Jacobson, Ivar y Rumbaugh, James. 2000. *Proceso Unificado de Desarrollo de Software*. 2000. Vols. 84-7829-036-2.

Casares, Thais y Cazorla, José M. 2009. *Propuesta de arquitectura para la herramienta de minería de datos de PATDSI*. Universidad de las Ciencias Informáticas. La Habana : s.n., 2009.

2006. Comisión de Reglamentos Técnicos y Comerciales. *Norma Técnica Peruana NTP-ISO/IEC 12207*. [En línea] 2006. [Citado el: 22 de Octubre de 2008.] <http://www.bvindecopi.gob.pe/normas/isoiec12207.pdf>.

1997. Consejo Superior de Informática. *Eurométodo V1*. [En línea] 1997. [Citado el: 29 de Octubre de 2008.] http://www.csae.map.es/csi/pdf/em_v1.pdf.

Corzo, Dayani. 2008. *Herramienta para el análisis y gestión de riesgos de la seguridad informática para BANDEC*. Universidad de las Ciencias Informáticas. La Habana : s.n., 2008.

Cuza, Betsy y Yanes, Rachel. 2007. *Gestión de Riesgos en el proyecto de Informatización del Conocimiento Geológico en Cuba*. Universidad de las Ciencias Informáticas. La Habana : s.n., 2007.

Del Carpio, Javier. 2006. *Análisis del riesgo en la administración de proyectos de tecnología de información*. [En línea] 2006. [Citado el: 29 de Noviembre de 2008.] <http://www.scielo.org.pe/pdf/id/v9n1/a13v9n1.pdf>.

2008. Departamento de Ciencias de la Computación e Inteligencia Artificial. *¿Qué es Softcomputing?* [En línea] 2008. [Citado el: 30 de Noviembre de 2008.] http://modo.ugr.es/es/soft_computing.

Departamento de Informática. *Gestión de Riesgos*. [En línea] [Citado el: 10 de Abril de 2009.] <http://www.di.uniovi.es/~aquilino/Asignaturas/ProyectosInformatica/Documentos/07-GestionRiesgos.pdf>.

Dirección General de Protección Civil y Emergencias. *Guía Técnica: Metodología para el análisis de riesgos. Visión General*. [En línea] [Citado el: 28 de Noviembre de 2008.] http://www.proteccioncivil.org/es/DGPCE/Informacion_y_documentacion/catalogo/carpeta02/carpeta22/guiatec/Vision_general/vision_4.htm.

Dueñas, Denise. 2008. *Aplicación de una estrategia para la Identificación de Riesgos en un proyecto caso de estudio*. Universidad de las Ciencias Informáticas. La Habana : s.n., 2008.

Estéves, J y otros. *Implementación y Mejora del Método de Gestión de Riesgos del SEI en un proyecto universitario de desarrollo de software*. [En línea] http://www.ewh.ieee.org/reg/9/etrans/ieee/issues/vol3/vol3issue1March2005/3TLA1_13Esteves.pdf.

European Centre for Softcomputing. [En línea] [Citado el: 29 de Noviembre de 2008.] <http://www.softcomputing.es/es/portada.php>.

Figueredo, Yelaine. 2007. *Análisis y Gestión de Riesgos para el desarrollo de las aplicaciones del proyecto Atención primaria de Salud.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2007.

Fuente, Aquilino Adolfo y Cueva, Juan Manuel. 2006. *Gestión de Riesgo.* [En línea] 2006. [Citado el: 28 de Octubre de 2008.] <http://www.di.uniovi.es/~aquilino/Asignaturas/ProyectosInformatica/Documentos/Proyectos.v2006.C7.V2.pdf>.

Gallagher, B.P. *October 1999: Handbook. Software Acquisition Risk Management Key Process Area (KPA)— A Guidebook Version 1.02.*

Galway, Lionel. 2004. *Quantitative Risk Analysis for Project Management. A Critical Review.* [En línea] 2004. [Citado el: 28 de Noviembre de 2008.] http://www.rand.org/pubs/working_papers/2004/RAND_WR112.pdf.

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

García, Noe Aller. 2007. *Mejora y ampliación de la aplicación de gestión de riesgos bajo el framework JRISK para empresa dedicada a realizar proyectos software.* [En línea] Universidad de Oviedo, 2007. [Citado el: 28 de Octubre de 2008.] <http://petra.euitio.uniovi.es/~i1079196/PFC/Documentacion.pdf>.

Gil, Pilar. 2009. *MOPROSOFT.* [En línea] 10 de Abril de 2009. <http://www.itpuebla.edu.mx/Eventos/MemoriasyResSemanalInformatica2007/04-PiliGOmezconferencia%20moprosoft%202007.pdf>.

2005. GOLIATH. *Securac's Acertus(TM) Governance Solution Achieves 'Powered by SAP NetWeaver(TM)' Status.* [En línea] 2005. http://goliath.ecnext.com/coms2/gi_0199-4126629/Securac-s-Acertus-TM-Governance.html.

González, Julio. 2005. *Análisis de los Riesgos en la Agencia Estatal de Meteorología mediante el empleo de Magerit y PILAR. Ministerio de Medio Ambiente.* [En línea] 2005. [Citado el: 1 de Diciembre de 2008.] http://www.csi.map.es/csi/pdf/2008_Magerit_AEMET.pdf.

- 2006.** GPI Consultores. *Disciplina de Administración del proyecto- Microsoft Solution Framework (MSF)*. [En línea] 2006. [Citado el: 12 de Marzo de 2009.] <http://www.gpicr.com/msf.aspx>.
- Gracia, Joaquín. 2005.** *CMM – CMMI*. [En línea] 2005. [Citado el: 21 de Octubre de 2008.] <http://www.ingenierosoftware.com/calidad/cmm-cmmi.php>.
- Hechavarría, Lisandra y Mayol, Ariadna. 2008.** *Gestión de Riesgo en el Proyecto Sistema de Gestión Fiscal*. . Universidad de las Ciencias Informáticas . La Habana : s.n., 2008.
- Hidalgo, Mirelys y Rodríguez, Erick. 2007.** *Una guía para el tratamiento de riesgos para el software educativo en la Universidad de las Ciencias Informáticas*. Universidad de las Ciencias Informáticas. La Habana : s.n., 2007.
- Higuera, Ronald. 2005.** Software Engineering Institute (SEI). *Team Risk Management*. [En línea] 2005. [Citado el: 12 de Marzo de 2009.] <http://www.stsc.hill.af.mil/crosstalk/1995/01/TeamRisk.asp>.
- Higuera, Ronald y Haimés, Yacov. 1996.** Software Engineering Institute (SEI). *Software Risk Management*. [En línea] 1996. [Citado el: 12 de Marzo de 2009.] <http://www.sei.cmu.edu/pub/documents/96.reports/pdf/tr012.96.pdf>.
- Higuera, Ronald y otros. 1994.** Software Engineering Institute (SEI). *An Introduction to Team Risk Management Version 1.0*. [En línea] 1994. [Citado el: 12 de Marzo de 2009.] <http://www.sei.cmu.edu/pub/documents/94.reports/pdf/sr01.94.pdf>.
- Hillson, David y Hulett, David. 2004.** *Calculando probabilidades de riesgos: métodos alternativos*. [En línea] 2004. [Citado el: 10 de Abril de 2009.] http://www.projectcharter.com/documents/white_papers_sp/Assessing%20risk%20prob%20Hillson-Hulett%20Sp%202004.pdf.
- 2007.** IEEE Standards Association. *IEEE Standard for Software Life Cycle Processes—Risk Management -Description*. [En línea] 2007. [Citado el: 21 de Octubre de 2008.] http://standards.ieee.org/reading/ieee/std_public/description/se/1540-2001_desc.html.

Ingersoll, K. 2000.. *Risk World Software: computer programs for risk assessment and risk management. Risk- Related software.* [En línea] 2000. [Citado el: 12 de Marzo de 2009.] <http://www.riskworld.com/SOFTWARE/SW5SW001.HTM>.

Institute of Electrical and Electronics Engineers (IEEE) . [En línea] [Citado el: 20 de Octubre de 2008.] <http://www.ieee.org>.

2008. International Project Management Association (IPMA). [En línea] 2008. [Citado el: 13 de Octubre de 2008.] <http://www.ipma.ch>.

International Standard Organization (ISO). [En línea] [Citado el: 20 de Octubre de 2008.] <http://www.iso.org/iso/about.htm>.

International Standard Organization (ISO). *ISO/IEC 16085:2006.* [En línea] [Citado el: 22 de Octubre de 2008.] http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=40723.

ISO SPICE. [En línea] [Citado el: 21 de Octubre de 2008.] <http://www.isospice.com/categories/ISO%7B47%7DIEC-15504-Standard/>.

Kuna, Horacio y otros. 2008. *Plan de Riesgos para la implementación, desarrollo y mantenimiento de componentes de Web 2.0 en Bibliotecas, caso de estudio en una Biblioteca Especializada.* [En línea] 2008. [Citado el: 12 de Marzo de 2009.] <http://www.amicus.udesa.edu.ar/6bibliotecadigital/ponencias/PONENCIA%20MISIONES%20RIESGOS%20Web2.0.doc>.

López, Herlys. 2008. *Aplicación de una estrategia para la planificación de la gestión de riesgos en un proyecto caso de estudio.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2008.

López, Yanisleidy y Álvarez, Tailys. 2007. *Propuesta para la Gestión de Riesgo en los proyectos productivos de la UCI.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2007.

MB Risk Management. *Financial System Software Company. Universal Add-ins London, England.* [En línea] [Citado el: 10 de Abril de 2009.] <http://www.mbrm.com/broch.pdf>.

2008. Microsoft. *Microsoft Operations Framework 4.0.* [En línea] 2008. [Citado el: 14 de Octubre de 2008.] <http://technet.microsoft.com/en-us/library/cc506049.aspx>.

Microsoft Corporation. [En línea] [Citado el: 21 de Octubre de 2008.]
<http://www.microsoft.com/en/us/default.aspx>.

2002. Microsoft Corporation. *MSF Risk Management Discipline. Versión 1.1.* [En línea] 2002.
<http://download.microsoft.com/download/d/f/a/dfa02d32-0aee-4603-be33-1b099d6886a0/MSF%20Risk%20Management%20Discipline%20v.1.1.pdf>.

Mora, Christian y Riofrío, Argenis. 2008. *Sistemas Operativos. Simulación de la Planificación RR.* [En línea] 2008. [Citado el: 10 de Noviembre de 2008.]
<http://www.utpl.edu.ec/blog/sistemasoperativos/category/round-robin/>.

Moreira, Dayamys. 2008. *Modelo de Gestión de Riesgos y análisis de propuesta de aplicación para la ejecución del mismo.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2008.
Mural Universidad de Valencia. [En línea] [Citado el: 21 de Octubre de 2008.]
<http://mural.uv.es/juanama/astronomia/montecarlo.htm>.

2009. PALISADE. *Fabricante de software para análisis de riesgos y de decisiones. @Risk Un nuevo estándar en análisis de riesgos.* [En línea] 2009. [Citado el: 10 de Abril de 2009.] <http://www.palisade-lta.com/risk/>.

Piñero, Pedro. 2005. *Un modelo para el aprendizaje y la clasificación automática basado en técnicas de softcomputing.* Universidad Central "Marta Abreu" de Las Villas. Villa Clara : s.n., 2005.

PMI. 2004. *Guía de los Fundamentos de la Dirección de Proyectos (PMBOK).* 1-930699-73-5. Pennsylvania : s.n., 2004.

2007. Portal del Gobierno de Chile. Servicio Nacional de Aduanas. *El Proceso de Gestión de Riesgos.* [En línea] 2007. [Citado el: 28 de Octubre de 2008.]
http://www.aduana.cl/prontus_aduana/site/artic/20070228/asocfile/20070228130834/asocfile120050916161822.pdf.

Pressman, Roger. *Ingeniería de Software, Un enfoque práctico.* [trad.] Darrel Ince. Vol. Quinta Edición.

2007. *Procedimiento para diagnóstico de riesgos.* Dirección de Producción 1, Universidad de las Ciencias Informáticas. La Habana : s.n., 2007.

Project Management Institute (PMI). [En línea] [Citado el: 13 de Octubre de 2008.] <http://www.pmi.org>.
Ramírez, Arielena y Salazar, Leidi. 2008. *Guía Metodológica para Administrar los Riesgos en los proyectos productivos de Realidad Virtual.* , Universidad de las Ciencias Informáticas. La Habana : s.n., 2008.

2003. *Rational Unified Process* . Rational Software Corporation. 2003.

2006. Repository projects python. [En línea] 2006. [Citado el: 10 de Noviembre de 2008.] <http://svn.python.org/view/python/trunk/Misc/HISTORY?rev=51814&view=markup>.

Risk World . [En línea] [Citado el: 22 de Octubre de 2008.] <http://www.riskworld.com>.

Rodríguez, Dayamí y Fernández, Haydee. 2007. *Análisis del Método de Estimación empleado para el desarrollo del proyecto SIGEP.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2007.

ROPPONEN, J. y LYYTINEN, K. 2000. *Components of Software Development Risk: Hot to address Them? IEEE transactions on software engineering.* [En línea] 2000.

Rosemberg, Linda, Hammer, Theodore y Gallo, Albert. 1999. NASA Software Assurance Technology Center. [En línea] 1999. [Citado el: 22 de Octubre de 2008.] http://satc.gsfc.nasa.gov/support/ASM_FEB99/crm_at_nasa.html.

Singh, Raghu. 1999. *An Introduction to International Standard ISI/IEC 12207. Software Life Cycle Process.* [En línea] 1999. [Citado el: 21 de Octubre de 2008.] <http://www.abelia.com/docs/12207tut.pdf>.

—. *International Standard ISO/IEC 12207 Software Life Cycle Processes.* [En línea] [Citado el: 21 de Octubre de 2008.] <http://www.abelia.com/docs/12207cpt.pdf>.

2000. Society for Risk Analysis (SRA). [En línea] 2000. [Citado el: 12 de Octubre de 2008.] <http://www.sra.org/journal.php>.

Software Engineering Institute (SEI). . *CMMI*. [En línea] Universidad Carnegie Mellon. [Citado el: 21 de Octubre de 2008.] <http://www.sei.cmu.edu/cmami/general/>.

2008. Software Engineering Process Technology (SEPT). *The Expert on ISO/IEC 12207* . [En línea] 2008. [Citado el: 22 de Octubre de 2008.] <http://www.12207.com/index.html>.

Supercomputing. *Software. Economy and Finance*. [En línea] [Citado el: 10 de Abril de 2009.] <http://www.supercomputing.it/software.php?cat=16>.

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

Van, Roger. 1992. Software Engineering Institute (SEI). *Software Development Risk: Opportunity, Not Problem*. [En línea] 1992. [Citado el: 10 de Octubre de 2008.] <http://www.sei.cmu.edu/pub/documents/92.reports/pdf/tr30.92.pdf>.

Williams, Ray C. 2003. Software Engineering Institute (SEI). *New Directions in Risk Management at the SEI*. [En línea] 2003. [Citado el: 15 de Octubre de 2008.] <http://www.sei.cmu.edu/risk/new-directions.pdf>.

Williams, Ray y Pandelios, George y otros. 1999. Software Engineering Institute (SEI). *Software Risk Evaluation (SRE) Team Member's Notebook (Version 2.0)*. [En línea] 1999. <http://www.sei.cmu.edu/pub/documents/99.reports/pdf/99tr029-app.pdf>.

YPFB Transporte. *Eavluación de Impacto Ambiental Construcción Lateral La Vertiente. Análisis de Riesgos*. [En línea] [Citado el: 11 de Marzo de 2009.] <http://www.transredes.com/pdfs/MedioAmb/eeia/eeiaLaVertiente/Cap6.pdf>.

Zulueta, Yeleny y Yadira, Ruíz. 2007. Ilustrados. *Tendencias Actuales de la Gestión de Riesgos*. [En línea] 2007. [Citado el: 28 de Noviembre de 2008.] <http://www.ilustrados.com/publicaciones/EEIkuyFZuAkrUVZhMc.php>.

Anexos

Anexo 1: Taxonomía del desarrollo de software

Ingeniería:

<u>Requerimientos:</u>	<u>Diseño:</u>	<u>Código y Unidades de Prueba:</u>	<u>Integración y Pruebas:</u>	<u>Especialidades:</u>
Estabilidad	Funcionalidad	Factibilidad	Ambiente	Mantenimiento
Completamiento	Dificultad	Pruebas de unidad	Producto	Fiabilidad
Claridad	Interfaces	Implementación	Sistema	Seguridad
Validez	Rendimiento			Factores humanos
Factibilidad	Capacidad de probarse			Especificaciones
Precedentes	Restricciones de hardware			
Escala	Software de terceras partes			

Ambiente de Desarrollo:

<u>Procesos de desarrollo:</u>	<u>Sistema de desarrollo:</u>	<u>Procesos de gestión:</u>	<u>Métodos de gestión:</u>	<u>Ambiente de trabajo:</u>
Formalidad	Capacidad	Planeación	Monitoreo	Actitud ante la calidad
Adecuación	Adecuación	Organización del proyecto	Gestión del personal	Cooperación
Control	Usabilidad	Experiencia en la gestión	Aseguramiento de la calidad	Comunicación
Familiaridad	Familiaridad	Interfaces del programa	Gestión de configuración	Moral
	Fiabilidad			Entusiasmo
	Soporte			
	Entregas			

Restricciones:

<u>Recursos:</u>	<u>Contrato:</u>	<u>Interfaces del programa:</u>
Cronograma	Tipo de contrato	Cliente
Personal	Restricciones	Contratista
Presupuesto	Dependencias	Subcontratistas
Facilidades		Contratista principal
		Vendedores
		Política

Anexo 2: Herramienta EAR/PILAR

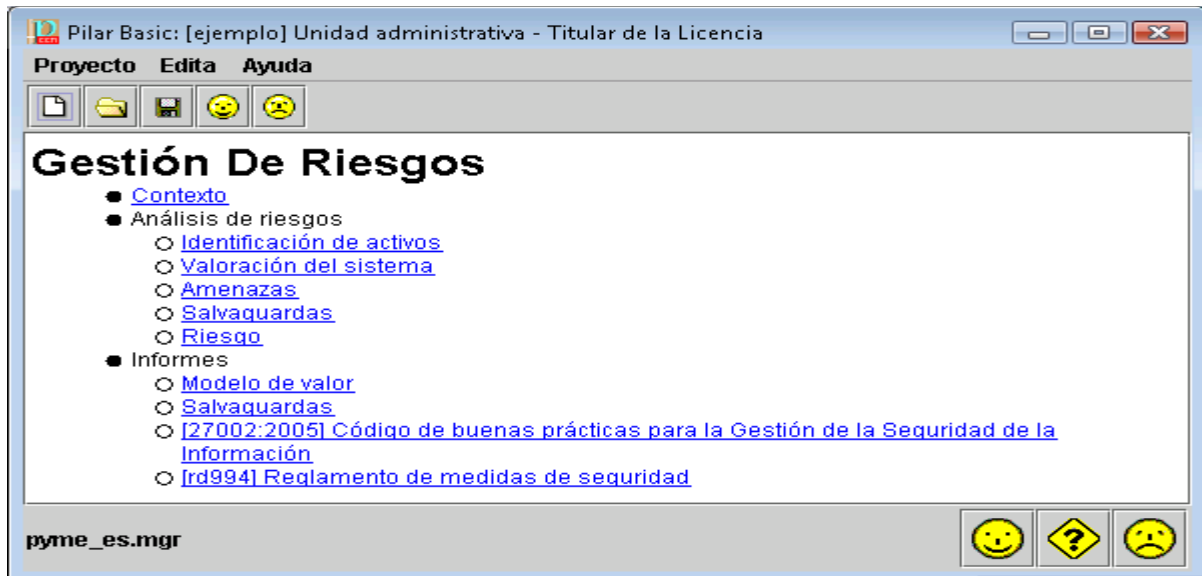


Figura 26 Herramienta EAR. Unidad Administrativa



Figura 27 Herramienta EAR

ejemplo: Valoración de las amenazas - usuario

Editar Exportar Importar

activo	nivel	[D]	[I]	[C]	[A]	[T]
<input type="checkbox"/> ACTIVOS						
<input type="checkbox"/> [B] Capa de negocio						
<input checked="" type="checkbox"/> [D_files] Expedientes en curso		100%	50%	100%	100%	100%
<input type="checkbox"/> [E.1] Errores de los usuarios	A	10%	10%	10%		
<input type="checkbox"/> [E.2] Errores del administrador	M	20%	20%	10%	10%	20%
<input type="checkbox"/> [E.3] Errores de monitorización (log)	M					50%
<input type="checkbox"/> [E.4] Errores de configuración	M	50%	10%	10%	50%	50%
<input type="checkbox"/> [E.15] Alteración de la información	A		1%			
<input type="checkbox"/> [E.16] Introducción de falsa información	MA		1%			
<input type="checkbox"/> [E.19] Divulgación de información	M			10%		
<input type="checkbox"/> [A.4] Manipulación de la configuración	B	50%	10%	50%	100%	100%
<input type="checkbox"/> [A.11] Acceso no autorizado	MA	100%	10%	50%	50%	
<input type="checkbox"/> [A.14] Interceptación de información (escucha)	A			50%		
<input type="checkbox"/> [A.15] Modificación de información	A		50%			
<input type="checkbox"/> [A.16] Introducción de falsa información	A		50%			
<input type="checkbox"/> [A.19] Divulgación de información	A			100%		
<input type="checkbox"/> [S_in_person] Tramitación presencial		100%	50%	50%	100%	100%
<input type="checkbox"/> [S_remote] Tramitación remota		100%	50%	50%	100%	100%
<input type="checkbox"/> [IS] Servicios internos						
<input type="checkbox"/> [E] Equipamiento						
<input type="checkbox"/> [SS] Servicios subcontratados						
<input type="checkbox"/> [L] Instalaciones						
<input type="checkbox"/> [P] Personal						

- 1 + -1

cargar biblioteca eliminar

Figura 28 Herramienta EAR. Valoración de las amenazas

ejemplo: riesgo acumulado - usuario

potencial current 3m 1y target

	activo / amenaza	F	E	impacto	riesgo
<input type="checkbox"/>	ACTIVOS		120d	[7]	{5.9}
<input type="checkbox"/>	☞ [B] Capa de negocio				
<input type="checkbox"/>	☞ [D_files] Expedientes en curso		2d	[7]	{5.9}
<input type="checkbox"/>	☞ [E.1] Errores de los usuarios	10	[2h]	[5]	{4.8}
<input type="checkbox"/>	☞ [E.2] Errores del administrador	1	[6h]	[5]	{3.9}
<input type="checkbox"/>	☞ [E.4] Errores de configuración	0,5	[6h]	[5]	{3.5}
<input type="checkbox"/>	☞ [E.18] Destrucción de la información	10	[2h]	[5]	{4.8}
<input type="checkbox"/>	☞ [A.4] Manipulación de la configuración	0,1	[2h]	[5]	{3.0}
<input type="checkbox"/>	☞ [A.18] Destrucción de la información	10	[2d]	[7]	{5.9}
<input type="checkbox"/>	☞ [S_in_person] Tramitación presencial		1h	[5]	{4.8}
<input type="checkbox"/>	☞ [S_remote] Tramitación remota		1h	[5]	{4.8}
<input type="checkbox"/>	☞ [IS] Servicios internos				
<input type="checkbox"/>	☞ [E] Equipamiento				
<input type="checkbox"/>	☞ [SS] Servicios subcontratados				
<input type="checkbox"/>	☞ [L] Instalaciones				
<input checked="" type="checkbox"/>	☞ [P] Personal				

- 1 + +1 dominio fuente gestiona leyenda html xml csv ☺ ?

Figura 29 Herramienta EAR. Riesgo acumulado

Anexo 3: Herramienta @Risk

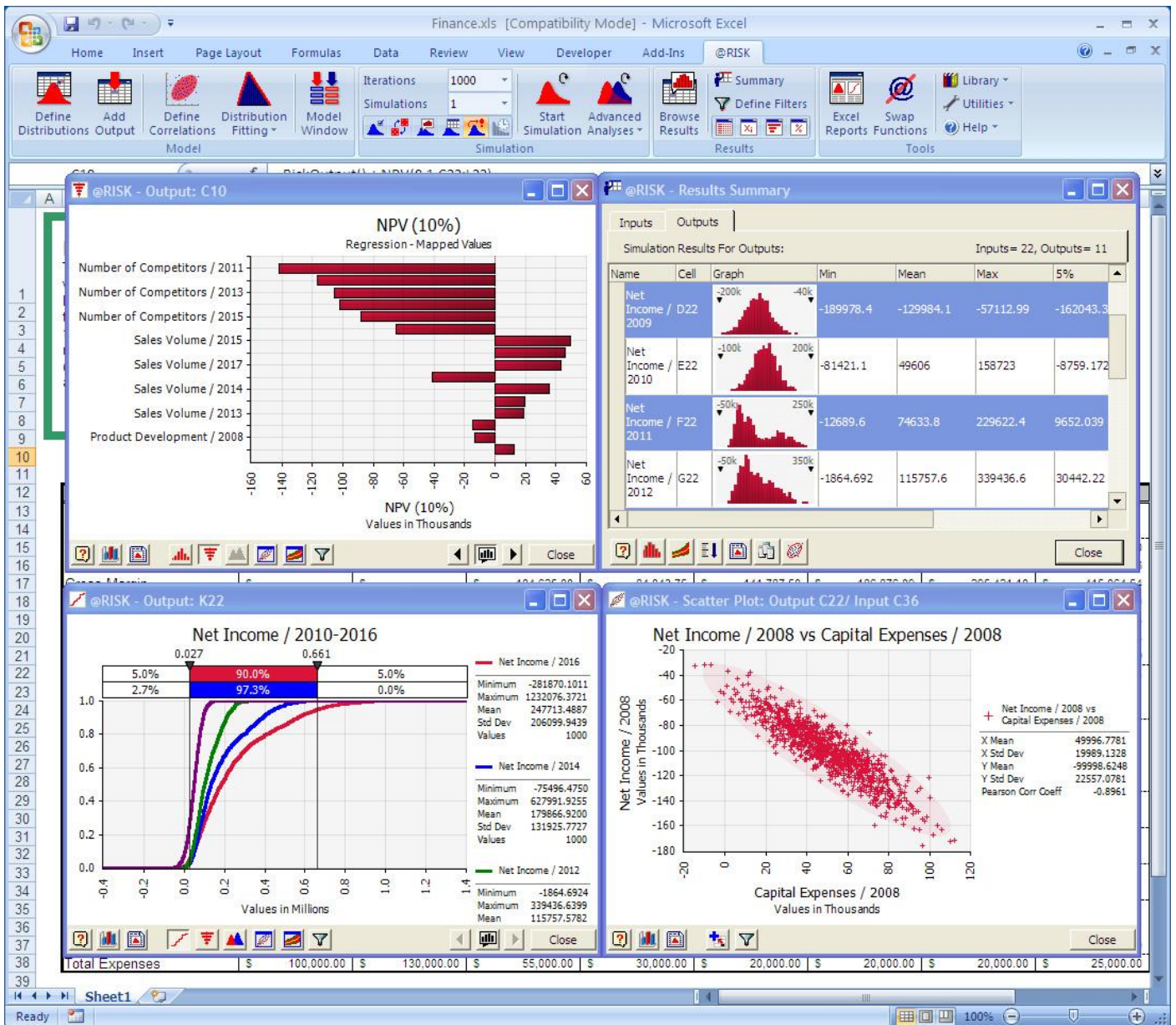


Figura 30 Herramienta @Risk

Anexo 4: Herramienta MBRM

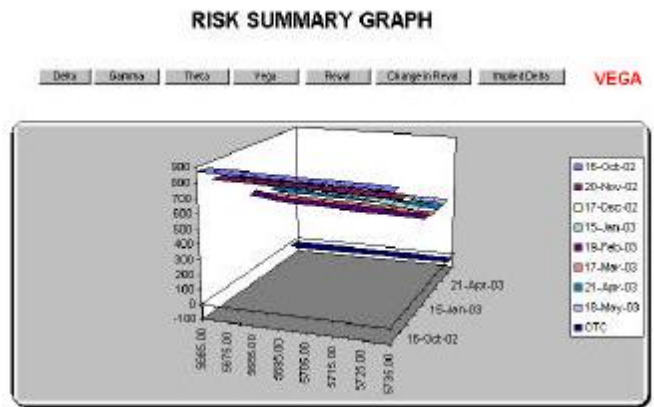


Figura 31 Herramienta MB Risk Management

Anexo 5: Glosario de Términos de Inteligencia Artificial

- **Antecedentes de la regla:** se construyen con los atributos que influyeron en la formación del conjunto cerrado
- **Atributos continuos:** pueden tomar cualquier valor, generalmente dentro de un rango dado. Ejemplo: la calificación en una evaluación, que puede tomar cualquier valor entre 0 y 10
- **Atributos de dominio continuos:** atributos que pueden tomar cualquier valor, generalmente dentro de un rango dado.
- **Atributos de dominios discretos:** atributos que solo pueden tomar como valor uno de los previstos en una lista de valores.
- **Atributos discretos:** sólo pueden tomar como valor uno de los provistos en una lista predefinida de valores. Ejemplo: el sexo del alumno, que solo puede tomar los valores F (femenino) o M (masculino).
- **Atributos simbólicos:** atributos cuyo valor es una cadena de caracteres
- **Clústeres:** se refiere a cuando los artículos de la misma clase son lo más similar posible y artículos de diferentes clases son tan disímiles como sea posible. Dicho agrupamiento también puede ser pensado como una forma de compresión de datos, donde un gran número de muestras se convierten en un pequeño número de prototipos o grupos representativos
- **Clusterización:** El objetivo principal de la clusterización es dividir los puntos que representan diferentes observaciones en clústeres, de tal modo que los puntos de un mismo clúster estén más cercanos entre ellos que los puntos que pertenecen a diferentes clústeres.
- **Conjuntos borrosos:** un conjunto borroso en un universo de discurso está caracterizado por una función de pertenencia, la cual a cada elemento en el dominio le asigna un grado de pertenencia al conjunto en el intervalo $[0,1]$. De esta forma un mismo elemento puede pertenecer a varios conjuntos simultáneamente solo que con cierto grado de pertenencia. Cada conjunto borroso tiene asociado además un término lingüístico de forma tal que la función de pertenencia asociada a un conjunto está ligada a una palabra como por ejemplo: bajo, medio, más o menos alto, alto, muy alto, etc.
- **Conjunto borroso asociado a cada antecedente:** será el que se corresponda con el término lingüístico presente en el valor del atributo en cuestión del caso
- **Conjunto cerrado:** un conjunto es cerrado, si y solo si, todos sus miembros tienen el mismo consecuente

- **Discretización:** es el proceso de transformar atributos de dominio continuo a dominios discretos. Dado un dominio definido como un intervalo $[a,b]$, discretizar el atributo significa producir una partición o intervalo. A partir de allí los valores de los atributos son etiquetas que representan cada elemento de la partición. Los métodos para discretizar son: Intervalos de igual tamaño, e Igual frecuencia por intervalo.
- **Estandarizar:** significa seguir procedimientos uniformes y constantes en todas las fases del acopio de datos
- **Función de membresía:** se usan para encontrar un grado de membresía. Son funciones que devuelven un valor entre 0 y 1 indicando el grado de membresía de un elemento con respecto a un conjunto. Esta función transforma la expresión ambigua “pueden o no pueden” en la expresión precisa “pertenecen con diferente grado de pertenencia”. Tiene valor 0 para los elementos que no pertenecen al conjunto, valor 1 para los elementos del conjunto bien definidos por la función característica y valores intermedios para los elementos de frontera.
- **Fuzzy:** El término Fuzzy significa impreciso, vago, poco claro. Cuando se aplica a la teoría de conjuntos Fuzzy, se refiere a elementos pertenecientes a las fronteras (difusas) de los conjuntos
- **Fuzzyficación:** se toma un dato del sistema y se lo convierte en dato difuso. El proceso de fuzzificación parte de los atributos discretizados, permitiendo obtener de cada atributo una variable lingüística, y de cada intervalo se construye una función de membresía para cada término lingüístico de dicha variable lingüística, especificando el solapamiento deseado.
- **Grado de pertenencia:** En un conjunto difuso a cada elemento del universo se le asocia un grado de pertenencia, que es un número entre 0 y 1, a ese conjunto. La pertenencia de un valor a una instancia concreta puede ser cuantificada por este grado.
- **Inteligencia artificial:** rama de la ciencia informática dedicada al desarrollo de agentes racionales no vivos.
- **Lógica borrosa:** Se define como la parte de la inteligencia artificial cuya máxima es que todo se basa en un grado. Es decir, todo es o no es en un determinado grado. Además, se dice que las reglas existentes en esta lógica, son reglas definidas desde el sentido común.
- **Método Cramer:** La Regla de Cramer es una regla para resolver un sistema de ecuaciones lineales. En el contexto de esta investigación se refiere a una de las técnicas para escoger el atributo idóneo en el algoritmo MLRUL. Es una medida de la interrelación entre las variables.

- **Método Mántaras:** técnica para escoger el atributo idóneo en el algoritmo MLRUL. Calcula las distancias entre las particiones como medida de selección del atributo
- **Método MLRelevance:** técnica para escoger el atributo idóneo en el algoritmo MLRUL. Es una nueva medida de selección del atributo. Este último brinda la heterogeneidad entre elementos que pertenecen a diferentes clases y la homogeneidad entre los elementos que pertenecen a la misma clase.
- **Reglas borrosas:** a partir del dato difuso, encontrar el grado de membresía de ése dato a cada conjunto difusos
- **Reglas de inferencia borrosas:** es un esquema para construir inferencias válidas. Estos esquemas establecen relaciones sintácticas entre un conjunto de fórmulas llamados premisas y una aserción llamada conclusión. Estas relaciones sintácticas son usadas en el proceso de inferencia, por el que se llega a nuevas aserciones verdaderas a partir de otras ya conocidas.
- **Softcomputing:** es una rama de la Inteligencia Artificial, centrada en el diseño de sistemas inteligentes capaces de manejar adecuadamente la información imprecisa, incierta y/o incompleta. Los sistemas borrosos, las redes neuronales, la computación evolutiva, el razonamiento probabilístico y las combinaciones de dichas técnicas son considerados como softcomputing
- **Variables lingüísticas:** constituyen el centro de las técnicas de modelación borrosa. Encapsula las propiedades de aproximación o conceptos imprecisos en una forma sistemática y conveniente computacionalmente. Estas reducen la aparente complejidad de describir un sistema por la correspondencia a una etiqueta semántica para el concepto fundamental. Aún una variable lingüística siempre representa un espacio borroso. En su estructura se encuentran: un conjunto borroso, y clasificadores.

Anexo 6: Registro de Riesgos

Registro de riesgos

Interno

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

Control de versiones

Fecha	Versión	Descripción	Autor
<dd/mmm/yy>	<x.x>	<detalles>	<nombre>

Reglas de Seguridad

El que recibe el documento asume la custodia y control, comprometiéndose a no reproducir, divulgar, difundir o de cualquier manera hacer de conocimientos público su contenido, excepto para cumplir el propósito para el cual se ha generado.

Estas reglas son aplicables a las 141 páginas de este documento.

Introducción

1.1. Propósito

[Definir términos Generales establecidos pro la Dirección de Calidad y Normas de la UCI para los proyectos Productivos]

1.2. Alcance

[Todos los proyectos de la UCI]

1.3. Referencias

[Lista de documentos a los que se hace referencia]

Código	Título
[1]	Documento 1
[2]	Documento 2
[3]	Modelo de Diseño - Módulo de Administración v0.0

1.4. Glosario

[En el glosario aparecen un grupo de términos básicos para los proyectos productivos de la UCI]

2. Riesgos

ID:	Etapas:	Estado:	Prioridad:
Categoría:		Entrevistado:	
Definición del riesgo: (evento): (consecuencia):			
Causas:			
Posible Probabilidad (opcional):		Posible Impacto (opcional):	

Probabilidad:	Impacto:
Frecuencia:	Certidumbre:
Marco de tiempo:	Estrategia de respuesta:
Responsables:	
Riesgos Relacionados:	
Observaciones:	

Anexo 7: Lista Priorizada de Riesgos Cuantificados

Lista Priorizada de Riesgos

Interno

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

Control de versiones

Fecha	Versión	Descripción	Autor
<dd/mmm/yy>	<x.x>	<detalles>	<nombre>

Reglas de Seguridad

El que recibe el documento asume la custodia y control, comprometiéndose a no reproducir, divulgar, difundir o de cualquier manera hacer de conocimientos público su contenido, excepto para cumplir el propósito para el cual se ha generado.

Estas reglas son aplicables a las 141 páginas de este documento.

1. Introducción

1.1. Propósito

[Esta lista de riesgos incluye aquellos riesgos que representan la mayor amenaza o presentan la mayor oportunidad para el proyecto. Se incluyen los riesgos que requieren la mayor contingencia de costes y aquellos que tienen más probabilidad de influir sobre el camino crítico.]

1.2. Alcance

[Todos los proyectos de la UCI]

1.3. Referencias

[Lista de documentos a los que se hace referencia]

Código	Título
[1]	Documento 1
[2]	Documento 2
[3]	Modelo de Diseño - Módulo de Administración v0.0

1.4. Glosario

[En el glosario aparecen un grupo de términos básicos para los proyectos productivos de la UCI]

2. Lista Priorizada de Riesgos

ID	Riesgo	Prioridad (Priorizado, Medianamente Priorizado, No Priorizado)	Mayor Contingencia de Costes (Marcar con una X)	Mayor probabilidad de influir sobre el camino crítico (Marcar con una X)
1	Alcance u objetivos no claros o malentendidos	Priorizado	X	X

Anexo 8: Ejemplo de Limpieza de Datos

La limpieza de datos se puede dividir en seis pasos:

1. Separar en elementos
2. Estandarizar
3. Verificar
4. Machear
5. Agrupar
6. Documentar

Para ilustrar estos seis pasos, se considera el siguiente ejemplo:

A través de una aplicación que maneja informaciones de clientes de una organización se introduce la siguiente dirección:

*Sergio Pérez y Elena Ruiz Representantes de Gerencia de Comercio
Ave. Porvenir, e/ Calle 4ta y 5ta, #27
Reparto. Vigía,
Sta. Clara, V. C.,
50200*

El **primer paso** en la limpieza de esta dirección es separarla en elementos:

Nombre (1): Sergio

Apellido (1): Pérez

Nombre (2): Elena

Apellido (2): Ruiz

Relación: Representantes de

Entidad: Gerencia de Comercio

Nombre de la Calle principal: Ave. Porvenir

Entre Calle 1: 4ta

Entre Calle 2: 5ta

Número de la Casa: #27

Reparto: Vigía

Ciudad: Sta. Clara

Provincia: V. C.

Código Postal: 50200

El **segundo paso** es estandarizar los elementos. Reconocemos que "Ave." es "Avenida". Se sospecha que el elemento "Sta. Clara" pudiera ser la ciudad de "Santa Clara", se hace el cambio provisional y en el paso de verificación se revisará si existe una ciudad o municipio con este nombre. Se hace lo mismo con el elemento "V. C.", se cambia por "Villa Clara".

El **tercer paso** es verificar si los elementos estandarizados contienen errores en su contenido. Aquí se verificaría, en el ejemplo, si el código postal 50200 pertenece a Santa Clara. Si se tuviera otra dirección de "Sergio Pérez" o de "Elena Ruiz" y la que se está analizando como ejemplo tuviera algún error, se pudieran comparar los códigos postales y sustituirlo por el correcto.

El **macheo** consiste en encontrar otros "Sergio Pérez" o "Elena Ruiz" en los registros de clientes existentes y asegurarse de que todos sus elementos de la dirección son idénticos.

El **agrupamiento** consiste en reconocer que "Sergio" y "Elena" constituyen una agrupación porque ellos comparten la misma dirección. Además, se pudiera tener en otra fuente de datos (externa o interna) que indique que estas personas son hermanos o están casadas; motivo de más para que compartan la misma dirección.

El **sexto paso** consiste en documentar los resultados de los pasos anteriores en metadatos. Esto ayuda a que las siguientes limpiezas sean más capaces de reconocer direcciones y a que los usuarios finales de las aplicaciones puedan llevar a cabo mejor las operaciones, así como para entender mejor la base de datos de los clientes.

Como se puede observar este proceso es bastante tedioso hacerlo manualmente y para hacerlo automatizado se necesita de aplicaciones sofisticadas que contengan algoritmos de análisis gramatical (parsing) de direcciones, algoritmos de macheo, e inmensas tablas con gran cantidad de entradas que provean sinónimos para las diferentes partes de las direcciones.

En algunos casos, es posible crear programas de limpieza efectivos. En el caso de bases de datos, grandes, imprecisas e inconsistentes, el uso de las herramientas comerciales, ya existentes, puede ser casi obligatorio.

Anexo 9: Declaración de Revisión

Declaración de Revisión

Nombre del Revisor: Henry Cruz Mulet

Cargo: Jefe de Producción

Proyecto al que Representa: ERP- Cuba

Declaro haber realizado la revisión de la Base de Riesgos propuesta en la presente Tesis de Pregrado: Propuesta de Modelo para el Análisis de Riesgos basado en técnicas de Soft Computing, de las autores Danelys Brito y Maylen Bon Pérez, a ser aplicada en el Centro de Tecnologías de Almacenamiento y Análisis de Datos (CENTALAD).

Para que así conste firmamos la presente a los 14 días del mes de mayo del año 2009

Henry Cruz Mulet

Firma del Revisor

Danelys Brito González

Firma del Autor

Maylen Bon Pérez

Firma del Autor

Declaración de Revisión

Nombre del Revisor: Ing. Henrik Pestano Pino

Cargo: Subdirector de Producción

Proyecto al que Representa: Paquete de Herramientas para la Ayuda a la Toma de Decisiones (PATDSI)

Declaro haber realizado la revisión de la Base de Riesgos propuesta en la presente Tesis de Pregrado: Propuesta de Modelo para el Análisis de Riesgos basado en técnicas de Soft Computing, de las autores Danelys Brito y Maylen Bon Pérez, a ser aplicada en el Centro de Tecnologías de Almacenamiento y Análisis de Datos (CENTALAD).

Para que así conste firmamos la presente a los 14 días del mes de mayo del año 2009

Ing. Henrik Pestano Pino

Firma del Revisor

Danelys Brito González

Firma del Autor

Maylen Bon Pérez

Firma del Autor